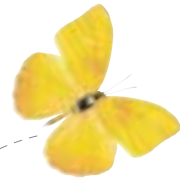


오라클 사용자를 위한 DB2 가이드

저자 : 한국IBM TSS Techline 안명주
Published by IBM 정보관리사업부



서문



DB2는 IBM Ted Codd 박사의 “관계형 모델”에 기반한 최초의 RDBMS이라 할 수 있습니다. 처음에는 Mainframe에서 운영되는 DBMS로 시작하여 약 15년 전부터 유닉스 및 윈도우 등 다양한 플랫폼을 지원하고 있습니다.

제가 DB2의 사용자 입장에서 보면 DB2는 시장의 어떤 DBMS 보다 사용하기 쉽고 편리하게 되어 있습니다. 그러나 한글화된 정보의 부족으로 많은 분들이 DB2에 대해 막연한 두려움을 가지고 있는 듯 합니다.

이에 어떻게 하면 사용자의 입장에서 보다 쉽고 명확하게 DB2의 사용 방법을 개발자분이나 DBA분들에게 전달할 수 있을까를 고민하게 되었고 그 결과, 중심을 DB2에 놓지 않고 시중에서 쉽게 접할 수 있는 오라클과 DB2를 비교한다면 보다 쉽고 명확하게 DB2 사용방법에 대해 전달할 수 있을 것 같아 이 책을 발간하게 되었습니다.

이 책은 오라클과 DB2의 비교로 시작하여 오라클 기능에 해당하는 DB2 기능을 실습을 통해 쉽고 빨리 익힐 수 있도록 구성하였습니다. 특히 오라클과 DB2의 차이점과 공통점을 한 눈에 알아보기 쉽게 비교하여 오라클에 익숙한 사용자분들이 DB2 또한 쉽게 익힐 수 있도록 구성하였습니다. 또한 오라클에 익숙하지 않더라도 DB2의 개념 및 사용 방법에 대해서 자세히 설명되어 있어 DB2를 익히고자 하는 분들에게도 적합한 DB2 입문서라 하겠습니다. 이 책에서 다루어진 제품은 오라클 11g와 DB2 9.5를 비교한 내용이니 참고해 주시기 바랍니다.

끝으로 이 책이 나오기까지 자기 일처럼 교정을 봐주신 IBM 테크라인 팀원분들, SIG 멤버들, 이승용 상무님, 손종민 본부장님과 마케팅의 최은영 차장님, 박현정 부장님 및 디자인을 도와 주신 오연주씨에게 감사의 말씀 드립니다.



▼ 저자 소개 안명주

오라클 데이터베이스를 기반으로 Billing 시스템 등 다수의 오라클 프로젝트 개발자로 참여하였으며 IBM에 조인한 이후 7년간 국내 솔루션을 DB2 기반으로 마이그레이션 하는 프로젝트에 참여해 왔다. 현재는 IBM Techline 에서 Information Management를 담당하여 DB2뿐 아니라 IBM의 정보 관리 솔루션에 대해 컨설팅 및 기술 지원을 담당하고 있다. 송실대, 고려대, 서울여대 등 국내 주요 대학에서 DB2를 강의해 오고 있다.



Contents

Unit 01 : 제품군 비교

- 서버 제품군 비교
- DB2 서버 제품군 소개
- 클라이언트 제품군 비교
- DB2 클라이언트 제품군 소개

Unit 02 : 아키텍처 비교

- 프로세스 비교
- 메모리 비교
- 물리적 구조 비교

Unit 03 : DB2 제품 설치 및 구성

- DB2 서버 설치
- DB2 클라이언트 설치
- DB2 제품 제거

Unit 04 : 용어 및 클라이언트 툴 비교

- 용어 비교
- Command Line 툴 비교
- DB2 명령행 처리기와 명령 편집기 소개

Unit 05 : 인스턴스 및 데이터베이스 비교

- 인스턴스 비교
- DB2 인스턴스 관리
- 데이터베이스 비교
- 파라미터 비교
- DB2 환경 변수 및 파라미터 설정

Unit 06 : 네트워크 설정 비교

- 서버 네트워크 설정 비교
- 원격 데이터베이스 설정 비교
- DB2 네트워크 설정



Unit 07 : 테이블 스페이스 비교

- 물리적 저장 모델 비교
- 테이블 스페이스 관리 비교
- DB2 테이블 스페이스 생성 및 관리

Unit 08 : 오브젝트 비교

- 스키마, 테이블 및 뷰 비교
- 인덱스 비교
- 시퀀스 비교
- 트리거 비교
- 참수 비교
- DB2 9.5에서 제공하는 오라클 기능
- PL/SQL과 SQLPL 비교

Unit 09 : Security 비교

- 권한 및 특권 비교
- Role 비교
- DB2 보안 모델

Unit 10 : 데이터 이동 비교

- 데이터 이동 방법 비교
- EXPORT/IMPORT/LOAD 유틸리티 비교
- DB2 EXPORT / MPORT /LOAD 소개

Unit 11 : 백업 및 복구 비교

- 로깅 방식 비교
- DB2의 로깅 방식 설정 방법
- 백업 비교
- DB2 백업 소개
- 복구 비교
- DB2 복구 소개

Unit 12 : 모니터링 비교

- 진단 및 Trace 비교
- 모니터링 방법 비교
- DB2 모니터링 소개
- DB2 스냅샷 모니터링 소개
- DB2 이벤트 모니터링 소개
- DB2의 db2pd 툴 소개
- DB2 활동 모니터 소개





Unit 01 : 제품군 비교

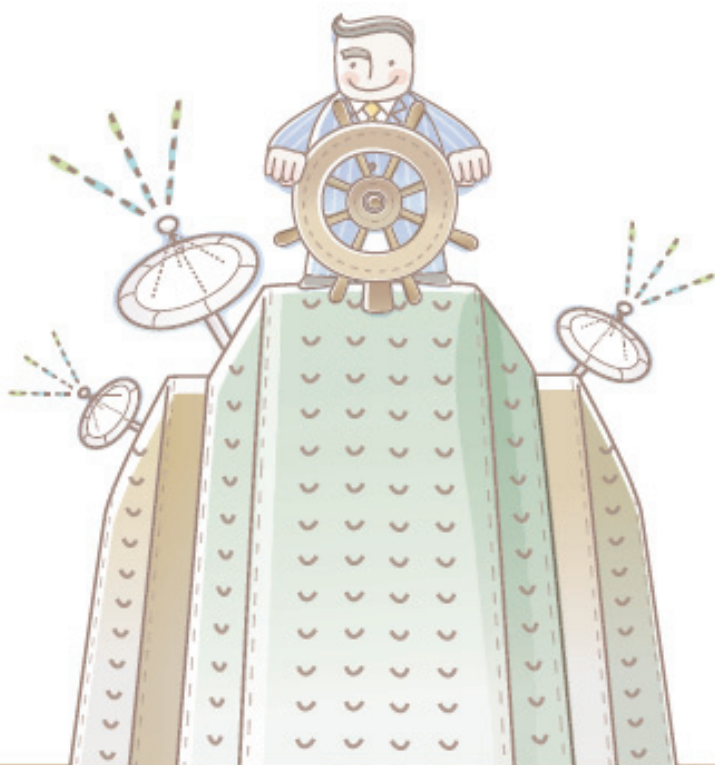
오라클 사용자를 위한 DB2 가이드


서버 제품군 비교

DB2 서버 제품군 소개

클라이언트 제품군 비교

DB2 클라이언트 제품군 소개



Point  아래 표는 오라클 서버 제품군과 DB2 서버 제품군을 비교한 표입니다. 지원 Platform 및 시스템 사양에 따라 지원되는 제품이 달라집니다.

대용량 처리를 위한 데이터베이스

	Oracle Enterprise Edition 11g	DB2 Enterprise Server Edition V9.5(ESE)
지원 운영체제	<ul style="list-style-type: none"> Windows, Linux, AIX, HP-UX, HP Compaq, Solaris, Tru64 	<ul style="list-style-type: none"> Windows, Linux, AIX, HP-UX, Solaris, Linux/390
CPU 및 RAM 제한	<ul style="list-style-type: none"> CPU 수 및 메모리에 제한이 없습니다. 	<ul style="list-style-type: none"> CPU 수 및 메모리에 제한이 없습니다.
특성	<ul style="list-style-type: none"> 플래시백 테이블, 데이터베이스 및 트래잭션 쿼리기능 및 Automatic Workload Management을 지원합니다. 	<ul style="list-style-type: none"> High Availability 기능 및 Workload Management 기능, Table Partitioning Feature가 포함되어 있습니다.

중소형 기업을 위한 데이터베이스

	Standard Edition (SE) / Standard Edition One (SE1) 11g	DB2 Workgroup Server Edition V9.5 (WSE)
지원 운영체제	<ul style="list-style-type: none"> Windows, Linux, AIX, HP-UX, HP Compaq, Solaris, Tru64 	<ul style="list-style-type: none"> Windows, Linux, AIX, HP-UX, Solaris, Linux/390
CPU 및 RAM 제한	<ul style="list-style-type: none"> SE일 경우 4 CPU까지 지원됩니다. SE1일 경우 2 CPU까지 지원됩니다. 	<ul style="list-style-type: none"> 4 CPU에 메모리는 16GB까지 지원됩니다. .
DataSize 제한	<ul style="list-style-type: none"> SE일 경우 데이터 크기에 제한이 없습니다. SE1일 경우 데이터 크기가 500GB 로 제한됩니다. 	<ul style="list-style-type: none"> 데이터 크기에 제한이 없습니다.
특성	<ul style="list-style-type: none"> Oracle RAC 기능이 존재하지 않습니다. 	<ul style="list-style-type: none"> High Availability 기능이 포함되어 있습니다 .

개발자 및 솔루션 업체를 위한 데이터베이스

	Oracle Express Edition(XE) 11g	DB2 Express Edition / DB2 Express -C V9 (EXP)
지원 운영체제	<ul style="list-style-type: none"> Windows, Linux 	<ul style="list-style-type: none"> Windows, Linux
CPU 및 RAM 제한	<ul style="list-style-type: none"> 1 CPU에 메모리는 1GB까지 지원됩니다. 	<ul style="list-style-type: none"> 2 CPU에 메모리는 4GB까지 지원됩니다.
DataSize 제한	<ul style="list-style-type: none"> 데이터 크기가 4GB로 제한됩니다. 	<ul style="list-style-type: none"> 데이터 크기에 제한이 없습니다.
특징	<ul style="list-style-type: none"> 무료로 개발/구축/배포 가능하나 CPU 및 메모리, 그리고 데이터 사이즈에 엄격히 제한되어 있습니다. 64비트는 지원되지 않습니다. PL/SQL만 지원되며 JAVA 및 Native Compilation이 지원되지 않아 Application 개발이 제한적입니다. 	<ul style="list-style-type: none"> DB2 Express-C 의 경우 DB2 Express Edition과 동일하나 추가 옵션을 구매하실 수 없습니다. 무료로 개발/구축/배포할 수 있습니다.

02 DB2 서버 제품군 소개

Point DB2는 Express Edition, Workgroup Server Edition, Enterprise Server Edition 와 같은 서버 제품군이 존재합니다. 모든 서버 제품군은 동일한 공통 코드 기반을 사용하였으므로 애플리케이션의 코드 변경없이 단일 프로세서 서버에서 다중 프로세서 서버 클러스터로 확장할 수 있습니다.

지원 여부	DB2 Express-C	DB2 Express (EXP)	Workgroup Server Edition (WSE)	Enterprise Server Edition (ESE)
지원 운영체제	Linux, Windows	Linux, Windows	Linux, Windows, AIX, HP-UX, Solaris	Linux, Windows, AIX, HP-UX, Solaris
메모리 제한	4GB까지 지원	4GB까지 지원	16GB까지 지원	무제한
32/64 bit 지원여부	32/64 bit 지원	32/64 bit 지원	32/64 bit 지원	32/64 bit 지원
CPU 제한	2 CPU까지 지원	2 CPU까지 지원	4 CPU까지 지원	무제한
DB Size 제한	무제한	무제한	무제한	무제한



DB2 Express-C는 무료로 다운로드 받아 사용할 수 있습니다. <http://www.ibm.com/db2/express> 에서 DB2 Express 제품 설치 이미지 및 자료를 다운로드 받아 사용하기 바랍니다. 또한 <http://www-306.ibm.com/software/data/db2/9/download.html> 에서는 클라이언트, 자료 등을 다운로드 받을 수 있습니다.

AS/400과 S/390 에서 지원되는 DB2 제품군은 호스트 DB2라고 합니다.

제품	설명
DB2 iSeries	i5 OS 에서 지원됩니다.
DB2 zSeries	zOS 에서 지원됩니다.

03 DB2 9.1 서버 제품군 특성

Point 모든 DB2 제품군이 동일한 엔진을 탑재하였으나 성능, 확장성, 가용성 및 보안을 추가적으로 보완하기 위해서는 추가옵션이 지원되는 제품을 선택하거나 별도 구매 옵션을 통해 구현할 수 있습니다. 아래 특성을 살펴본 후 비즈니스 상황에 맞는 제품을 선택합니다.

주요 기능 요약	Express-C FTL	Express	WSE	ESE
Table Partitioning	지원 안함	지원 안함	지원안함	지원
Database Partitioning	지원 안함	지원 안함	지원 안함	Database Partitioning Feature 별도 구매
Connection Concentrator	지원 안함	Workload Management Feature for Express 별도 구매	Workload Management Feature for Workgroup 별도 구매	지원
Governor	지원 안함			지원
Query Patroller	지원 안함			Performance Optimization Feature for Enterprise 별도 구매
Performance Expert	Performance Expert 제품 구매	Performance Expert 제품 구매	Performance Expert 제품 구매	
MQT	지원 안함	Performance Optimization Feature for Express 별도 구매	Performance Optimization Feature for Workgroup 별도 구매	지원
MDC	지원 안함			지원
Query Parallelism	지원 안함			지원
Compression: Row Level	지원 안함	지원 안함	지원 안함	Storage Optimization Feature 별도 구매
Compression : Backup	지원 안함	지원 안함	지원 안함	
Label Based Access Control	지원 안함	지원 안함	지원 안함	Advanced Access Control Feature 별도 구매
Geodetic Extender	지원 안함	지원 안함	지원 안함	Geodetic Data Management Feature 옵션 별도구매
High Availability Disaster Recovery	지원	High Availability Feature for Express 옵션 별도 구매	High Availability Feature for Workgroup 옵션 별도 구매	지원
Online Reorganization	지원 안함			지원
Tivoli System Automation (TSA)	지원			지원
PureXML Storage	지원	PureXML Feature for Express 별도 구매	PureXML Feature for Workgroup 별도 구매	PureXML Feature for Enterprise 별도 구매
Homogeneous Replication	지원			
Heterogeneous and Q Replication	지원 안함	WebSphere Replication Server 제품 별도 구매		
Homogeneous Federation	지원 안함	IBM Homogeneous Federation Feature for DB2 database 별도 구매		
Heterogeneous Federation	지원 안함	WebSphere Federation Server 제품 별도 구매		


04 DB2 9.5 서버 제품군 특성

Point 모든 DB2 제품군이 동일한 엔진을 탑재하였으나 성능, 확장성, 가용성 및 보안을 추가적으로 보완하기 위해서는 추가옵션이 지원되는 제품을 선택하거나 별도 구매 옵션을 통해 구현할 수 있습니다. 아래 특성을 살펴본 후 비즈니스 상황에 맞는 제품을 선택합니다.

주요 기능 요약	Express-C FTL	Express	WSE	ESE
Homogenous SQL Replication	지원	지원	지원	지원
Homogenous Q Replication	지원 안함	지원 안함	지원 안함	Homogeneous Replication Feature for ESE
High Availability Disaster Recovery	지원	High Availability Feature	지원	지원
Tivoli System Automation	지원		지원	지원
Online Reorg	지원 안함		지원	지원
Advanced Copy Services	지원 안함		지원	지원
MQT	지원 안함	지원 안함	Query Optimization Feature	지원
MDC	지원 안함	지원 안함		지원
Query Parallelism	지원 안함	지원 안함		지원
Connect Concentrator	지원 안함	지원 안함		지원
Table Partitioning	지원 안함	지원 안함	지원 안함	지원
Governor	지원 안함	지원 안함	지원 안함	지원
Homogenous Federation	지원 안함	Homogeneous Federation Feature for Express	Homogeneous Federation Feature for WSE	Homogeneous Federation Feature for ESE
Database Partitioning	지원 안함 <small>참고1</small>	지원 안함 <small>참고1</small>	지원 안함 <small>참고1</small>	지원 안함 <small>참고1</small>
Compression : Row Level	지원 안함	지원 안함	지원 안함	Storage Optimization Feature
Compression : Backup	지원 안함	지원 안함	지원 안함	
Label Based Access Control	지원 안함	지원 안함	지원 안함	Advanced Access Control Feature
Geodetic Extender	지원 안함	지원 안함	지원 안함	Geodetic Data Management Feature
Query Patroller	지원 안함	지원 안함	지원 안함	Performance Optimization Feature
Workload Management	지원 안함	지원 안함	지원 안함	
Performance Expert	Performance Expert 제품 구매	Performance Expert 제품 구매	Performance Expert 제품 구매	
PureXML Storage	지원	pureXML Feature for Express	pureXML Feature for WSE	pureXML Feature for ESE
Net Search Extender	지원	지원	지원	지원
Spatial Extender	지원	지원	지원	지원


참고 1)

Data Partitioning Feature (DPF)는 DB2 Warehouse Edition Version 9.5를 통해서만 가능합니다. 기존 DPF 라이선스는 자동으로 IBM Base Warehouse Edition Feature for DB2 Version 9.5로 업그레이드 됩니다.

Point  오라클과 DB2가 지원하는 특성을 비교합니다. 특성별로 서버에 포함된 특성도 있으며 별도 옵션 또는 별도 제품을 통해 지원되는 특성도 있습니다. 비교하는 특성은 DBMS 아키텍처에 기반하는 기능이므로 유사한 기능일 뿐 동일하지는 않습니다.

	오라클 11g	IBM DB2 V9.X	설명
고가용성	<ul style="list-style-type: none"> Oracle RAC 	<ul style="list-style-type: none"> HADR은 Enterprise Server Edition과 Workgroup Server Edition(DB2 9.5 버전부터)에 포함되어 있습니다. 	<ul style="list-style-type: none"> 시스템 장애시 중단 없이 DB Server가 실행될 수 있도록 하는 기능입니다.
	<ul style="list-style-type: none"> Oracle Partitioning 	<ul style="list-style-type: none"> Data Partitioning은 DB2 Enterprise Server에 포함되어 있습니다. 	<ul style="list-style-type: none"> DB2는 데이터 파티셔닝 기능 뿐 아니라 데이터베이스 파티셔닝 기능을 제공하고 있습니다. 데이터베이스 파티셔닝 기능은 유료 옵션입니다.
	<ul style="list-style-type: none"> Data Guard 	<ul style="list-style-type: none"> HADR에 포함된 기능입니다. WebSphere Replication Server (이기종 DBMS 복제 솔루션) 	<ul style="list-style-type: none"> 데이터 복제 솔루션으로 IBM WebSphere Replication Server가 있습니다. 이 제품은 이기종간의 DBMS 복제도 지원합니다.
확장성	<ul style="list-style-type: none"> Oracle RAC 	<ul style="list-style-type: none"> 데이터베이스 파티셔닝 Feature 	<ul style="list-style-type: none"> 오라클은 Shared Disk 구조이며 DB2는 Shared Nothing 구조로 확장하는데 오라클은 제약점이 발생합니다.
보안성	<ul style="list-style-type: none"> Oracle Fine-Grained Auditing 	<ul style="list-style-type: none"> DB2 Audit Utility 	<ul style="list-style-type: none"> Audit 정책을 Business 상황에 맞게 구성할 수 있습니다.
	<ul style="list-style-type: none"> Oracle Label Security 	<ul style="list-style-type: none"> Label Based Access Control 	<ul style="list-style-type: none"> Row Level 보안을 제공합니다.
관리 편의성	<ul style="list-style-type: none"> Enterprise Manager 	<ul style="list-style-type: none"> 제어센터 	<ul style="list-style-type: none"> DBMS 관계 작업을 통합 콘솔을 통해 관리할 수 있습니다.
	<ul style="list-style-type: none"> Oracle Diagnostic Pack 	<ul style="list-style-type: none"> DB2 Engine에 포함됨. 	<ul style="list-style-type: none"> 오라클은 별도 옵션으로 Oracle Diagnostic Pack을 구매해야 되나 DB2는 유사 기능이 엔진에 포함되어 있습니다.
	<ul style="list-style-type: none"> Oracle Tuning Pack (Diagnostic Pack을 구매해야 함) 	<ul style="list-style-type: none"> DB2 Engine에 포함됨. 	<ul style="list-style-type: none"> 오라클은 별도 옵션으로 Oracle Tuning Pack을 구매해야 되나 DB2는 유사 기능이 엔진에 포함되어 있습니다.

06 클라이언트 제품군 비교

Point  아래 표는 오라클 클라이언트 제품과 DB2 클라이언트 제품을 비교한 표입니다. 두 제품 모두 애플리케이션 개발을 위한 JDBC, ODBC, API가 포함되어 있으며 설치 타입에 따라 관리를 위한 툴도 제공 됩니다.

클라이언트 제품군


오라클 11g	IBM DB2 V9.5	설명
<ul style="list-style-type: none"> 오라클 Client 11g 	<ul style="list-style-type: none"> IBM Data Server Runtime Client v9.5 IBM Data Server Client v9.5 	<ul style="list-style-type: none"> DB2는 오라클과는 달리 DB2 Client와 DB2 Runtime Client 두개의 클라이언트 제품군이 존재합니다. 오라클 Client의 Runtime 설치 타입은 DB2의 RunTime Client 제품에 해당합니다. 오라클 Client의 Administrator, Custom 설치 타입에 해당하는 제품은 DB2 Client 제품입니다. DB2 Client 제품을 설치할 때 "일반 설치", "최소 설치", "사용자 설치" 로 설치 타입을 지정하실 수 있습니다.
<ul style="list-style-type: none"> 오라클 Transparent Gateway 	<ul style="list-style-type: none"> DB2 Connect Enterprise Edition DB2 Connect Personal Edition 	<ul style="list-style-type: none"> Mainframe에 존재하는 DB2에 접속하기 위한 Client 제품입니다.

관리 툴 및 개발 툴

오라클 11g	IBM DB2 V9	설명
<ul style="list-style-type: none"> Oracle Enterprise Manager 	<ul style="list-style-type: none"> DB2 Control Center 	<ul style="list-style-type: none"> 데이터베이스 및 인스턴스를 구성하고 관리합니다.
<ul style="list-style-type: none"> Oracle SQL Plus Worksheet iSQL 	<ul style="list-style-type: none"> DB2 명령 편집기 	<ul style="list-style-type: none"> GUI 기반으로 SQL문 혹은 스크립트를 실행할 수 있으며 실행했던 SQL 문을 저장할 수도 있습니다.
<ul style="list-style-type: none"> Oracle SQL Plus 	<ul style="list-style-type: none"> DB2 명령행 (CLP) 	<ul style="list-style-type: none"> OS 명령행에서 실행되는 프로그램으로 SQL 문을 실행할 수 있습니다.
<ul style="list-style-type: none"> Oracle Net Config Assistant 	<ul style="list-style-type: none"> DB2 Configuration Assistant 	<ul style="list-style-type: none"> 네트워크를 설정합니다.
<ul style="list-style-type: none"> Oracle SQL Analyze 	<ul style="list-style-type: none"> DB2 Visual Explain db2exfmt, db2expln 	<ul style="list-style-type: none"> 쿼리를 분석하여 비용 산정 및 Access Plan 등을 알려줍니다.
<ul style="list-style-type: none"> ADDM (Automatic Database Diagnostic Monitor) 	<ul style="list-style-type: none"> DB2 이벤트 분석기 	<ul style="list-style-type: none"> 성능정보를 수집하여 분석합니다.
<ul style="list-style-type: none"> Oracle SQL Developer 	<ul style="list-style-type: none"> Data Server Developer Tool 	<ul style="list-style-type: none"> Stored procedure 및 User Defined Function등 기타 다른 오브젝트를 개발하는 툴입니다.

참고) 관리 툴 및 개발 툴은 서버 제품군 또는 클라이언트 제품군에 포함된 툴입니다.

07 DB2 클라이언트 제품군 소개

Point  Unix/Linux/Windows에 설치된 DB2 서버에 접속하기 위해서는 클라이언트 머신에 아래와 같은 DB2 클라이언트 제품이 설치되어야 합니다. 데이터베이스 서버에 접속하는 목적 및 편의성등을 고려하여 적합한 제품을 선택하여 설치합니다.

DB2 Client 및 Runtime Client 제품을 이용하여 DB2 UDB 서버에 액세스할 수 있습니다.

제 품	설 명
IBM Data Server Runtime Client	<ul style="list-style-type: none"> ▪ ODBC, JDBC 등을 포함한 실행 환경을 제공하는 최소한의 클라이언트 모듈입니다. ▪ DB2 V9 이전까지는 DB2 Runtime Client와 DB2 Run Time Client Lite가 따로 존재하였으나 DB2 V9.1부터 두 제품이 결합되어 Runtime Client로 이름이 변경되었으며 버전 9.5부터 이름이 IBM Data Server Runtime Client로 변경되었습니다.
IBM Data Server Client	<ul style="list-style-type: none"> ▪ Run-Time Client의 기능을 포함하며, 다양한 GUI 도구들을 이용한 관리 작업을 가능하게 하는 클라이언트 모듈과 ESQL 등의 개발이 가능하도록 하는 프리컴파일러, 헤더 파일, 라이브러리 등을 제공하는 클라이언트 모듈입니다 ▪ DB2 V9 이전까지는 DB2 Application Development Client와 DB2 Administration Client 가 따로 존재하였으나 DB2 V9부터는 두 제품이 결합되어 DB2 Client 로 이름이 변경되었으며 다시 V9.5부터는 IBM Data Server Client로 변경되었습니다.

DB2 Connect를 이용하면 OS/390, AS/400 등의 호스트 머신에 있는 데이터베이스를 액세스할 수 있습니다.


제 품	설 명
DB2 Connect Personal Edition	<ul style="list-style-type: none"> ▪ Linux와 Windows 플랫폼에서 지원되는 단일 사용자용 호스트 데이터베이스 액세스 제품입니다.
DB2 Connect Enterprise Edition	<ul style="list-style-type: none"> ▪ Windows, AIX, HP-UX, Solaris, Linux, Linux/390 플랫폼에서 지원되는 다중 사용자용 호스트 데이터베이스 액세스 제품입니다.

Data Server Developer Tool을 통해 DB2 서버 프로그램을 작성할 수 있습니다.


제 품	설 명
Data Server Developer Tool	<ul style="list-style-type: none"> ▪ Data Server Developer Tool은 버전 9.1에서 제공되었던 DB2 Developer Workbench를 대체하는 제품으로 DB2 Developer Workbench에서 제공되었던 기능 뿐 아니라 데이터베이스 디자인, 개발, 전개, 관리 기능까지 더 강화가 되었습니다. ▪ Data Server Developer Tool은 스토어드 프로시저 및 사용자 정의 함수(UDF)의 작성, 편집, 디버깅 및 테스트 뿐 아니라 웹서비스 전개, SQL 및 Xquery 생성 및 실행, Java Application 개발 등을 위한 포괄적인 개발 환경입니다.



<http://www-306.ibm.com/software/data/db2/9/download.html> 에서 DB2 Client 제품을 다운로드 받아 설치할 수 있습니다.

Point  다음은 DB2를 지원하는 툴입니다. 유닉스, 리눅스, 윈도우 및 z/OS 등 다양한 플랫폼을 지원하는 툴입니다.

DB2 Tools	기능 설명
DB2 Web Query Tool	<ul style="list-style-type: none"> • 사용자를 여러 엔터프라이즈 데이터베이스로 동시에 안전하게 연결합니다. <ul style="list-style-type: none"> ▪ 복합 질의, 데이터 비교 및 맞춤형 프리젠테이션을 가능하게 합니다. ▪ 여러 SQL 구문과 매개변수화된 변수, DB2별 SQL과 같은 고급 SQL 및 질의 작성 기능을 제공하고 테이블 및 보기를 포함한 데이터베이스 개체를 조사할 수 있습니다. ▪ 질의 결과를 쉽게 보고 다운로드하여 가져올 수 있으며 다양한 파일 형식으로 결과를 변환할 수 있습니다. ▪ PDA, 무선 전화기 및 문자 호출기와 같은 WAP 기기를 포함한 전자 메일 클라이언트를 통해 비즈니스 정보에 신속하게 전체적으로 액세스할 수 있습니다. ▪ 표준 브라우저를 지원하며 관리자, 개발자 및 일반 사용자가 여러 DB2 플랫폼을 지원하는 질의를 작성, 공유 및 실행하며, 결과를 XML 및 전송 가능한 기타 파일 형식으로 변환할 수 있습니다.
DB2 Recovery Expert	<ul style="list-style-type: none"> • 강력한 진단, 보고 및 자동 복구 기능을 제공하며 Windows, UNIX 및 Linux 플랫폼에서 데이터의 정확성과 높은 가용성을 유지합니다. <ul style="list-style-type: none"> ▪ 시스템이 온라인 상태를 유지하면서 데이터베이스 자산을 유연하게 자동으로 복구할 수 있습니다. ▪ 전문가 또는 초보 DBA가 전체 재해 복구 절차에 의존하지 않고도 데이터베이스 개체를 안전하고 정확하며 신속하게 복구할 수 있습니다. ▪ 정밀 복구 옵션을 제공하여 안전한 데이터베이스 개발 및 유저 관리를 지원합니다. ▪ 테이블, 색인 또는 데이터를 포함하여 변경 또는 누락되거나 부정확한 데이터베이스 자산을 지능적으로 분석 및 진단하는 기능을 제공합니다. ▪ 종종 데이터베이스를 오프라인으로 전환하지 않고도 데이터 자산을 올바른 지정 시점 (point-in-time) 으로 재구성하는 절차를 자동화합니다.
DB2 High Performance Unload	<ul style="list-style-type: none"> • 전사적 이동을 위한 데이터의 업로드와 추출, 또는 적재 적소의 재구성을 위한 빠르고 신속한 기능을 제공하며 높은 수준의 병렬화 및 여러 개의 논리적 노드에 걸친 데이터의 분산 기능을 제공합니다. <ul style="list-style-type: none"> ▪ 순차 테이블 읽기 수행을 통해 IBM DB2 데이터에 신속히 액세스할 수 있습니다. ▪ 최소 비용으로 단일 언로드 중 여러 출력 파일을 만듭니다. ▪ 비 IBM 제품 도구에서도 효과적인 마이그레이션을 지원합니다.
DB2 Performance Expert	<ul style="list-style-type: none"> • DBA가 Linux/Unix/Windows 플랫폼에서 데이터베이스 성능 관리 및 조정을 단순화할 수 있도록 지원합니다. <ul style="list-style-type: none"> ▪ 전문가 분석, 실시간 온라인 모니터, DB2 애플리케이션 및 SQL문을 분석하고 최적화하기 위한 광범위한 보고서를 제공합니다. ▪ 성능 데이터 및 분석 도구 저장을 위한 성능 웨어하우스를 포함합니다. ▪ 분석 기능을 (경험에 의한 방식으로, 질의) 정의하고 적용하여 성능 병목 현상을 확인합니다. ▪ 최적의 처리 능력을 보장하기 위한 시스템 조정에 대한 권장사항을 제안합니다. ▪ 현재 버퍼풀 작업에 대한 정보를 확보하기 위해 데이터를 수집하고 관련 이벤트 활동에 대한 보고서를 제공하는 DB2 Bufferpool을 포함합니다. ▪ 애플리케이션 및 시스템 관련 정보를 포함하여 DB2 Connect 게이트웨이를 모니터링합니다.
기타	<ul style="list-style-type: none"> • 보안을 강화하기 위해 데이터베이스 암호화 솔루션인 DB2 Encryption Expert와 데이터베이스에 대한 변경내역을 관리하는 Change Management Expert 툴이 있습니다.

Point  다음은 Quest사에서 제공하는 DB2 지원 제품입니다.



DB2 Tools	기능 설명
Toad for IBM DB2	<ul style="list-style-type: none"> DB2 UDB에 대해 이해하기 쉬운 그래픽 유저 인터페이스를 제공함으로써 데이터베이스 개발을 단순화하여 개발자 효율성을 극대화 합니다. 또한 전세계적으로 성능이 입증된 바 있는 토드는 개발자들이 표준 툴을 활용하고 있으며 DB2 분야 역시 개발과 오브젝트 관리에 적용되어 생산성 향상을 도모할 수 있습니다.
Quest Central for DB2	<ul style="list-style-type: none"> 리눅스, 유닉스, 윈도우와 z/OS 기반의 DB2를 관리하는 통합 솔루션으로 하나의 콘솔에서 종합적인 데이터베이스 관리 및 SQL 튜닝, 스페이스 관리와 성능 진단이 가능하도록 다양한 플랫폼을 지원합니다.
Spotlight on DB2	<ul style="list-style-type: none"> Spotlight on DB2는 모든 데이터베이스의 현황을 종합적이고 그래픽컬하게 관리자에게 보여줌으로서, 실시간으로 DB2 UDB 와 z/OS 성능 진단 및 문제 발생시 관리자로 하여금 보다 빠르게 대처할 수 있게 합니다.
Knowledge Xpert for DB2	<ul style="list-style-type: none"> z/OS와 윈도우, 유닉스 기반의 DB2 UDB 운영과 관련하여 유저들이 보편적으로 경험하는 공통된 문제점을 해결하는데 도움이 될 수 있도록 전문가의 가이드를 윈도우상에서 포괄적인 기술 지원과 더불어 신속하게 제공합니다.
DataFactory	<ul style="list-style-type: none"> 데이터베이스 개발, 테스트 및 성능을 시뮬레이션할 수 있도록 최대 수백만 개의 유용한 가상 데이터를 제공함으로써 해당 애플리케이션이 실무에 적용되기 전에 성능을 미리 검토할 수 있는 기능을 제공합니다.
Foglight Catridge for DB2/UDB	<ul style="list-style-type: none"> 데이터베이스 및 네트워크를 비롯하여 애플리케이션 서버, 프로세스 스케줄러, 웹서버를 포함한 기업의 중요한 애플리케이션 전체의 요소들을 총괄적으로 모니터링 할 수 있는 강력한 콘솔로써 시스템 문제점을 사전에 진단하여 줍니다. 또한 복잡한 DB2/UDB 데이터베이스 환경을 실시간으로 24시간 모니터링할 수 있는 유용한 솔루션입니다.
End User Management	<ul style="list-style-type: none"> 사용자가 예측 가능한 각종 반응 양식을 시뮬레이션하고 이에 따른 데이터베이스의 성능 측정 및 발생 가능한 문제점을 미리 측정함으로써 서비스 수준을 최적화하여 사용자 입장에서 애플리케이션을 쉽게 사용할 수 있는 기능을 제공합니다.

📍 제품군 관련 Q/A

- 📍 DB2 Workgroup Server Edition을 구매했는데 System FailOver를 위해 대책을 마련하고 싶습니다. 어떤 제품을 구매하면 되는지 그리고 오라클 RAC와 무엇이 다른지 알고 싶습니다.

DB2 9.5에서는 Workgroup Server Edition에 HADR 옵션이 포함되어 있습니다. HADR 기능을 이용할 경우 시스템 장애 및 디스크 장애에 대처할 수 있습니다. 오라클은 Shared Disk 구조로 디스크 장애시 복구 방법이 없기 때문에 디스크 미러링 솔루션 또는 오라클의 dataguard 라는 복제 솔루션이 필요합니다. 하지만 DB2는 HADR 기능안에 로그 Shipping을 통한 복제기능을 제공하므로 시스템 장애 뿐 아니라 디스크 장애에 대해서도 대처할 수 있습니다.

- 📍 DB2에서 제공되는 데이터 압축 기능과 오라클의 데이터 압축 기능은 어떤 차이가 있는지요 ?

오라클은 페이지 레벨에서 데이터 압축을 수행합니다. 각 페이지 헤더에 compression dictionary가 생성되므로 압축률이 떨어집니다. 하지만 DB2는 테이블 레벨에서 데이터 압축을 수행하므로 페이지 레벨보다 압축률이 상당히 좋습니다. 또한 DB2는 insert, import, bulk load 등에 대한 작업중에도 데이터 압축을 지원하지만 오라클은 bulk load에 대해서만 데이터 압축을 수행합니다.

- 📍 DB2 Enterprise Server Edition에서 제공하는 DPF (Database Partitioning Feature)는 어떤 기능인가요?

DPF는 다중 데이터베이스 파티션 기능을 이용하여 병렬 데이터베이스를 구축하는 기능입니다. 즉 여러 시스템에 파티션을 생성하여 데이터베이스를 분리하지만 사용자에게는 논리적인 단일 데이터베이스로 보임으로써 확장성 및 병렬성, 편의성을 제공합니다. DPF는 DB2 9.5부터 IBM Base Warehouse Feature로 명칭이 변경되었습니다.

- 📍 DB2는 가장 좋은 쿼리 실행 계획을 얻기 위해 자동으로 오브젝트에 대한 통계를 수집하는지요? 그리고 오라클은 어떤지요?

DB2는 모든 서버 제품군에서 통계자료를 자동으로 수집합니다. 오라클은 Oracle Enterprise Manager Diagnostic Pack 제품을 별도로 구매하면 이 기능을 이용할 수 있습니다.

- 📍 오라클은 Oracle Label Security 기능을 제공하는데 DB2도 비슷한 기능이 있는지요?

DB2는 Label Based Access Control를 제공합니다. 이 기능은 오라클의 Label Security 기능과 유사합니다. 즉 데이터에 액세스할 수 있는 사용자 보안을 한층 더 보강합니다. DB2는 정부산하기관에 적합한 계층적 구조와 상업 애플리케이션에 적합한 규칙 기반의 구조를 모두 지원하는 반면 오라클은 계층적 구조만을 지원합니다.



Unit 02 : 아키텍처 비교


오리플 사용자를 위한 DB2 가이드

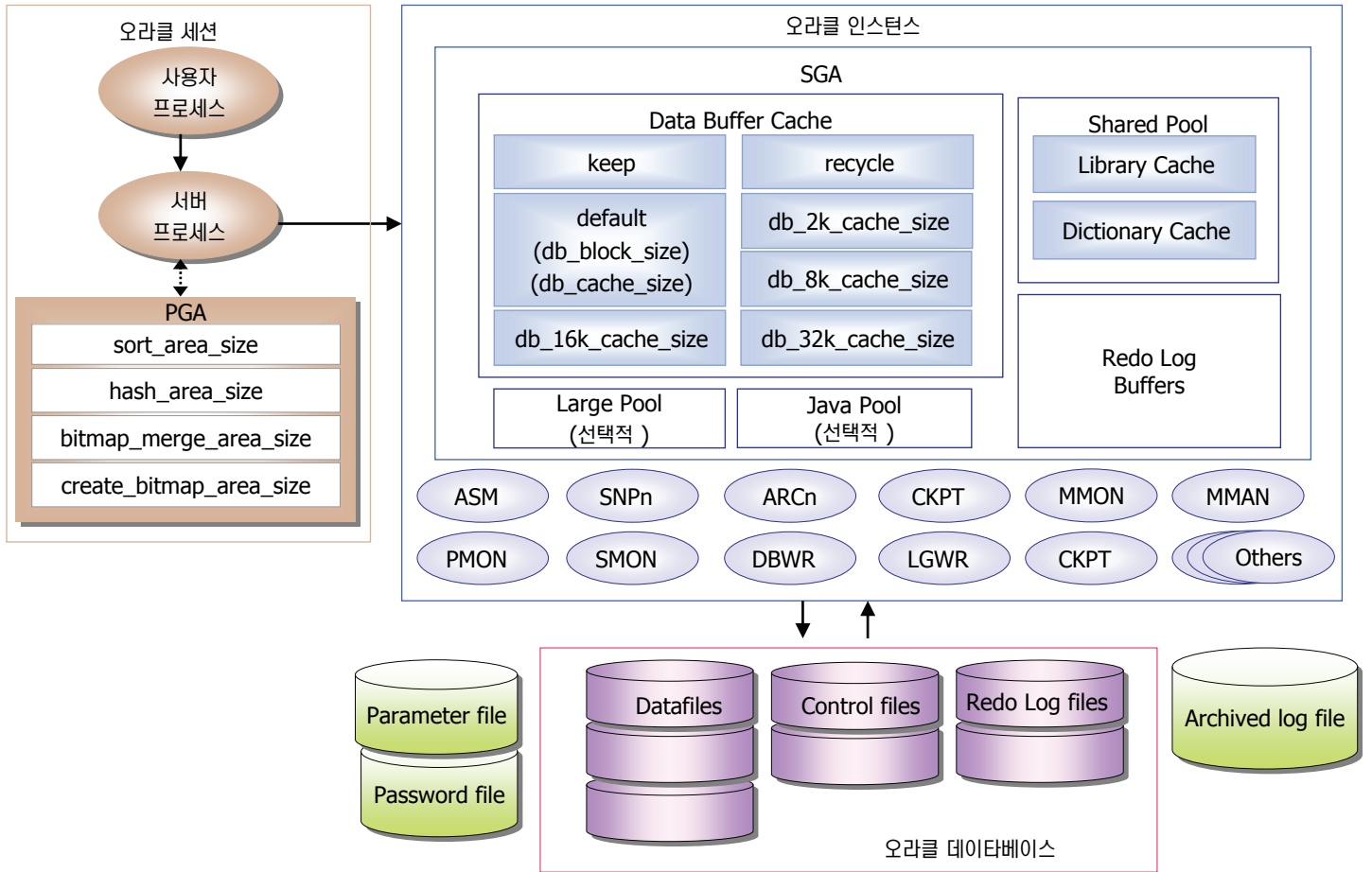
프로세스 비교

메모리 비교

물리적 구조 비교



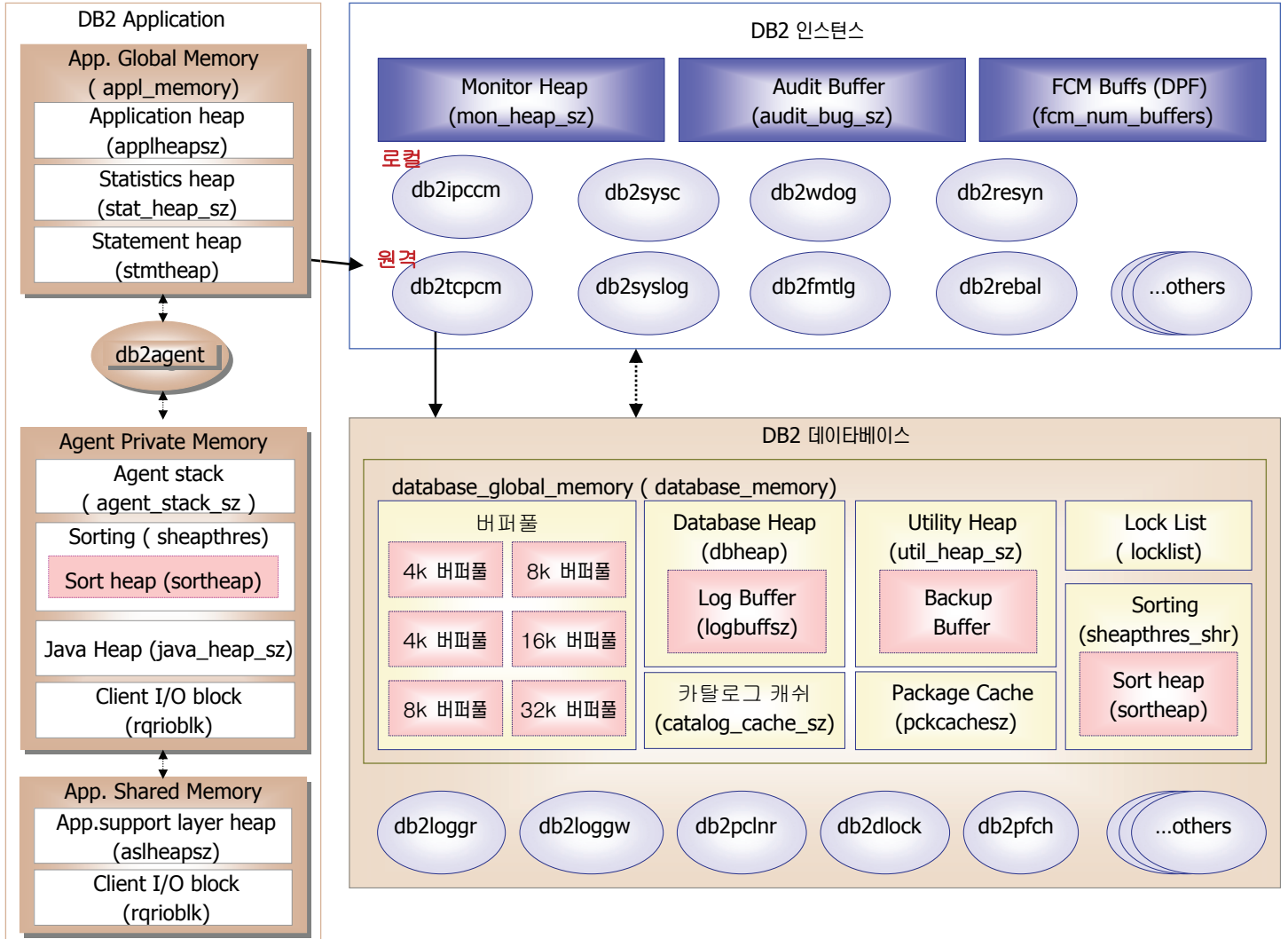
Point  오라클 서버는 파일, 프로세스, 메모리로 구성되며 아래 그림과 같이 오라클 인스턴스와 오라클 데이터베이스로 구성되어 있습니다.



* 부연설명

- 오라클 인스턴스
 - 백그라운드 프로세스와 메모리로 구성되어 있습니다.
 - 인스턴스가 시작될때 System Global Area (SGA)가 할당되며 오라클 백그라운드 프로세스가 시작됩니다.
- 버퍼 캐쉬
 - 오라클의 버퍼 캐쉬는 7개의 버퍼 캐쉬만 존재하며 새로운 버퍼캐쉬를 생성할 수 없습니다.
- 오라클 데이터베이스
 - 데이터베이스 정보를 저장하는 물리적인 저장 공간입니다.
 - 데이터베이스 파일로 구성되어 있으며 이 파일은 OS Level의 파일입니다
 - 오라클 데이터베이스는 오라클 인스턴스 시작시 nomount 옵션이 없을 경우 자동으로 활성화됩니다.


Point DB2 서버는 파일, 프로세스, 메모리로 구성되며 아래 그림과 같이 DB2 인스턴스와 DB2 데이터베이스로 구성되어 있습니다.

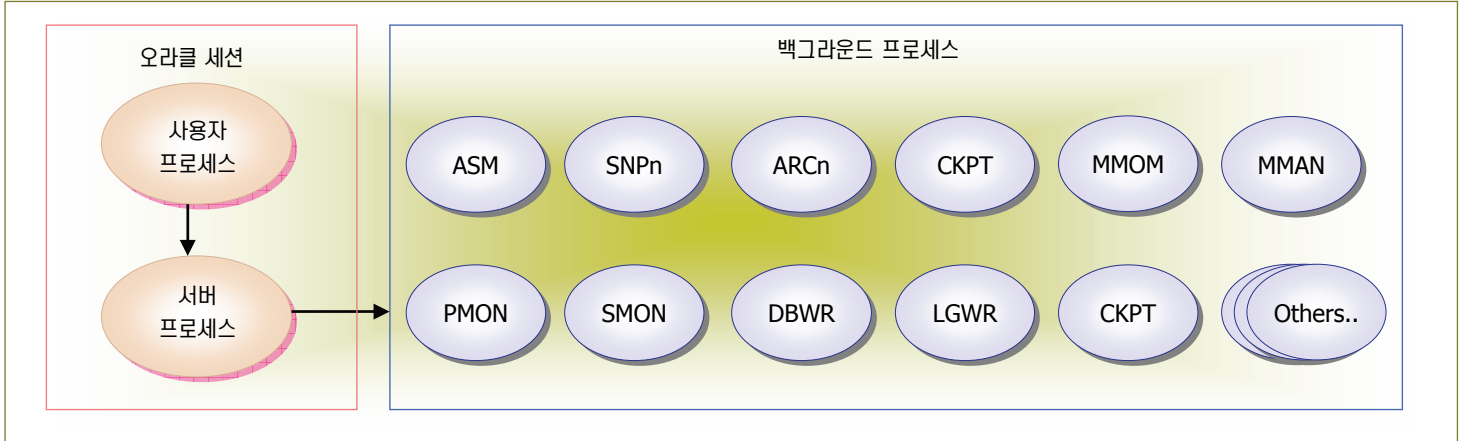


* 부연설명

- DB2 인스턴스
 - DB2 UDB 엔진의 기능을 사용할 수 있는 논리적인 환경입니다.
 - 인스턴스가 시작될때 인스턴스에 필요한 메모리 (예를 들면 Monitor Heap, Audit Buffer,FCM Buffer 등) 및 프로세스가 시작됩니다.
- 버퍼 풀
 - DB2는 오라클과는 달리 필요할때 필요한 사이즈의 버퍼 풀을 생성하여 이용할 수 있습니다.
- DB2 데이터베이스
 - DB2 데이터베이스는 오라클과는 달리 프로세스 및 메모리, 그리고 물리적인 저장 공간인 데이터 파일로 구성됩니다.
 - 인스턴스와 데이터베이스는 별도로 인스턴스를 시작한 후에 별도로 데이터베이스를 기동해야 합니다. DB2 데이터베이스 프로세스 및 메모리가 시작되기 위해서는 "Activate <Database Name>"을 수행하거나 혹은 해당 데이터베이스에 제일 먼저 접속하는 클라이언트가 있을 경우 자동으로 시작됩니다.

03 오라클 프로세스

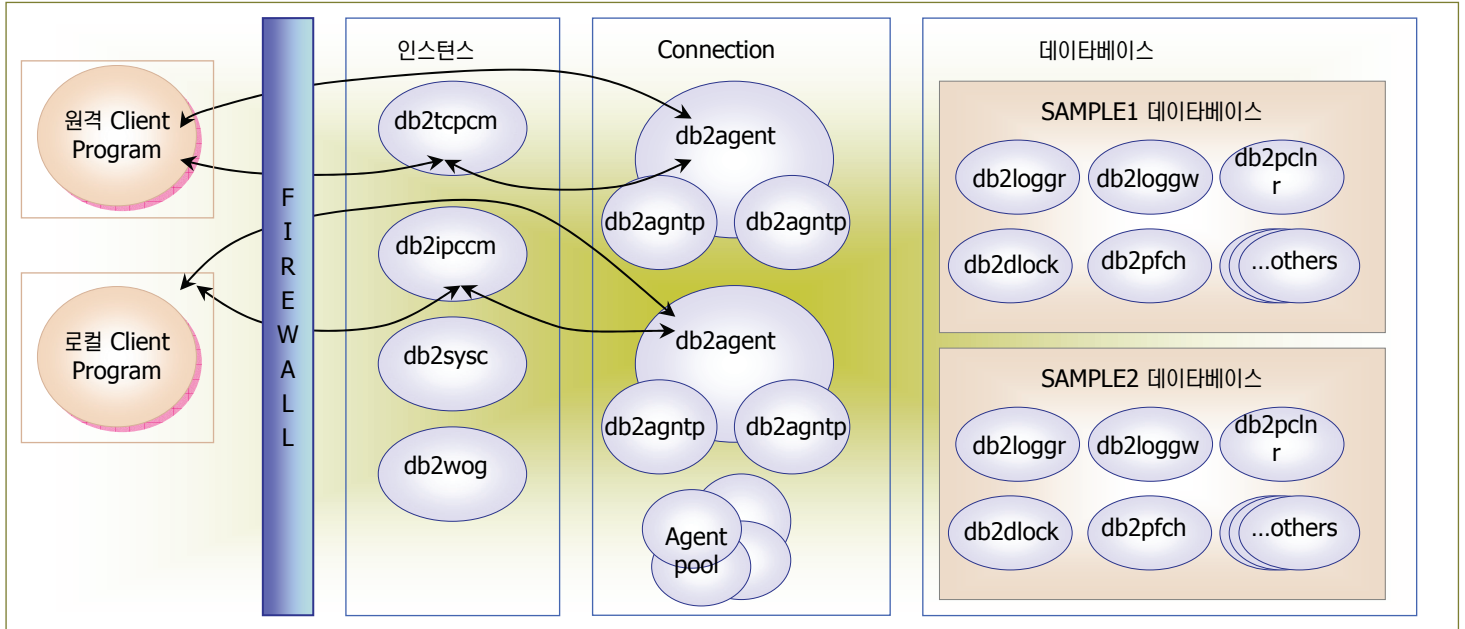
Point  오라클 프로세스는 인스턴스 시작시 옵션 및 파라미터에 따라 프로세스가 구동됩니다. 오라클 프로세스는 크게 사용자 프로세스와 백그라운드 프로세스로 구성됩니다.




레벨	프로세스	설명
N/A	사용자 프로세스	사용자가 실행시킨 SQL문을 서버 프로세스에 전달하고 그 결과를 서버 프로세스로부터 받는 역할을 수행합니다.
	서버 프로세스	사용자 프로세스의 요청 (SQL 문)을 처리하며 전달받은 SQL문을 Parse, Execute, Fetch 작업을 통해 실행합니다.
	LGWR	Log Writer로 SGA에 있는 redo log buffer의 내용을 순차적으로 redo log file로 기록합니다.
	DBWR *	Database Buffer에 있는 Dirty Buffer를 data file에 기록합니다.
	PMON	프로세스 모니터로 사용자 프로세스의 장애 발생시 프로세스 복구를 수행하며 프로세스가 사용하던 자원을 해제합니다.
	SMON	시스템 모니터로 오라클 인스턴스를 관리하는 프로세스입니다. 인스턴스 충돌시 자동으로 인스턴스를 복구하며 시스템 장애 발생시 수행하지 못한 트랜잭션을 복구합니다.
	CKPT	버퍼 캐쉬에 있는 모든 변경 데이터를 데이터 파일에 기록하며 체크 포인트를 기록하기 위해 헤더를 변경함으로써 데이터 파일 및 제어 파일을 수정합니다.
	RECO	분산 데이터베이스에서 네트워크 장애나 시스템 고장으로 처리하지 못한 분산 트랜잭션을 처리합니다.
	ARCH *	온라인 Redo Log File이 꽉차면 이를 지정된 저장소에 저장합니다. 이 프로세스는 Redo Log가 Archive Mode이며 자동 Archiving 기능이 활성화되어 있을때만 존재합니다.
	SNPn *	분산 옵션에서 최대 10개까지의 스냅샷 갱신 프로세스 (SNP0 ~ SNP10)가 자동으로 Table Snapshot을 화면에 표시합니다.
	LCKn *	RAC Mode에서 최대 10개까지의 잠금 프로세스 (LCK0 ~ LCK10)가 인스턴스의 Lock을 관리합니다.
	RVWR	Flashback을 지원하는 프로세스입니다.
	ASM	Storage를 자동으로 관리하는 프로세스입니다.

04 DB2 프로세스

Point DB2 프로세스는 오라클과는 달리 인스턴스 시작시 구동되는 프로세스와 Database Activate시 구동되는 프로세스가 다릅니다. Level 칼럼을 두어 인스턴스 및 데이터베이스시작시 구동되는 프로세스를 구별합니다.



Level	프로세스	프로세스 정의	설명
인스턴스	db2sysc	System controller	DBMS를 제어하는 프로세스로 중요 DB2 서비스 이벤트를 처리합니다. 이 프로세스 없이는 DBMS가 작동되지 않습니다.
	db2wdog	Watchdog	비정상 종료를 처리하는 시스템 감시자입니다. Unix 및 Linux에만 존재합니다
	db2resyn	Re-sync agent	전역 재동기 목록을 스캔하는 재동기 에이전트입니다.
	db2fcms, db2fcmr	FCM daemon	FCM (Fast Communication Manager)의 송수신을 담당하는 Daemon입니다.
	db2fmtlg	DB2 format log	LOGRETAIN 값이 "ON" 이며 USEREXIT 값이 "OFF" 일때 log 디렉토리에 로그 파일을 미리 할당합니다.
	db2ipccm, db2tccm	리스너 프로세스	리스너 프로세스입니다. 클라이언트와의 통신 방법 및 프로토콜 타입에 따라 각 담당 프로세스가 달라집니다. 로컬 클라이언트의 요청일 경우에는 db2ipccm 프로세스가 담당하며 원격 클라이언트의 요청의 경우 db2tccm이 담당합니다.
	db2fmp	Fenced process	분리 Stored Procedure 또는 User Define Function 실행을 담당합니다.
데이터베이스	db2loggr	Log Reader	트랜잭션 처리 및 복구를 처리하는 로그 파일 유지보수를 담당합니다.
	db2loggw	Log Writer	로그 파일에 로그 레코드를 기록합니다.
	db2dlock	Deadlock detection	교착상태를 감지합니다.
	db2pfchr	Pre-fetcher	디스크의 데이터를 버퍼 풀로 기록합니다.
	db2pclr	Page cleaner	버퍼풀 페이지를 디스크에 기록합니다.
애플리케이션	db2agent		클라이언트로부터 들어오는 요청을 처리합니다.


Point  오라클과 DB2의 프로세스를 비교한 표입니다. 비교하는 두 프로세스는 유사한 일을 담당하거나 또는 그 역할을 일부 기능으로 수행하는 프로세스입니다.

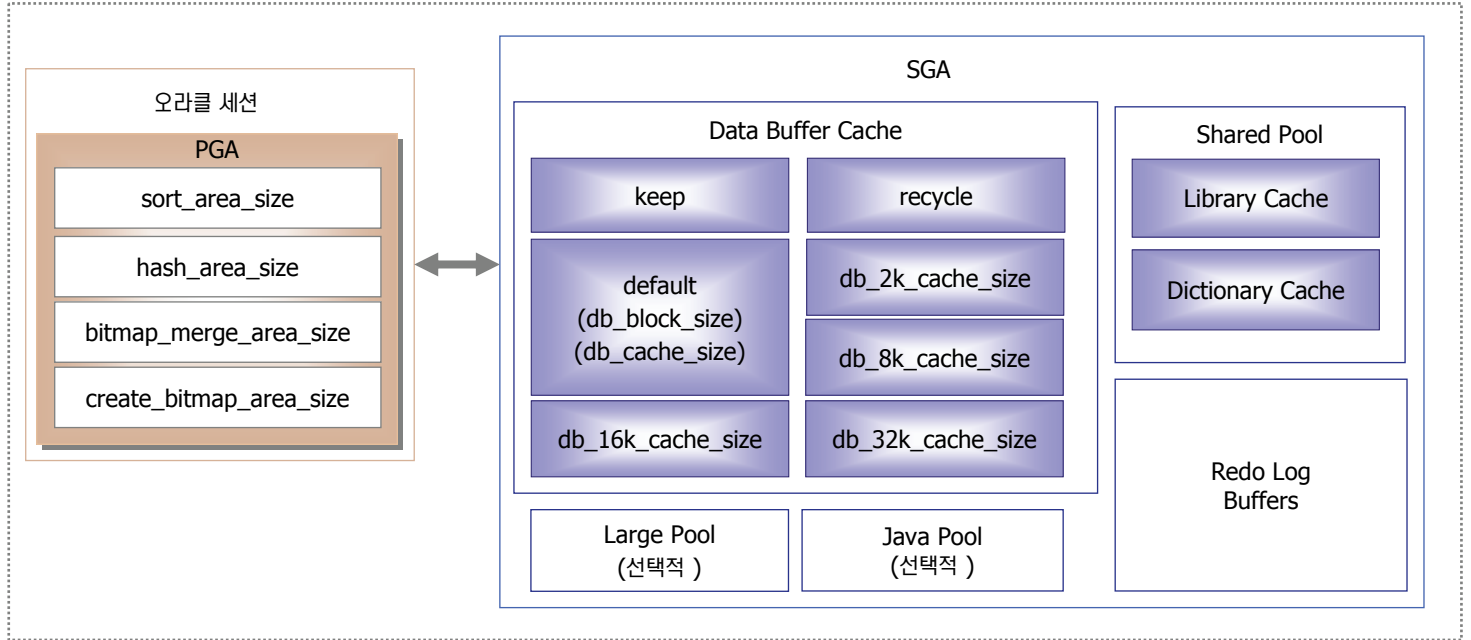
Oracle	DB2
PMON	db2wdog 및 db2loggr
SMON	해당하는 프로세스가 없습니다.
ARCx	db2fmtlg 및 db2cart
Server	db2agent
DBWx	db2pclnr
CKPT	Softmax라는 soft checkpoint를 두어 db2pclnr 프로세스가 버퍼풀의 Dirty page를 디스크에 기록하도록 합니다.
LGWR	db2loggw
해당하는 프로세스가 없습니다.	db2dlock
Server Process	db2pfchr
User Process가 일부 역할 수행	db2ipccm (DB2 Communication Manager)
User Process가 일부 역할 수행	db2tcpcm (DB2 TCP Manager)
Server Process	db2agent (DB2 Coordinating agent)

부연설명

- 오라클과 DB2는 IBM Ted Codd 박사의 "관계형 모델"이라는 동일한 뿌리로 시작되었기 때문에 아키텍처가 유사합니다. 그러나 성능 및 확장성, 자율성 등을 고려하여 발전된 방향이 달라 프로세스 및 메모리, 파일 구조, 옵티마이저 등이 많이 다릅니다. 정확히 모든 기능 및 프로세스 등이 일대일로 매핑되는 것이 아니므로 참조용으로 이해하시기 바랍니다.
- DB2는 위에서 설명한 프로세스 이외에도 많은 프로세스들이 존재하며 구성 파라미터에 따라 생성되는 프로세스도 있습니다. 자세한 사항은 "DB2 온라인 정보 센터" 참조하시기 바랍니다.

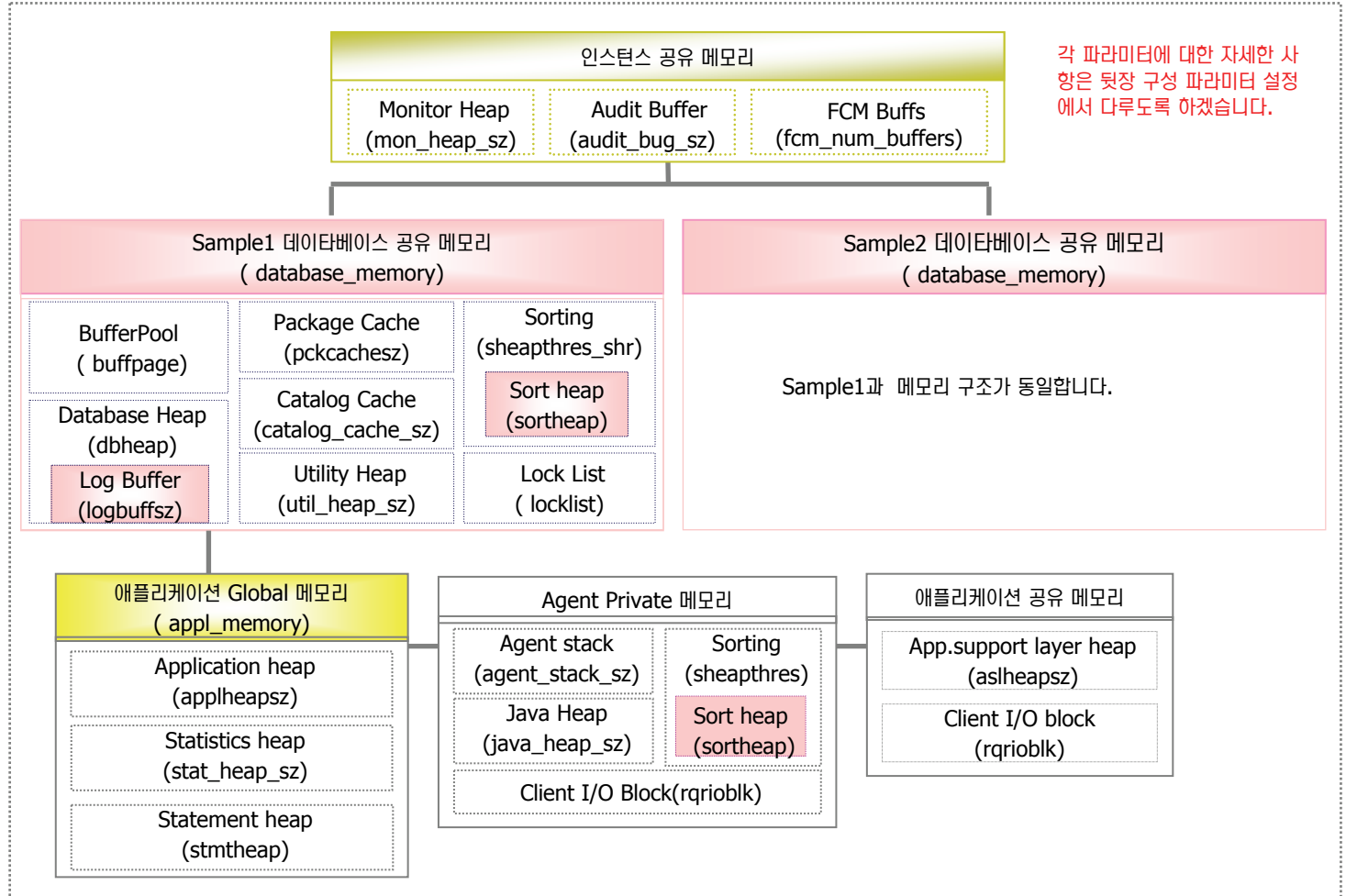
DB2 온라인 정보 센터 : <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp>

Point  오라클의 메모리 구조는 인스턴스 구동시 생성되는 SGA영역과 세션이 연결될 때 생성되는 PGA영역으로 나뉩니다.




Oracle	설명
SGA	<ul style="list-style-type: none"> 인스턴스 및 데이터베이스의 제어 정보를 가지는 공유 메모리 영역입니다. SGA_MAX_SIZE 파라미터로 사이즈에 제한을 둘 수 있습니다.
Buffer Cache	<ul style="list-style-type: none"> 디스크 I/O를 줄이기 위해 테이블 공간에 있는 데이터 Block을 저장하는 메모리 공간입니다. DB_BLOCK_SIZE로 초기 block 사이즈를 지정합니다.
Shared Pool	<ul style="list-style-type: none"> Library Cache와 Dictionary Cache로 구성됩니다. SHARED_POOL_SIZE로 사이즈에 제한을 둘 수 있습니다. <ul style="list-style-type: none"> Library Cache : 가장 최근에 사용된 SQL 및 PL/SQL 문에 대한 정보가 저장됩니다. Dictionary Cache : 데이터베이스의 Meta Data 정보를 저장하는 영역입니다.
Redo Log Buffer	<ul style="list-style-type: none"> 데이터 Block의 모든 변경에 대해 기록하는 영역으로 복구를 위한 영역입니다. LOG_BUFFER로 사이즈를 지정합니다.
Large Pool.	<ul style="list-style-type: none"> SGA에서 선택적으로 생성가능한 영역으로 Backup, Restore, LOAD시 사용되는 메모리 영역입니다. LARGE_POOL_SIZE 로 사이즈를 지정합니다.
Java Pool	<ul style="list-style-type: none"> SGA영역에서 선택적으로 생성가능한 영역으로 Java 명령어 Parsing시 필요한 영역입니다. JAVA_POOL_SIZE 로 사이즈를 지정합니다.
PGA	<ul style="list-style-type: none"> PGA 메모리는 인스턴스의 구성에 따라 PGA에 저장하는 내용이 달라질 수 있습니다. 일반적으로 PGA는 다음과 같이 두가지 영역으로 나뉩니다. <ul style="list-style-type: none"> Private SQL 영역 : 바인드 정보 및 run-time 메모리 구조에 대한 데이터를 저장합니다. Session 메모리 : 세션 변수 및 그 세션과 관련된 여러가지 정보를 저장합니다.

Point DB2의 메모리 구조는 인스턴스 공유 메모리, 데이터베이스 공유 메모리, 애플리케이션 global 메모리, Agent Private 메모리, 애플리케이션 공유 메모리로 구성됩니다. 각 메모리는 인스턴스 시작시, 데이터베이스 활성화시, 응용프로그램 접속시에 할당됩니다.




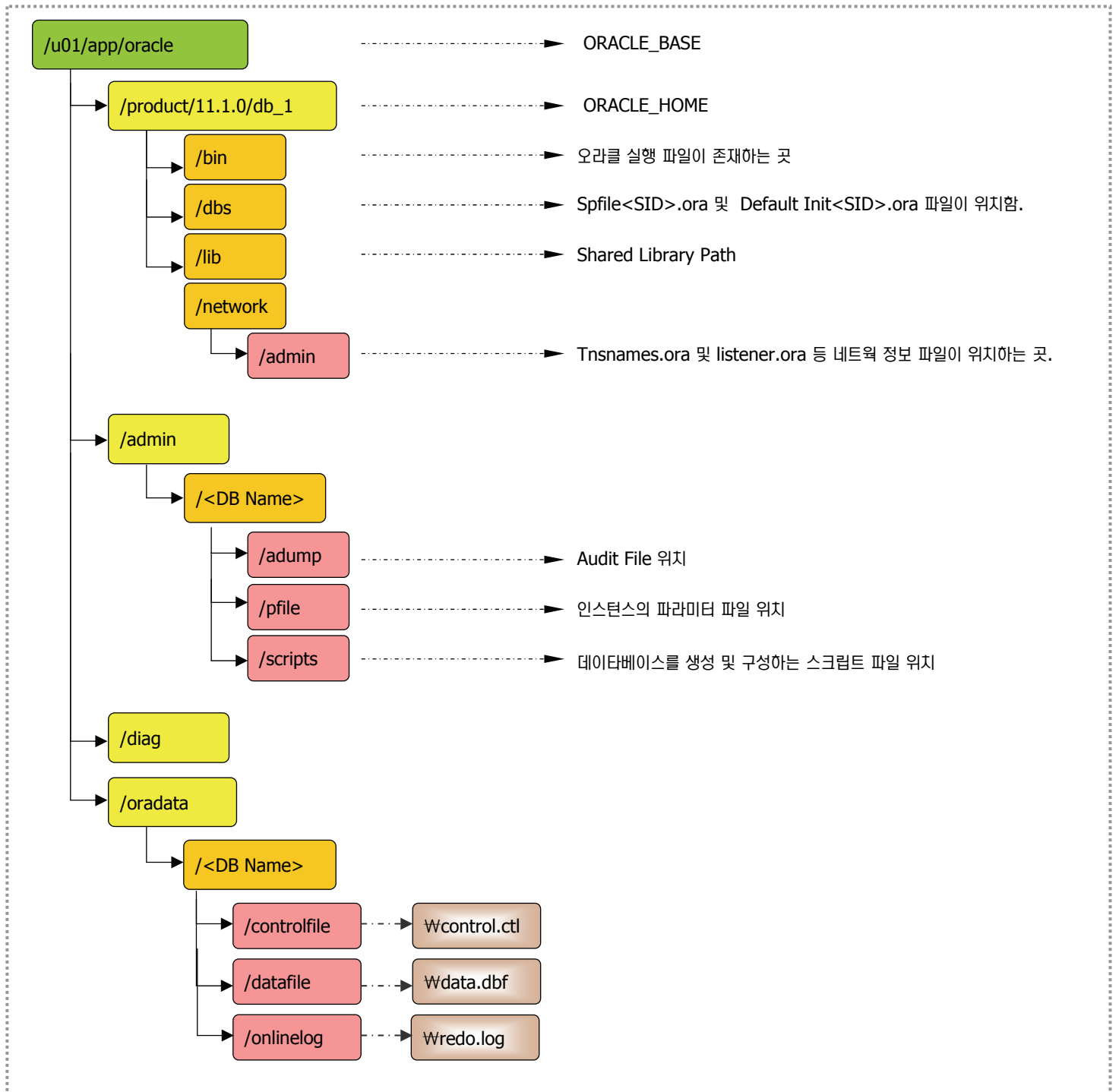
각 파라미터에 대한 자세한 사항은 뒷장 구성 파라미터 설정에서 다루도록 하겠습니다.

메모리	설명
인스턴스 공유 메모리	<ul style="list-style-type: none"> 인스턴스당 한 개의 공유 메모리가 생성되며 인스턴스 시작시 할당되며 중지할 경우 해제됩니다. 인스턴스 관리를 위해 할당된 공간입니다. instance_memory 구성 파라미터의 값을 지정할 경우 사이즈를 지정할 수 있으며 "Automatic" 설정시 DB2가 자동으로 현재 필요한 메모리 값을 설정합니다. Default 로는 "Automatic" 값으로 설정되어 있습니다.
데이터베이스 공유 메모리	<ul style="list-style-type: none"> 데이터베이스당 한 개의 데이터베이스 공유 메모리가 생성됩니다. 이 영역은 데이터베이스가 활성화 되거나 처음으로 연결될 때 할당되며 데이터베이스가 비활성화 되거나 마지막으로 연결이 해제될 때 할당이 해제됩니다. 백업/복구, 잠금, SQL 실행과 같은 일을 수행할 때 사용되는 영역입니다.
애플리케이션 Global 메모리	<ul style="list-style-type: none"> 응용프로그램의 요청을 처리하기 위해 Database Agent에 의해 할당되는 메모리입니다. stmtheap, stat_heap_sz, applheapsz로 구성됩니다. <ul style="list-style-type: none"> ahhpleahsz : DB에 접속한 모든 응용프로그램에 의해 사용되는 메모리입니다. stat_heap_sz : RUNSTATS 명령어를 수행하여 통계정보를 수집할 때 사용되는 메모리입니다. stmtheap : SQL 또는 Xquery를 컴파일에 사용되는 작업 공간입니다.
Agent Private 메모리	<ul style="list-style-type: none"> 클라이언트의 요청을 처리하기 위해 생성되는 Database Agent에 할당되는 영역입니다.
애플리케이션 공유 메모리	<ul style="list-style-type: none"> 데이터베이스에 접속한 응용프로그램간의 통신을 위해 할당되는 버퍼입니다.

Point  오라클과 DB2의 메모리를 비교한 표입니다. 비교하는 두 메모리 구조는 유사한 정보를 저장하는 영역입니다.

Oracle	DB2	설명
SGA	Database Manager 와 Database Shared Memory	<ul style="list-style-type: none"> 인스턴스 및 데이터베이스의 제어 정보를 가지는 공유 메모리 영역입니다.
Buffer Cache	BufferPool	<ul style="list-style-type: none"> 디스크 I/O 를 줄이기 위해 테이블 공간에 있는 데이터 Block을 저장하는 메모리 공간입니다. 오라클 : <ul style="list-style-type: none"> Default 로는 하나만 존재하며 keep, recycle 버퍼 캐시가 존재합니다. 각 페이지 사이즈마다 하나씩 존재합니다. init.ora 버퍼 캐쉬에 대한 파라미터가 정의되어 있습니다. 동적으로 사이즈 변경이 가능합니다. DB2 : <ul style="list-style-type: none"> 필요한 개수만큼 생성이 가능합니다. CREATE BUFFERPOOL ~ 명령어를 수행하여 버퍼풀을 생성합니다. 동적으로 사이즈 변경이 가능합니다.
Redo Log Buffer	Log Buffer	<ul style="list-style-type: none"> 데이터베이스의 데이터 Block의 모든 변경에 대해 기록하는 영역으로 복구를 위한 영역입니다.
Library Cache	Package Cache	<ul style="list-style-type: none"> 실행계획과 실행 코드가 저장되는 영역입니다.
Dictionary Cache	Catalog Cache	<ul style="list-style-type: none"> 데이터베이스의 Meta Data 정보를 저장하는 메모리로 오라클에서는 Data Dictionary 정보를, DB2에서는 Catalog 정보를 저장하는 영역입니다.
PGA	Application Heap 및 Agent Private Memory 일부	<ul style="list-style-type: none"> 애플리케이션이 데이터베이스에 연결시 할당되는 영역입니다. 오라클 : <ul style="list-style-type: none"> 데이터베이스 접속시 각 서버 프로세스에 의해 할당되는 메모리 영역입니다. DB2 : <ul style="list-style-type: none"> 애플리케이션 접속시 application heap이 할당됩니다.
Large Pool	Utilities Heap	<ul style="list-style-type: none"> Backup, Restore, LOAD시 사용되는 메모리 영역입니다.
존재하지 않음	Lock List	<ul style="list-style-type: none"> Lock 정보를 저장하는 메모리 영역입니다.

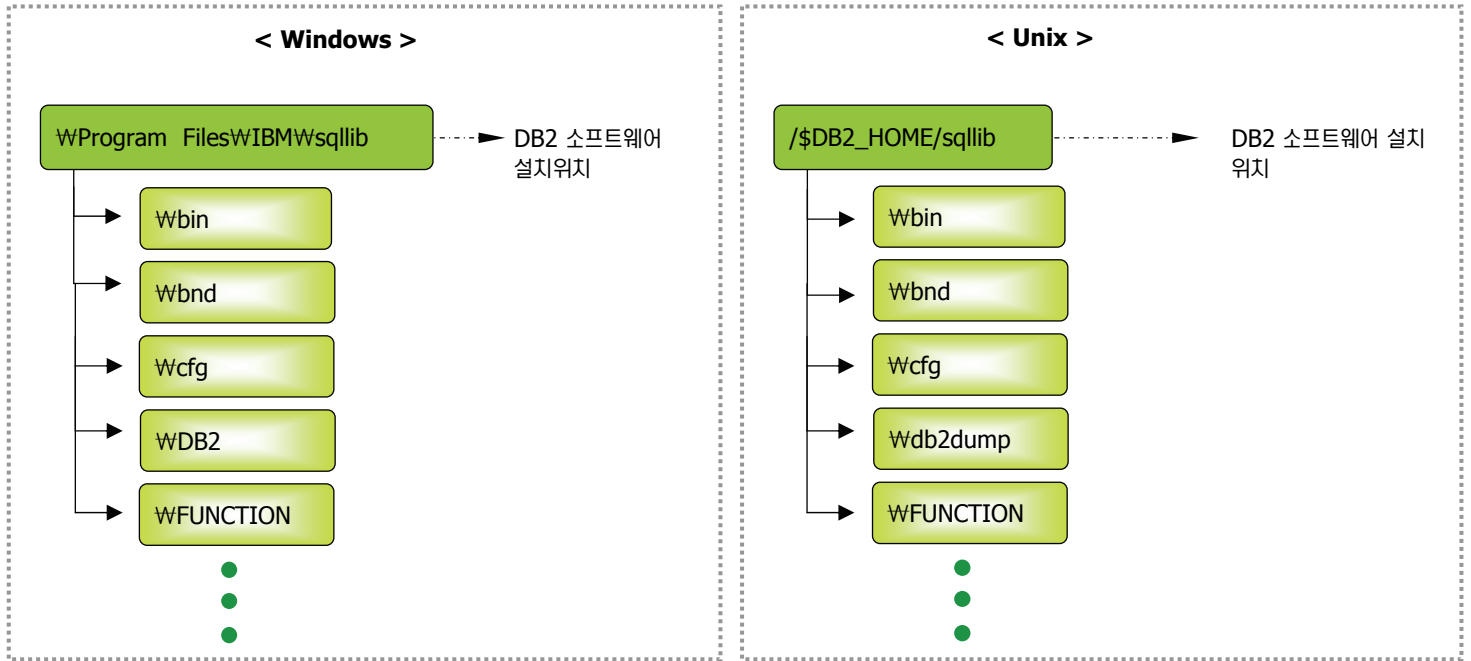
Point  오라클은 Optimal Flexible Architecture (OFA) 기반으로 오라클 제품 설치 및 데이터베이스를 생성하도록 가이드합니다.



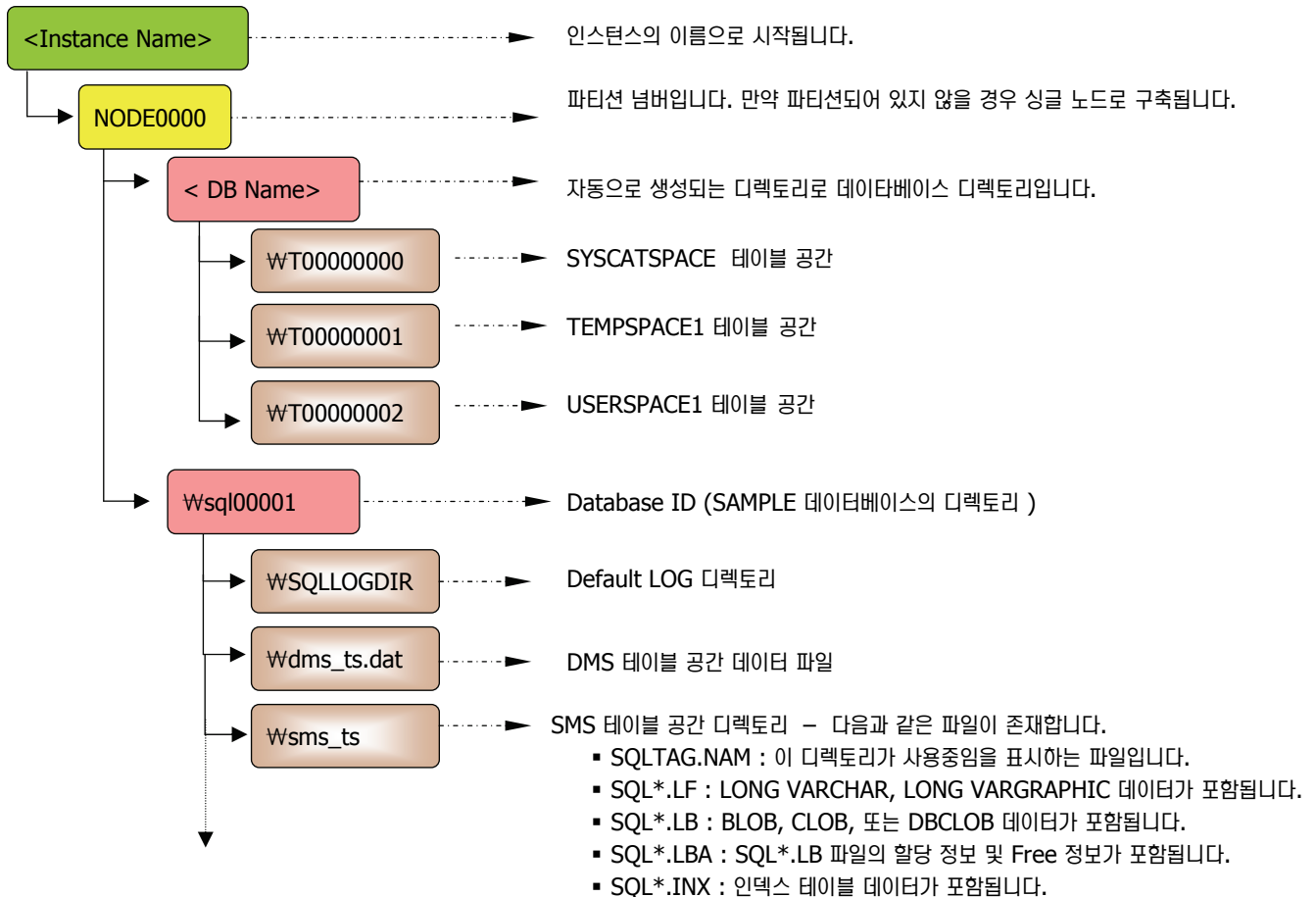
**** ASM 사용시 아래 구조로 생성되지 않습니다. (오라클 11.1.0 기반의 오라클 데이터베이스 위치입니다.)**

10 DB2 물리적 구조

Point 다음은 root 사용자로 DB2를 설치할 경우 DB2 제품이 설치되는 물리적인 위치입니다. 아래 그림은 Windows 와 AIX 에 제품이 설치될 때의 구조입니다. OS 별로 차이가 날 수 있으므로 DB2 V9.5 의 도움말을 참고하시기 바랍니다.



< 데이터베이스의 생성 위치 >



아키텍처 관련 Q/A

- 오라클은 SGANE에 Buffer Cache가 존재합니다. DB2는 그렇지 않은 것 같은데 Buffer Cache와 같은 데이터를 올려놓는 영역을 어디서 어떻게 정의해야 되는 건가요?

DB2는 오라클의 버퍼 캐쉬와 같은 기능을 하는 버퍼 풀이 존재합니다. 그러나 오라클과는 달리 버퍼풀을 필요할 때마다 생성할 수 있으며 처음 데이터베이스 생성시에는 디폴트로 하나의 버퍼풀이 생성됩니다. 향후에 거론되겠지만 테이블 스페이스의 page 크기가 다를 경우 그 page 크기에 해당하는 버퍼풀이 반드시 존재해야 합니다.

- 오라클은 undo 데이터를 저장하는 undo segment (예전에는 rollback segment) 영역이 있어 data의 multi version을 저장하고 동시성을 향상시킵니다. DB2는 undo data를 어느 곳에 저장하는지요? 그리고 data의 multi version을 어떻게 관리하는지요?

DB2는 undo data를 redo log buffer에 저장합니다. 그러나 오라클과는 달리 데이터의 multi version을 관리하지는 않습니다. 그러나 동시성을 위해 여러가지 isolation level을 두어 동시성을 향상시키며 또한 이외에도 동시성을 향상시키기 위해 옵티마이저와 여러가지 변수를 두고 있습니다.

- 오라클은 설치전에 파일 시스템을 생성하여 그 파일 시스템에 오라클 소프트웨어와 데이터베이스를 생성합니다. DB2의 물리적 위치를 살펴 보니 DB2는 이렇게 할 필요가 없는 것 같은데 DB2 소프트웨어는 어디에 설치되며 또한 인스턴스의 소유자가 되는 사용자의 홈 디렉토리에 파일들이 설치가 되는 것 같은데 이것은 무엇인지요?

오라클은 소프트웨어만 설치하는데 거의 10GB가 필요합니다. 이를 위해서는 별도의 파일 시스템이 필요합니다. 하지만 DB2는 소프트웨어만 설치하는데 있어서 650MB 공간만 필요합니다. 옵션에 따라 설치에 필요한 공간이 약간씩 달라지지만 대략 1GB를 넘지 않습니다. 이로 인해 별도의 파일 시스템을 둘 필요는 없습니다. 그러나 별도의 파일시스템을 두어 설치하면 편리합니다. DB2가 설치된 후에는 인스턴스를 생성하게 되면 인스턴스의 소유자의 홈 디렉토리에 DB2를 구동하는데 필요한 여러가지 파일들이 생성됩니다. 이는 대부분 DB2가 설치된 디렉토리의 파일들과 link를 건 것으로 파일 시스템을 차지하지는 않습니다.



Unit 03 : DB2 제품 설치 및 구성

오라클 사용자를 위한 DB2 가이드

DB2 서버 설치

DB2 클라이언트 설치

DB2 제품 제거



01 DB2 V9.5 설치전 준비사항

Point DB2를 설치하기 전에 반드시 시스템의 하드웨어 및 소프트웨어 요구 사항을 확인합니다. 아래 표는 DB2 v9.5의 설치 요구사항이며 DB2 버전이 업그레이드 될 경우 하드웨어 및 소프트웨어의 설치 요구 사항은 변동될 수 있습니다.

시스템 요구사항

시스템 Resource	요구 사항
디스크	제품 설치시 필요한 디스크 공간은 설치 유형 및 파일 시스템 유형에 따라 달라집니다. 그러나 대략 DB2 서버 제품군 설치시 600MB~800MB가 필요하며 데이터베이스에 대한 공간은 별도로 필요합니다. Linux / Unix 운영 체제의 경우 /tmp 디렉토리에 2GB 여유공간을 권장합니다.
메모리	DB2 데이터베이스 시스템에서는 최소한 256MB RAM 이 필요합니다. DB2 및 DB2 GUI 도구만 실행하는 시스템의 경우 최소 512MB RAM 이 필요합니다. 그러나 성능을 위해서는 1GB RAM을 권장합니다.

지원 Platform 및 소프트웨어 설치 요구 사항

운영체제	설치 요구 사항	하드웨어
AIX	<ul style="list-style-type: none"> AIX 5.3 <ul style="list-style-type: none"> 64 비트 AIX 커널 TL5 SP3 최소 C++ runtime Level xIc.rte 8.0.0.8 및 xIc.aix50.rte.8.0.0.8 	<ul style="list-style-type: none"> eServer™ pSeries® IBM® System p™ IBM System p5™
HP-UX	<ul style="list-style-type: none"> HP-UX 11iv2 (11.23.0505) <ul style="list-style-type: none"> 2005년 5월 Base Quality (QPKBASE) 필요 2005년 5월 Applications Quality (QPAPPS) 번들 필요 HP-UX 11iv3 (11.31) 	<ul style="list-style-type: none"> Itanium 기반의 HP® Integrity Series 시스템
Linux	<ul style="list-style-type: none"> Red Hat Enterprise Linux (RHEL) 4 Update 4 Red Hat Enterprise Linux (RHEL) 5 SUSE Linux Enterprise Server (SLES) 9 Service Pack 3 SUSE Linux Enterprise Server (SLES) 10 Service Pack 1 	<ul style="list-style-type: none"> x86 (Intel® Pentium®, Intel Xeon®, 및 AMD) 32-bit Intel 및 AMD 프로세서 x64 (64-bit AMD64 및 Intel EM64T 프로세서) POWER™ (IBM® eServer™ OpenPower™, System i™) 또는 Linux 를 지원하는 pSeries® systems eServer System z™ or System z9™
Solaris	<ul style="list-style-type: none"> Solaris 9 <ul style="list-style-type: none"> 64 비트 커널 패치 111711-12 and 111712-12 필요 Raw 디바이스가 사용될 경우 패치 122300-11 필요 64 비트 Fujitsu PRIMEPOWER 및 솔라리스 9 커널 업데이트 패치 112233-01 또는 그 이후 버전 필요 Solaris 10 <ul style="list-style-type: none"> 64 비트 커널 Raw 디바이스가 사용될 경우 패치 125101-07 필요 	<ul style="list-style-type: none"> UltraSPARC
Window	<ul style="list-style-type: none"> Windows® XP Professional (32 비트 및 x64) <ul style="list-style-type: none"> Windows XP Professional 서비스 팩 2 이상 버전 필요. Windows Vista Ultimate (32 비트 및 x64) Windows Vista Business (32 비트 및 x64) Windows Vista Enterprise (32 비트 및 x64) Windows 2003 Standard Edition (32 비트 및 x64) Windows 2003 Enterprise Edition (32 비트 및 x64) Windows 2003 Datacenter Edition (32 비트 및 x64) 	<ul style="list-style-type: none"> 지원되는 Windows 운영 체제(32비트 및 64비트)를 실행할 수 있는 모든 Intel® 및 AMD 프로세서



시스템 설치 요구사항은 변경될 수 있으므로 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp> 에서 최신의 정보를 참고하시기 바랍니다.

02 DB2 V9.5 설치전 준비사항

Point DB2는 아래와 같이 다양한 설치 방법이 있습니다. 상황에 따라 설치 방법중 하나를 선택하여 설치합니다.

설치 방법 비교

설치 방법	Windows	Linux 또는 Unix	설치 소요 시간
DB2 설치 마법사를 이용한 설치	지원	지원 (X 서버 필요)	약 10 ~ 20분 소요
응답 파일을 이용한 설치	지원	지원	약 5 ~ 10분 소요
db2_install 명령어를 이용한 설치	지원 안함	지원	약 5 ~ 10분 소요

설치 과정 비교

설치 방법	설명
DB2 설치 마법사	<ul style="list-style-type: none"> DB2 설치 마법사는 Linux, Unix 및 Windows에서 사용 가능한 GUI 설치 프로그램입니다. DB2 설치 마법사는 DB2 제품을 설치하고 초기 설정 및 구성 태스크를 수행합니다. DB2 설치 마법사를 통해 인스턴스 생성 및 응답 파일 작성이 가능합니다. 주) Linux / Unix 플랫폼에서 root 사용자가 아닌 다른 사용자가 설치할 경우 DB2 인스턴스 한개만 생성할 수 있습니다. 이는 root 사용자가 아닐 경우 다른 사용자의 Resource를 사용할 권한이 없기 때문입니다.
응답 파일 설치	<ul style="list-style-type: none"> 응답 파일은 설치 설정값이 저장된 파일입니다. DB2 설치시 응답 파일에 저장된 설치 설정값에 따라 설치가 수행됩니다. 응답파일은 DB2 제품 설치를 수행하지 않고도 작성이 가능하며 향후 타 시스템에서 DB2를 설치할 경우 응답파일에 지정한 동일한 설치 설정값으로 DB2제품을 설치할 수 있습니다. Linux 및 Unix 운영체제의 경우 타사 제품의 솔루션에 DB2 설치 이미지를 임베드할 때 응답파일을 이용하여 타사 제품의 솔루션과 함께 DB2를 자동으로 설치할 수 있습니다. 응답 파일 작성 방법 : <ul style="list-style-type: none"> 응답 파일 생성 프로그램을 사용하여 기존의 설치 구성값을 복사하여 타 시스템에 설치할 수 있습니다. DB2 설치 마법사를 이용하여 사용자가 선택한 설정값을 응답파일에 저장할 수 있습니다. DB2 설치 이미지에서 제공되는 샘플 응답 파일을 수정하여 사용할 수 있습니다.
db2_install 명령어	<ul style="list-style-type: none"> 사용자가 선택한 DB2 제품의 모든 구성 요소를 설치하며 DB2 설치 마법사보다 설치 시간이 빠릅니다. -L 옵션을 사용할 경우 지원할 추가 언어를 선택할 수 있습니다. 설치 구성 요소, 즉 최소 설치 또는 사용자 설치등을 선택할 수 없습니다. 설치중에 사용자 및 그룹 생성, 인스턴스 생성을 수행하지 않으므로 사후 조치가 필요합니다. 설치중에 DB2 인스턴스 생성을 원하는 사용자는 DB2 설치 마법사를 사용하도록 합니다. 설치후 수동으로 인스턴스 생성 및 구성이 필요합니다.



DB2 Express – C 는 무료로 다운로드하여 사용할 수 있습니다. <http://www.ibm.com/db2/express> 에서 DB2 Express 제품군에 대한 정보 및 설치 이미지를 다운로드 받을 수 있습니다.

Point 이 곳에서는 DB2 서버 제품에 대한 설치 권한에 대해 알아보도록 합니다.

설치 권한


- DB2 서버를 설치하기 위해서는 Root 권한이 필요합니다. 그러나 DB2 V9.5부터는 설치하는 사용자가 root가 아니어도 가능합니다. 다만 아래와 같이 몇 가지 제약사항이 있으므로 주의하여 설치하도록 합니다. (Unix 와 Linux에서만 비 root 사용자 설치가 가능합니다.)

기준	Root 사용자 설치	비 Root 사용자 설치
DB2 모든 버전을 설치할 수 있는가?	✓ 설치 가능합니다.	✓ DB2 V9.5 만 설치가 가능합니다.
사용자가 설치 디렉토리를 선택할 수 있는가 ?	✓ 가능합니다.	✓ 선택이 불가능합니다. DB2 제품은 사용자의 홈 디렉토리 아래에 설치됩니다. (\$HOME/sqllib)
DB2 인스턴스를 여러 개 생성할 수 있는가 ?	✓ 다중으로 생성 가능합니다.	✓ 하나만 생성 가능합니다.

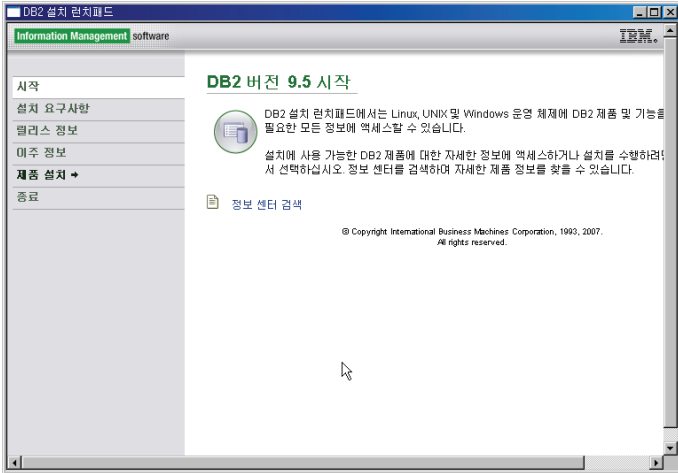
비 Root 사용자 설치시 주의 사항

- DB2 버전 9.5 이전에는 root 사용자만이 제품 설치, FixPack 적용 및 롤백, 인스턴스 구성이 가능하였으나 이제 비 root 사용자일 경우에도 이러한 태스크를 Linux 및 Unix 플랫폼에서 수행할 수 있습니다.
- 비 Root 사용자로 DB2를 설치할 경우 비 Root 인스턴스를 자동으로 생성하고 구성합니다. 또한 root 권한 없이도 DB2 제품 사용이 가능합니다.
- 비 Root 설치는 다음과 같은 사용자가 이용할 때 편리합니다.
 - 시스템 관리자를 방해하지 않고 DB2 제품을 설치하려는 수천개의 워크스테이션 및 사용자가 있는 기업.
 - DB2 제품을 사용하는 응용프로그램 개발자 및 root 권한이 없고 DB2 제품을 임베드하는 소프트웨어를 개발하는 솔루션 업체.
- 비 root 사용자 설치시 제약 사항
 - 제품 제약 사항: 다음과 같은 일부 DB2 제품은 비 root 설치에서는 지원되지 않습니다.
 - 데이터 서버 개발자 도구
 - DB2 Embedded 응용프로그램 서버 (DB2 EAS)
 - DB2 Query Patroller
 - DB2 Net Search Extender
 - 로컬로 설치된 DB2 정보 센터
 - 기능 및 도구 제약 사항: 다음 기능 및 도구는 비 root 설치시 사용할 수 없습니다.
 - DB2 Administration Server (DAS) 및 관련 명령어 (예) dasCRT, daslist, dasmigr 및 dasupdt)
 - 구성 지원 프로그램 및 제어 센터
 - db2governor가 우선순위를 높이는 기능은 지원되지 않습니다.
 - Work Load Manager (WLM) 기능을 이용할 경우 비 root DB2 인스턴스의 서비스 클래스에서 에이전트의 우선순위를 설정하려는 시도는 허용되나 에이전트의 우선순위는 존중되지 않고 SQLCODE 오류도 리턴되지 않습니다.
 - 시스템 재시작시 비 root DB2 인스턴의 자동 시작은 지원되지 않습니다.
 - Health Monitor에서 "경보 발생시 스크립트 또는 태스크 조치 실행" 및 "경보 통지 보내기" 기능은 지원되지 않습니다.
 - 단일 파티션 데이터베이스만을 지원합니다.
 - db2ls 명령어의 결과값은 root 사용자가 실행한 결과값과 다릅니다. (자세한 사항은 db2ls 명령어 참조)
 - 인스턴스 한계 :
 - 설치중에 단일 인스턴스만을 생성할 수 있습니다.
- 같은 시스템에 root 설치 및 비 root 설치가 가능하나 비 root 인스턴스의 갱신 및 삭제는 비 root 인스턴스의 소유자에게만 가능합니다.
 - 인스턴스 관련 명령어를 사용할 수 없습니다. (예) db2icrt, db2iupdt, db2idrop, db2imigr)
 - ulimit 값 조정 : 비 root 인스턴스의 경우 설치 중에만 변경이 가능합니다.
- 비 root 사용자 설치의 제약사항을 극복해 주는 명령어 :
 - db2rfe : 비 root 설치시 아래와 같은 기능은 처음에는 사용할 수 없으나 db2rfe 명령어를 통해 사용이 가능합니다.
 - 운영 체제 기반 인증 및 고가용성 (HA) 기능
 - /etc/services 파일에서 서비스 이름을 예약하는 기능

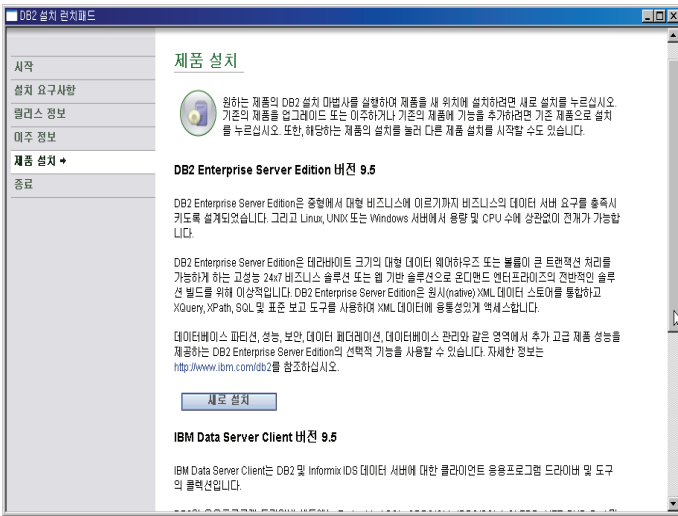
04 설치 마법사를 이용한 DB2 설치

Point  설치 마법사를 이용하여 윈도우 환경에서 DB2 Enterprise Server 제품군을 설치합니다.

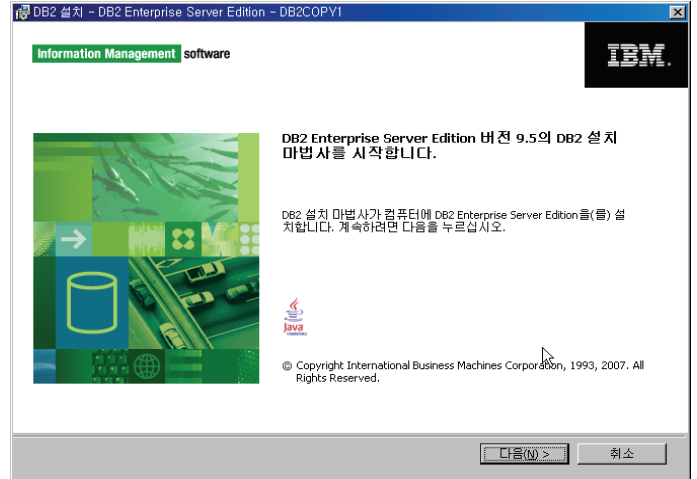
1. DB2 Enterprise Server Edition CD를 열어 db2setup을 실행합니다. 아래와 같은 화면이 나타나면 왼쪽 패널에 제품설치를 선택합니다.



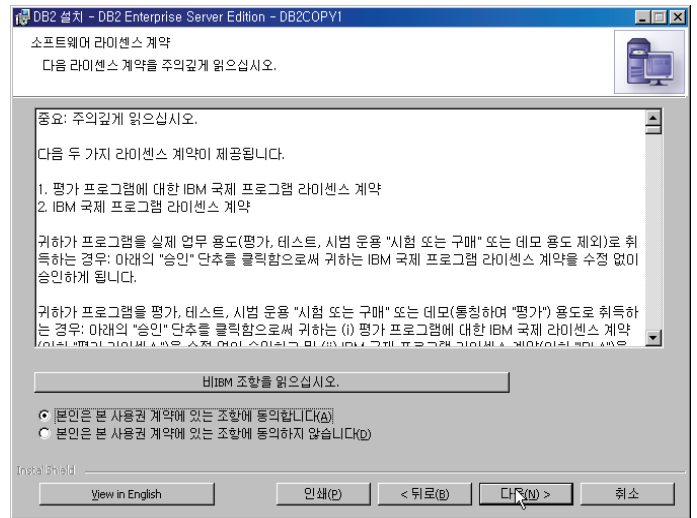
2. DB2 Enterprise Server Edition 버전 9.5의 "새로 설치" 버튼을 클릭합니다. DB2 Client 제품 설치시에는 IBM Data Server Client 버전 9.5 영역의 "새로 설치"를 선택합니다.



3. 다음과 같은 화면이 나타나면 "다음" 버튼을 눌러 진행합니다.



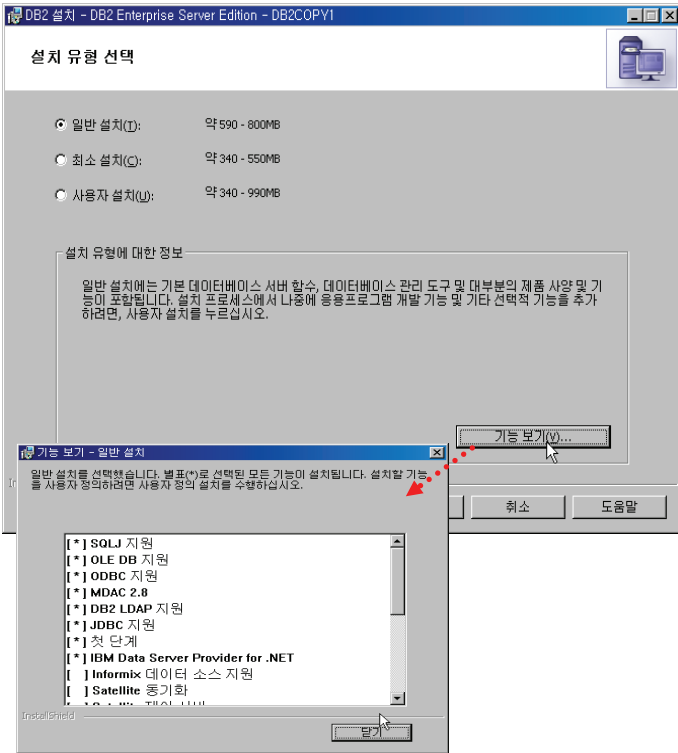
4. 사용자 계약서에서 동의함을 선택하고 "다음" 버튼을 클릭합니다.



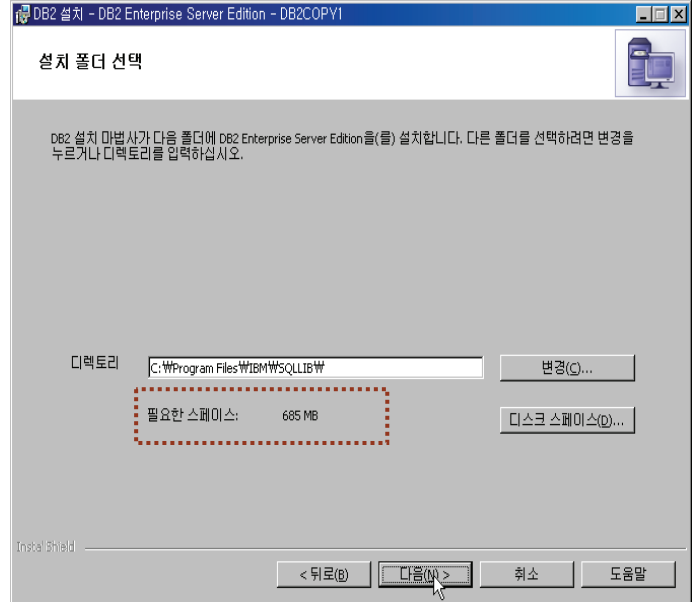
05 설치 마법사를 이용한 DB2 설치

Point DB2의 플랫폼은 윈도우 환경이며 설치 마법사를 이용하여 설치합니다.

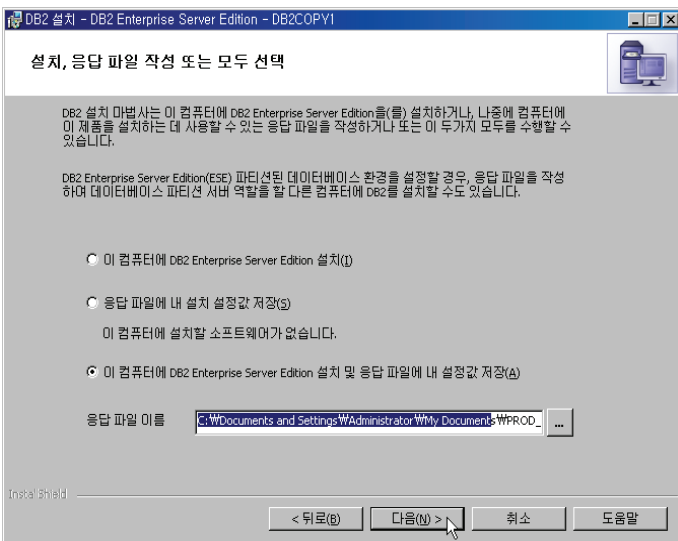
- 설치 유형 선택에서 "일반 설치"를 선택합니다. 일반 설치시 설치될 기능을 확인해 보고 싶을 경우 "기능 보기" 버튼을 클릭하면 아래 그림과 같이 설치할 기능을 확인할 수 있습니다. 설치할 기능을 추가하거나 제거할 경우 사용자 설치를 선택합니다. 다음 버튼을 클릭합니다.



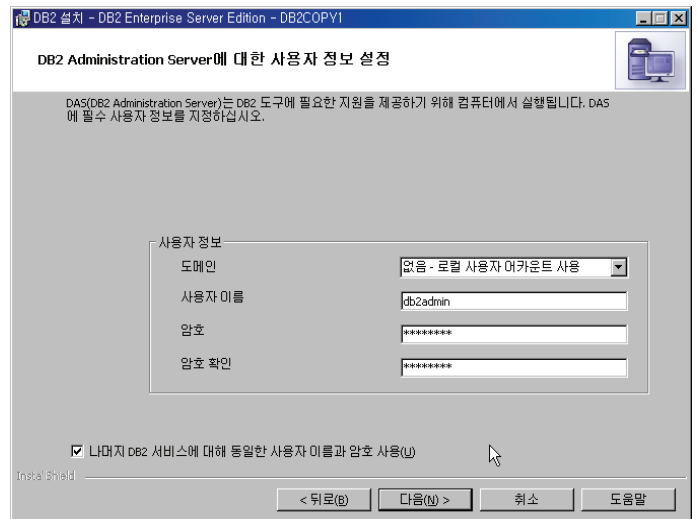
- DB2를 설치할 폴더를 선택합니다. Default 값은 Window의 경우 C:\WProgram Files\WIBM\WSQLIBW 입니다.



- 아래 그림과 같이 DB2 Server 설치 및 설치 설정값들을 응답파일에 저장하도록 선택한후 "다음" 버튼을 클릭합니다.



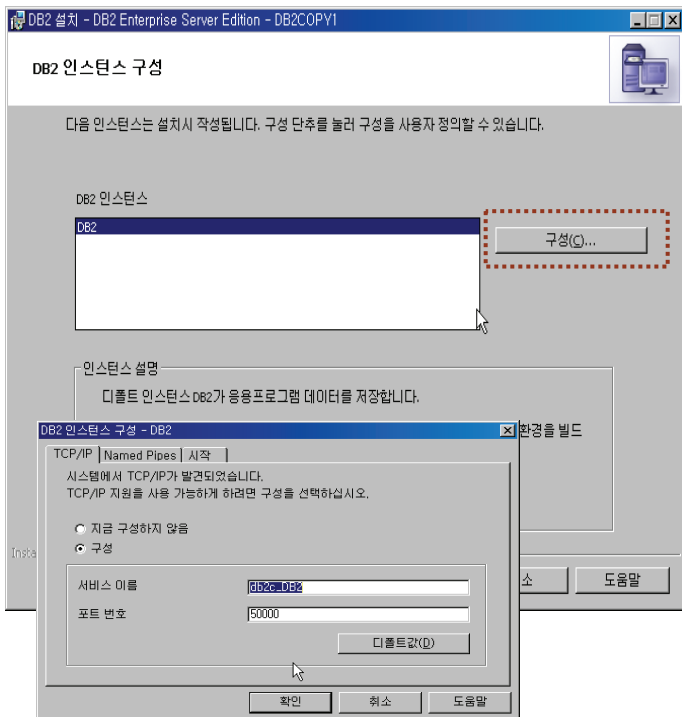
- DB2 Administration Server(DAS)를 생성합니다. DAS 서버는 제어센타 및 구성 지원 프로그램과 같은 DB2 도구에 대한 지원 서비스를 제공합니다. DAS 서버 및 인스턴스를 관리할 사용자 및 그 사용자 암호를 지정합니다. OS에 존재하지 않는 새로운 사용자 정보를 입력할 경우에는 OS 사용자가 자동으로 OS 상에 생성됩니다.



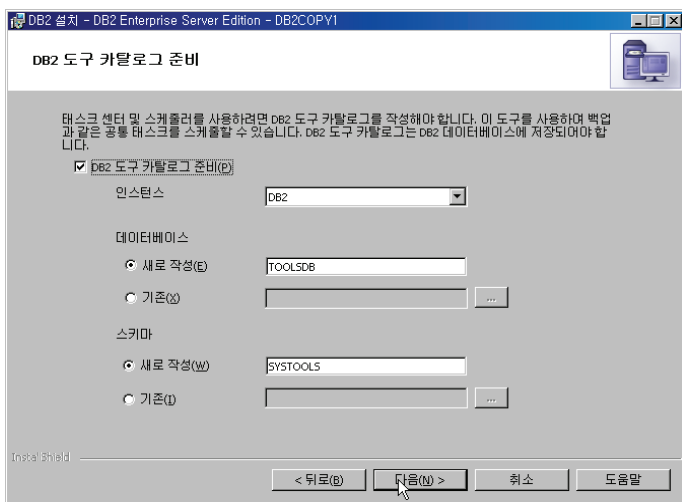
06 설치 마법사를 이용한 DB2 설치

Point DB2의 플랫폼은 윈도우 환경이며 설치 마법사를 이용하여 설치합니다.

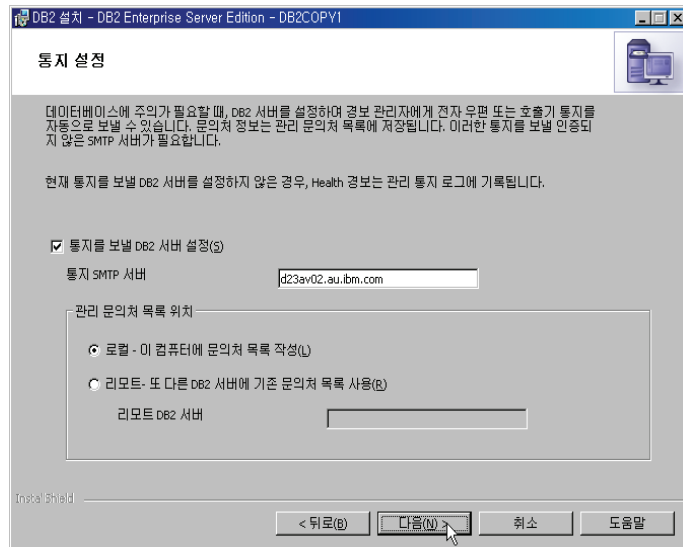
9. DB2 인스턴스를 구성합니다. "구성" 버튼을 눌러 인스턴스가 Listen 할 포트를 지정합니다. Default 로는 버튼에 따라 차이가 있으나 50000 혹은 60000으로 설정됩니다.



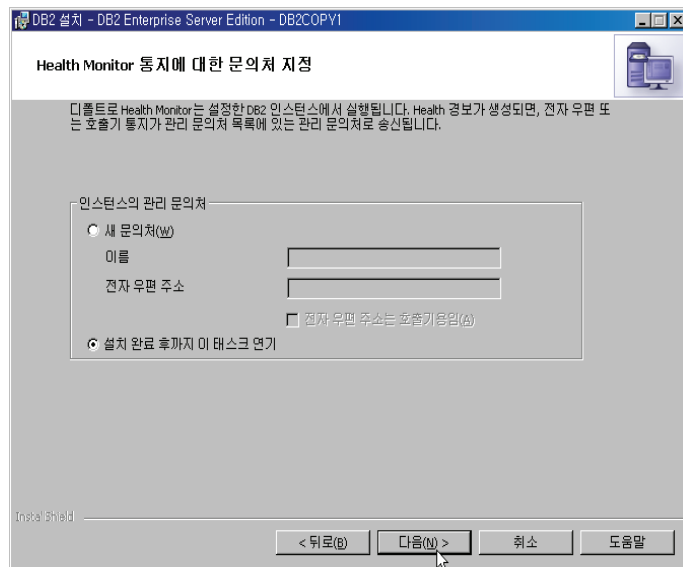
10. 태스크 센터 및 스케줄러와 같은 특정 DB2 도구를 사용하기 위해서는 DB2 도구 카탈로그가 필요합니다. 하지만 이와같은 DB2 도구를 사용하지 않을 경우에는 생성하지 않아도 됩니다.



11. 데이터베이스에 문제 발생 또는 주기가 필요할 경우 자동으로 관리자에게 통지를 보낼 수 있도록 SMTP 서버를 지정합니다.



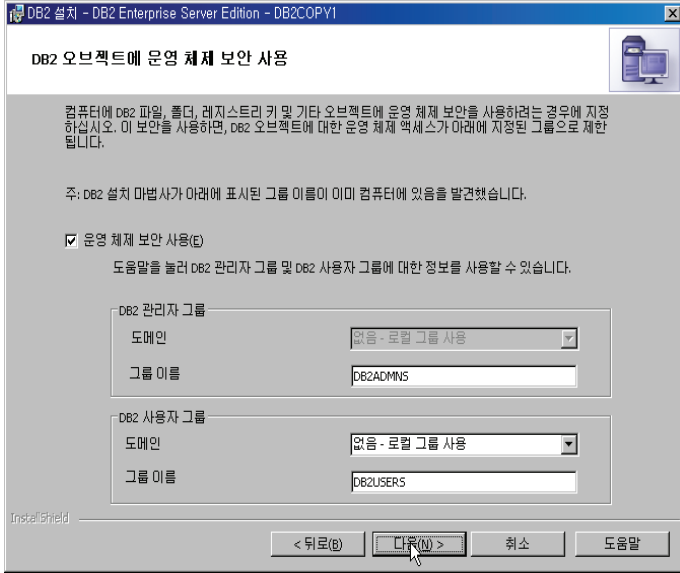
12. Health Monitor에 경보가 생성되면 앞에서 설정한 SMTP 서버를 통해 이곳에서 등록한 Email 로 통지를 보냅니다. 하지만 문의처가 정해지지 않았을 경우 태스크를 연기한 후 향후에 등록할 수 있습니다.



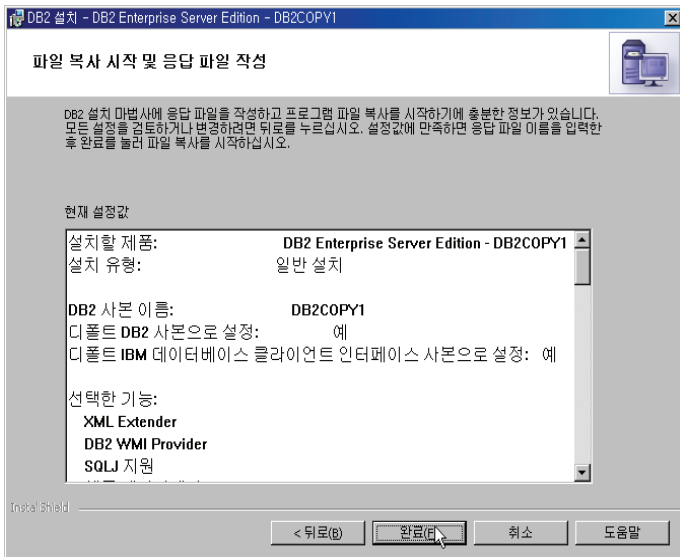
07 설치 마법사를 이용한 DB2 설치

Point DB2의 플랫폼은 윈도우 환경이며 설치 마법사를 이용하여 설치합니다.

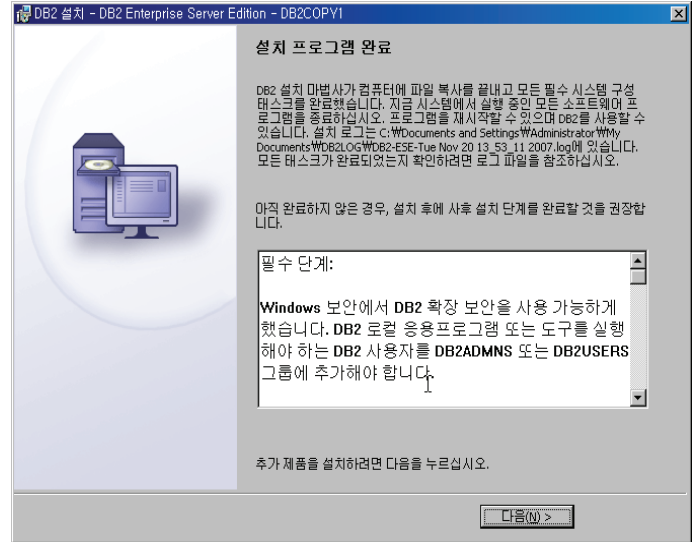
13. 컴퓨터에서 DB2 파일, 폴더, 레지스트리 키 및 기타 오브젝트에 운영체제 보안을 사용할 경우 아래와 같이 지정합니다. 운영 체제 보안을 사용하면 DB2 오브젝트에 대한 운영체제 액세스가 지정된 그룹으로 제한됩니다.



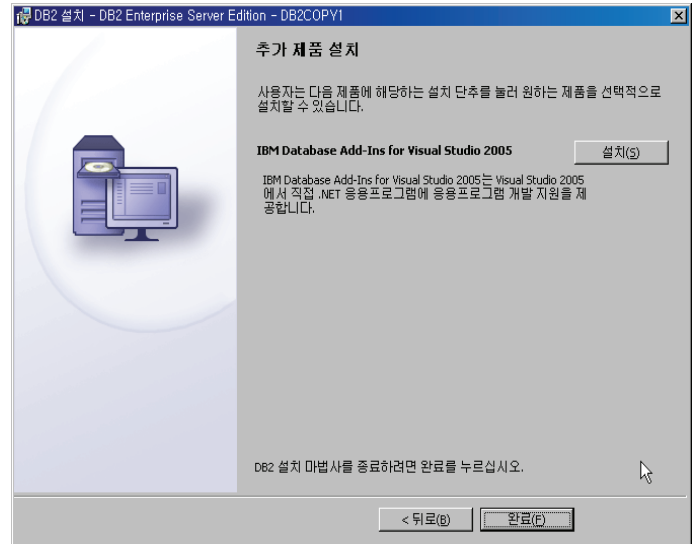
14. 아래 화면은 이제까지 지정한 설정값입니다. 만약 이러한 설정값을 응답 파일로 저장할 경우 다른 컴퓨터에서 동일한 설정값으로 DB2를 설치할 수 있습니다. 설정값들을 확인한후 "완료" 버튼을 클릭합니다.



15. DB2 설치가 완료되면 아래와 같이 설치 프로그램 완료 화면이 나타납니다. "다음" 버튼을 눌러 추가 제품을 설치합니다.



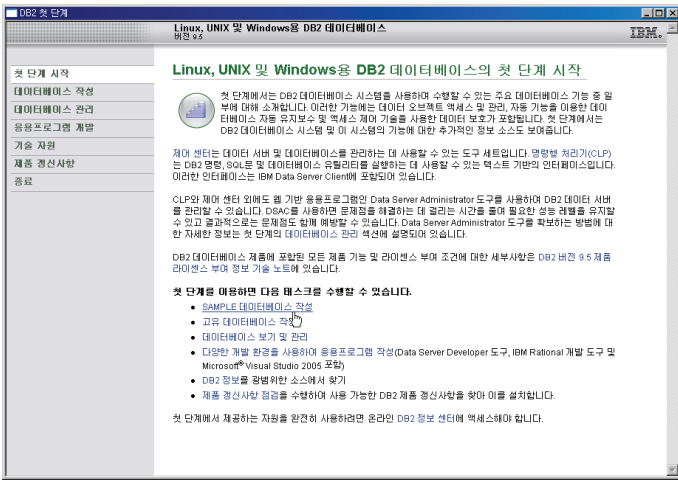
16. Visual Studio 2005 개발자들이 편리하게 DB2를 이용하여 개발할 수 있도록 Visual Studio 2005에 DB2 플러그인을 지원합니다. 추가 제품을 설치하지 않을 경우 "완료" 버튼을 클릭합니다.



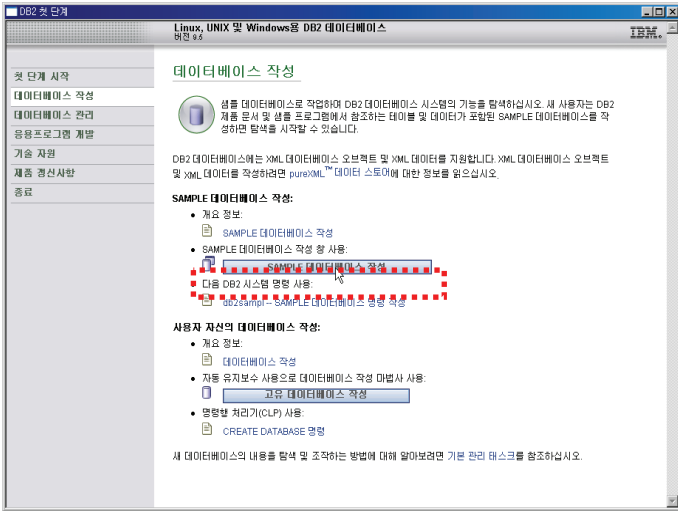
08 설치 마법사를 이용한 DB2 설치

Point DB2의 플랫폼은 윈도우 환경이며 설치 마법사를 이용하여 설치합니다.

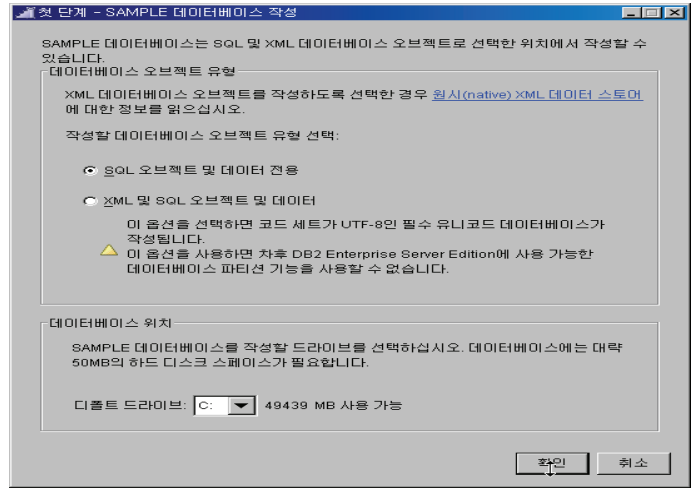
17. DB2 설치가 완료되면 다음과 같이 첫단계 시작 화면이 나타납니다. 오라클의 HR Schema와 같이 DB2도 Sample 데이터베이스를 작성할 수 있습니다. 아래 "SAMPLE 데이터베이스 작성"을 클릭하여 SAMPLE 데이터베이스를 작성합니다.



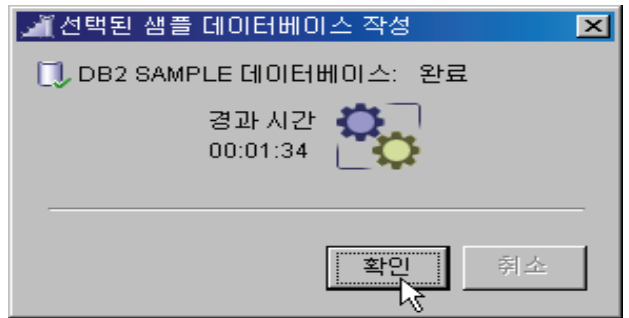
18. "SAMPLE 데이터베이스 작성"을 클릭합니다.



19. 데이터베이스내에 생성할 오브젝트 타입을 선택합니다. 이 곳에서는 향후 실습을 위해서 "SQL 오브젝트 및 데이터 전용"을 선택합니다. 만약 XML 오브젝트도 생성하기 원하면 "XML 및 SQL 오브젝트 및 데이터"를 선택합니다. 그러나 XML 오브젝트를 생성할 데이터베이스의 코드 세트는 UTF-8이 됩니다.



20. 다음 화면과 같이 Sample 데이터베이스 생성이 완료되었다라는 메시지를 확인합니다.



09 명령어를 이용한 DB2 설치

Point DB2의 플랫폼은 AIX 환경이며 db2_install 명령어를 이용하여 설치합니다.

1. DB2 이미지가 있는 디렉토리로 이동하여 다음과 같이 db2prereqcheck 명령어를 수행합니다. 이 명령어는 DB2를 설치하기 전에 설치 요구사항을 만족하는지에 대해 체크합니다. 아래와 같이 최소 설치 요구사항이 충족되지 않을 경우에는 에러메시지가 나타나고 충족될 경우 어떤 메시지도 나타나지 않습니다. 설치 요구사항이 충족되지 않을 경우 메시지를 확인하여 시스템 관리자에게 통보하여 OS 플랫폼 및 XL C++ Runtime을 Upgrade 하도록 합니다.

```

c\> 텔넷 172.22.83.3
spcp6303@root:/nfs_sw/db2v95/DB2_ESE_U95>ls
DB2_ESE_U95_AIX.tar  doc/
db2/                installFixPack*
db2_deinstall*     nlpack/
db2_install*       readmefirst/
db2prereqcheck*   readmefirst.htm
db2setup*          readmefirst.txt
spcp6303@root:/nfs_sw/db2v95/DB2_ESE_U95>./db2prereqcheck
spcp6303@root:/nfs_sw/db2v95/DB2_ESE_U95>_
    
```

2. 위의 설치 요구사항이 모두 만족될 경우 다음과 같이 db2_install 명령어를 -h 옵션과 함께 실행하면 설치 옵션 정보가 나타납니다.

```

c\> 텔넷 172.22.83.3
spcp6303@root:/nfs_sw/db2v95/DB2_ESE_U95>./db2_install -h
DBI1045I Usage:

db2_install [-b <installpath>] [-p <db2producttobeinstalled>]
             [-c <imagelocation>] [-l <logfile>] [-f NOTSAMP]
             [-t <trcFile>] [-n] [-L <language>] [-h|-?]

Explanation:

-b          This option is mandatory for root install when -n is sp
cified
           but optional for non-root install. Specifies the path w
ere the
           DB2 product will be installed. Mandatory when -n is spe
cified.
-l          The length of the path is limited to 128 characters and
is a
           full path name.
-p          Specifies the DB2 product to be installed. Mandatory wh
-n is
    
```

3. db2_install 명령어에 옵션을 지정하여 설치할 수 있으며 옵션을 지정하지 않을 경우 다음과 같이 대화식으로 설치가 가능합니다. db2_install 명령어를 실행한 후 DB2가 설치될 디렉토리를 지정합니다. Default 설치경로인 /opt/IBM/db2/V9.1 (AIX의 경우)에 설치할 경우에는 no 를 입력하고 다른 설치 경로를 지정할 경우 yes 를 입력하고 설치 디렉토리를 입력한 후 설치할 제품을 선택합니다. 여기서는 DB2 Enterprise Server Edition을 설치하므로 ESE를 입력합니다.

```

c\> 텔넷 172.22.83.3
spcp6303@root:/nfs_sw/db2v95/DB2_ESE_U95>./db2_install

Default directory for installation of products - /opt/IBM/db2/V9.1.

5
*****
Do you want to choose a different directory to install [yes/no] ?
no

Specify one of the following keywords to install DB2 products.
  ESE
  CLIENT
  RTCL

Enter "help" to redisplay product names.
Enter "quit" to exit.
*****
ESE_
    
```

4. 설치가 완료되면 아래그림과 같은 메시지가 나타납니다. /tmp/db2_install.log.<ProcessID>의 설치 로그 파일을 열어 설치가 되었는지 확인합니다.

```

c\> 텔넷 172.22.83.3
Task #43 end

Task #44 start
Description: Initializing instance list
Estimated time 5 second(s)
Task #44 end

Task #45 start
Description: Updating global profile registry
Estimated time 3 second(s)
Task #45 end

The execution completed successfully.

For more information see the DB2 installation log at
"/tmp/db2_install.log.581654".
spcp6303@root:/nfs_sw/db2v95/DB2_ESE_U95>
spcp6303@root:/nfs_sw/db2v95/DB2_ESE_U95>_
    
```

/tmp/db2_install.log.<ProcessID> 파일의 예)

```

c\> 텔넷 172.22.83.3
Installing DB2 file sets:.....Success
Setting DB2 library path:.....Success
Installing or updating SA MP Base Component:.....Success
Installing national language support for SA MP Base Component:.....
.Success
Installing or updating DB2 HA scripts for SA MP Base Component:.....
.Success
Updating global registry:.....Success
The installation of IBM Tivoli Monitoring for Databases has complet
d. The log
file can be found at /opt/IBM/db2/V9.5/itna/logs/stm_install.log.

Starting DB2 Fault Monitor:.....Success
Updating the db2ls link:.....Success
Registering DB2 licenses:.....Success
Setting default global profile registry variables:.....Success

Task #24 start
    
```

Point db2_install 명령어로 DB2 서버를 설치할 경우 설치 속도는 빠르지만 인스턴스 생성 등 사후 조치가 필요합니다. 이번장에서는 인스턴스 생성 방법을 간단히 보여주며 향후에 인스턴스 생성 및 구성에 관해 자세히 다루도록 합니다.

1. root 사용자로 로그인하여 인스턴스 Owner가 될 사용자와 그룹을 생성합니다. 또한 향후 다시 설명드리겠지만 스토어드 프로시저 또는 사용자 함수를 실행할때 DB2 인스턴스와 별도의 메모리영역과 프로세스를 실행할 사용자와 그룹을 생성합니다. 이는 사용자가 개발한 함수 또는 프로시저에 버그가 있을 경우 DB2 엔진에 영향을 주지 않기 위해서입니다.

```

c\> 텔넷 172.22.83.3
spcp6303@root:>mkgroup db2igrp
spcp6303@root:>mkuser pgrp=db2igrp inst95
spcp6303@root:>mkgroup db2fgrp
spcp6303@root:>mkuser pgrp=db2fgrp fenc95
spcp6303@root:>_
  
```

2. 두 사용자 생성이 완료되면 아래와 같이 DB2가 설치된 디렉토리로 이동합니다. 하위 디렉토리인 instance로 이동하여 아래 그림과 같이 db2icrt 명령어를 이용하여 인스턴스를 생성합니다.

```

c\> 텔넷 172.22.83.3
spcp6303@root:>cd /opt/IBM/db2/V9.5/instance
spcp6303@root:/opt/IBM/db2/V9.5/instance>./db2icrt -u fenc95 inst95
DBI1070I Program db2icrt completed successfully.

spcp6303@root:/opt/IBM/db2/V9.5/instance>_
  
```

3. 인스턴스 Owner로 로그인 하거나 또는 아래 그림과 같이 su 명령어를 이용하여 인스턴스 owner로 변경한 후 "db2sampl" 명령어를 이용하여 SAMPLE 데이터베이스를 생성합니다. 이는 오라클 sample인 HR Schema와 유사합니다.

```


c\> 텔넷 172.22.83.3
spcp6303@root:>su - inst95
$ db2sampl

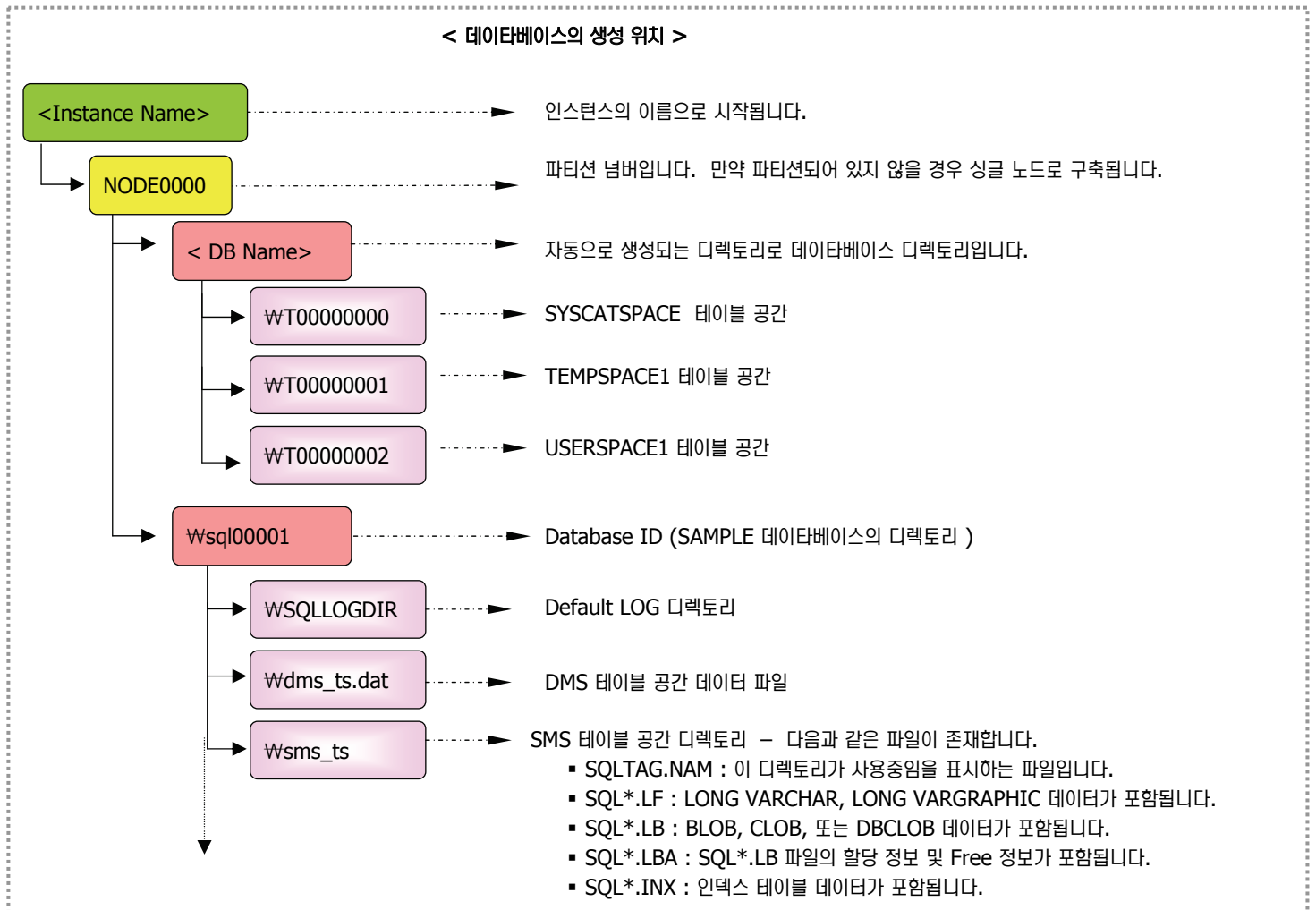
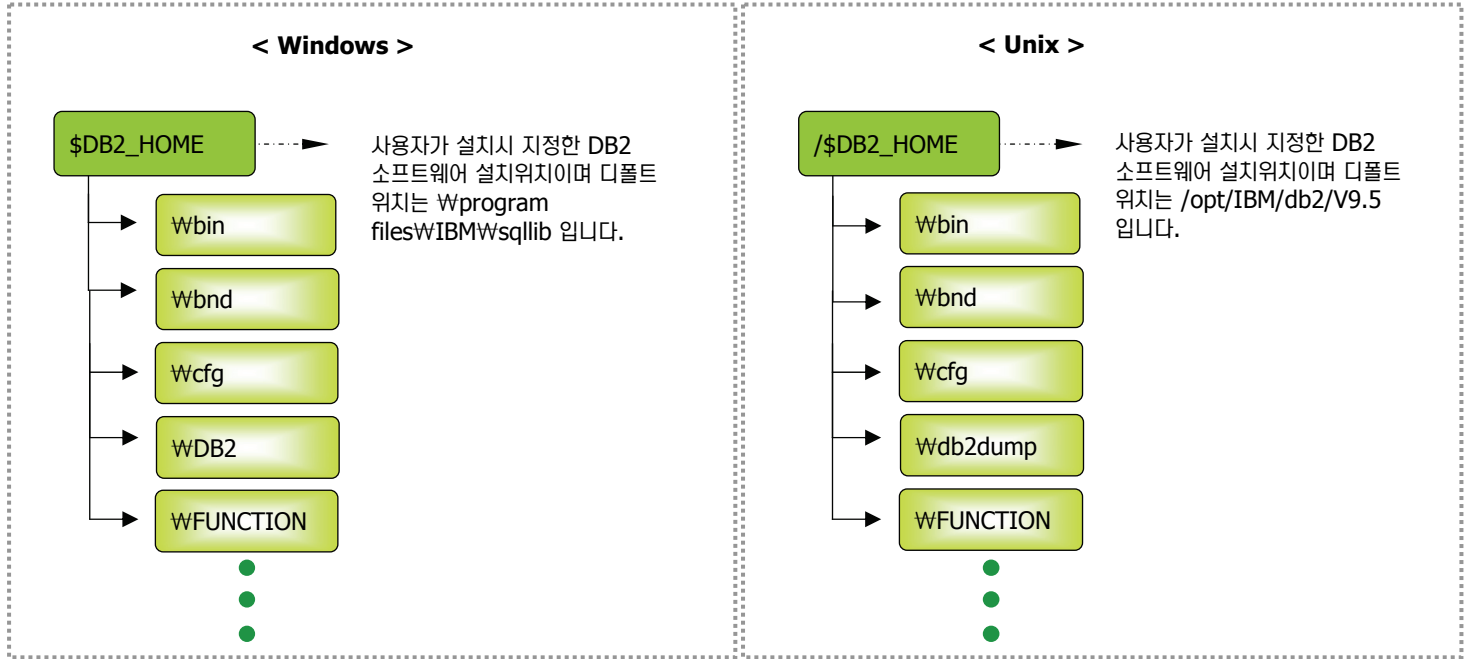
Starting the DB2 instance...
Creating database "SAMPLE"...
Connecting to database "SAMPLE"...
Creating tables and data in schema "INST95"...
Stopping the DB2 instance...

'db2sampl' processing complete.

$ _
  
```

11 DB2 제품 설치 디렉토리

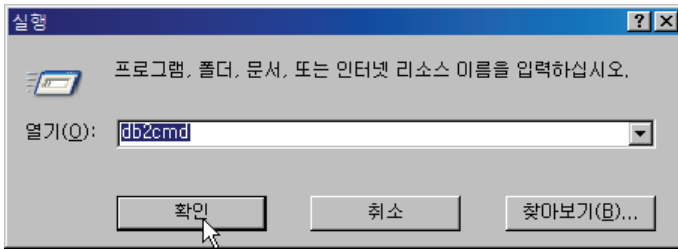
Point  다음은 root 사용자로 DB2를 설치할 경우 DB2 제품이 설치되는 물리적인 위치입니다. 아래 그림은 Windows 와 AIX에 제품이 설치될 때의 설치 디렉토리 구조입니다. OS 별로 차이가 날 수 있으므로 DB2 V9.5의 도움말을 참고하시기 바랍니다.



12 GUI Tool을 통한 라이선스 등록

Point DB2의 플랫폼은 윈도우 환경이며 설치 마법사를 이용하여 설치합니다.

1. DB2 Administration Server(DAS)는 제어 센터 및 구성 지원 프로그램과 같은 DB2 도구에 대한 지원 서비스를 제공합니다. 설치시 디폴트로 DAS 서버는 생성되나 만약 생성되지 않았을 경우 제어센터 및 구성 지원 프로그램과 같은 DB2 도구를 사용할 때 제약점이 발생합니다. 다음과 같이 DAS 서버가 생성되었는지 확인한 후 생성이 되었다면 "db2admin start" 명령어를 이용하여 DAS 서버를 시작하도록 합니다. 시작 => 실행 을 선택한 후 열기란에 "db2cmd"을 입력합니다.



2. DB2 CLP창이 나타나면 "db2admin" 을 입력하여 DAS 서버가 존재하는지 확인합니다.

```
C:\W> db2admin
```

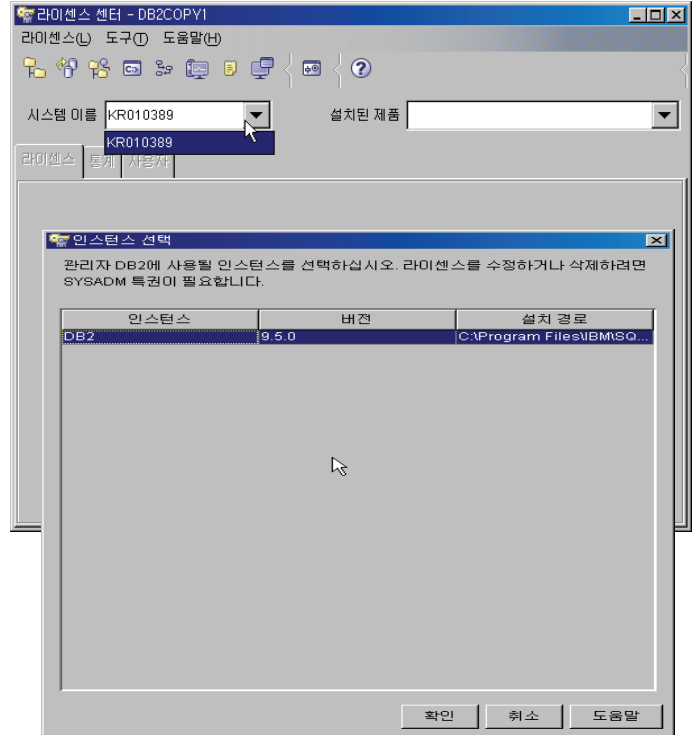
```
DB2DAS00
```

3. DAS 서버가 존재할 경우 다음과 같이 DAS 서버를 시작합니다. 이미 시작되었을 경우 다음과 같이 "이미 사용 중입니다"라는 메시지가 나타납니다.

```
C:\W> db2admin start
```

```
SQL4409W DB2 Administration Server가 이미 사용 중입니다.
```

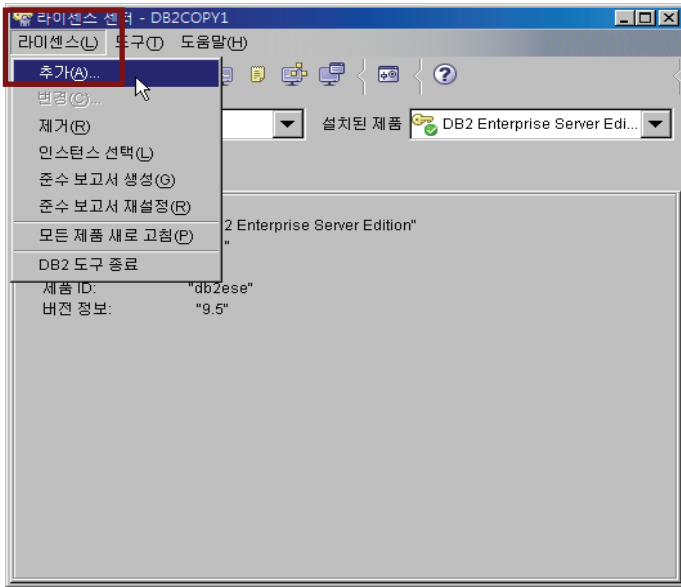
4. 시작 => 모든 프로그램 => IBM DB2 => DB2COPY1 (디폴트) => 일반 관리 도구 => 라이선스 센터를 선택하면 아래와 같은 화면이 나타납니다. 왼쪽 위의 시스템 이름 항목에서 DB2 라이선스를 등록할 시스템 이름을 선택한후 별도의 창이 뜨면 인스턴스를 선택하고 확인 버튼을 클릭합니다.



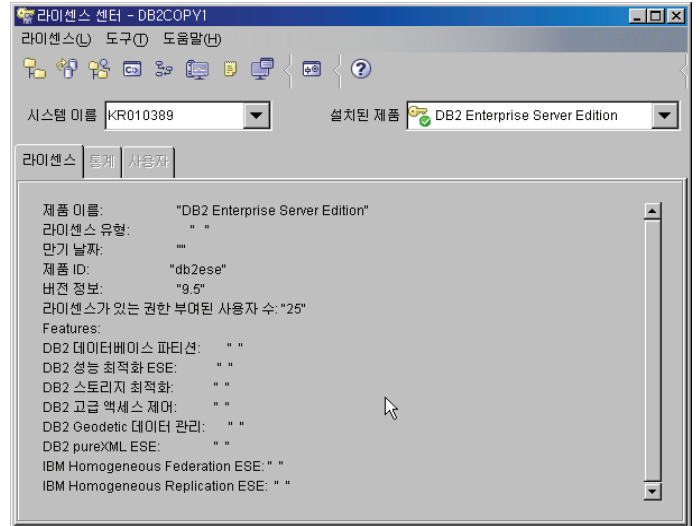
13 GUI Tool을 통한 라이선스 등록

Point DB2의 플랫폼은 윈도우 환경이며 설치 마법사를 이용하여 설치합니다.

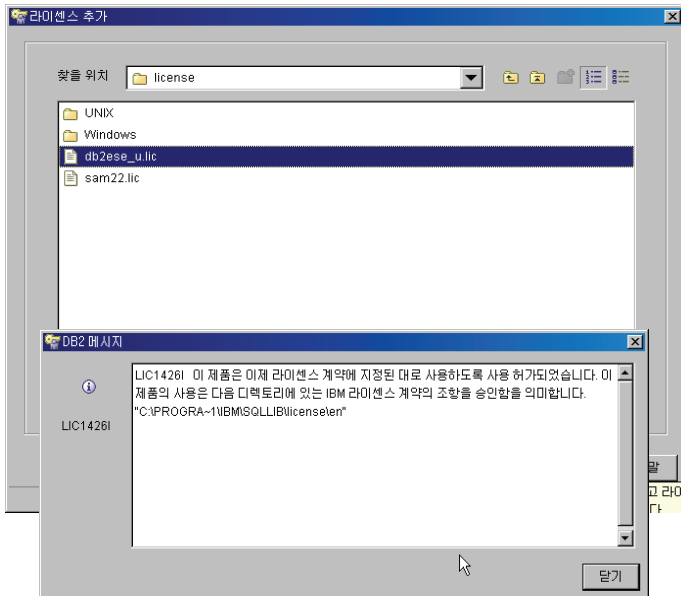
5. 아래와 같이 라이선스 정보가 없을 경우 라이선스가 등록되지 않은 경우입니다. 메뉴에서 라이선스를 선택한후 추가를 클릭합니다.



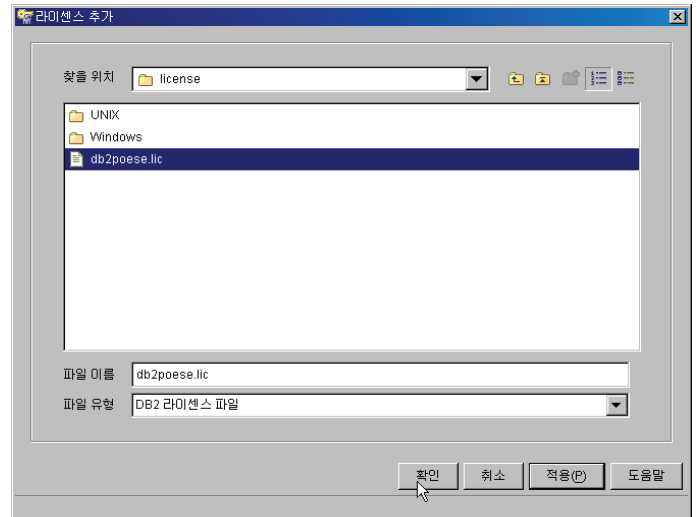
7. 라이선스 등록이 모두 완료되면 아래와 같이 DB2 License 정보를 확인할 수 있습니다. 아래 라이선스의 경우에는 권한 부여된 사용자 수로 구매한 제품으로 25명의 사용자 수를 구매한 제품입니다. 아래 화면과 같이 Feature의 경우 별도의 라이선스가 필요하므로 별도의 라이선스를 구매한 후 등록해야 합니다.



6. 아래와 같이 라이선스 파일(db2eseX.lic 파일)을 찾아 선택한 후 확인 버튼을 선택하면 라이선스 등록이 완료되었다라는 메시지 창이 나타납니다.



8. 위의 Feature 중 구매한 Feature가 있을 경우 앞에서 수행한 동일한 방법으로 라이선스를 등록하면 됩니다.



14 명령어를 통한 라이선스 등록

Point DB2의 플랫폼은 AIX 환경이며 db2_install 명령어를 이용하여 설치합니다.

1. root 사용자로 로그인하여 DB2 설치 디렉토리로 이동합니다. DB2 V9.5의 Default 설치 위치는 /opt/IBM/db2/V9.5 입니다. 설치 디렉토리의 하위 디렉토리인 adm 디렉토리로 이동합니다. 이곳에서 license 관리 명령어인 db2licm 을 -l 옵션과 함께 사용할 경우 라이선스 등록 정보를 확인할 수 있습니다.

```

cvt 텔넷 172.22.83.3
sncp6303@root:~/>cd /opt/IBM/db2/V9.5/adm
sncp6303@root:~/>./db2licm -l
Product name:          "DB2 Enterprise Server Edition"
License type:          "License not registered"
Expiry date:           "License not registered"
Product identifier:    "db2ese"
Version information:   "9.5"

sncp6303@root:~/>

```

2. 구매한 라이선스 파일을 특정 디렉토리에 저장한후 아래와 같이 db2licm 명령어를 -a 옵션과 함께 수행하여 등록합니다. 라이선스 파일의 이름은 db2ese-c.lic 파일입니다.

```

cvt 텔넷 172.22.83.3
sncp6303@root:~/>./db2licm -a /db2lic/db2ese_c.lic
LIC14021 License added successfully.

LIC14261 This product is now licensed for use as outlined in your License Agreement. USE OF THE PRODUCT CONSTITUTES ACCEPTANCE OF THE TERMS OF THE IBM LICENSE AGREEMENT, LOCATED IN THE FOLLOWING DIRECTORY: "/opt/IBM/db2/V9.5/license/en_US.iso88591"
sncp6303@root:~/>

```

3. 다시 한번 1번과 동일하게 db2licm 명령어를 -l 옵션과 함께 수행할 경우 아래와 같이 라이선스가 등록된 것을 확인할 수 있습니다. 별도 구매 Feature의 경우에도 라이선스 구매 후 db2licm 명령어를 이용하여 등록합니다.

```

cvt 텔넷 172.22.83.3
sncp6303@root:~/>./db2licm -l
Product name:          "DB2 Enterprise Server Edition"
License type:          "CPU Option"
Expiry date:           "Permanent"
Product identifier:    "db2ese"
Version information:   "9.5"
Features:
DB2 Database Partitioning: "Not licensed"
DB2 Performance Optimization ESE: "Not licensed"
DB2 Storage Optimization: "Not licensed"
DB2 Advanced Access Control: "Not licensed"
DB2 Geodetic Data Management: "Not licensed"
DB2 pureXML ESE: "Not licensed"
IBM Homogeneous Federation ESE: "Not licensed"
IBM Homogeneous Replication ESE: "Not licensed"

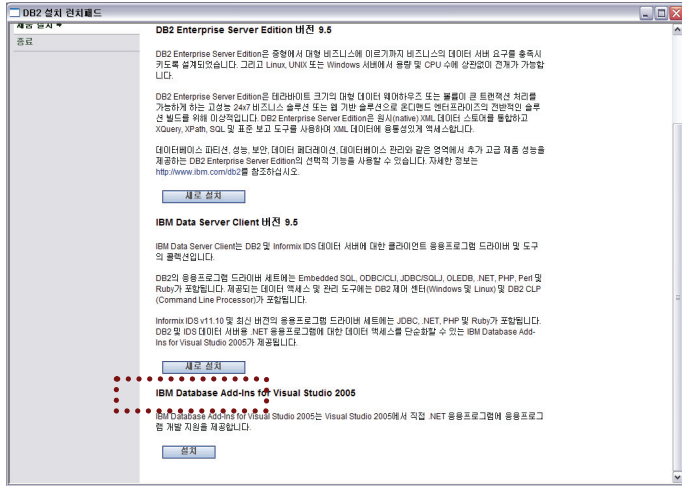
sncp6303@root:~/>

```

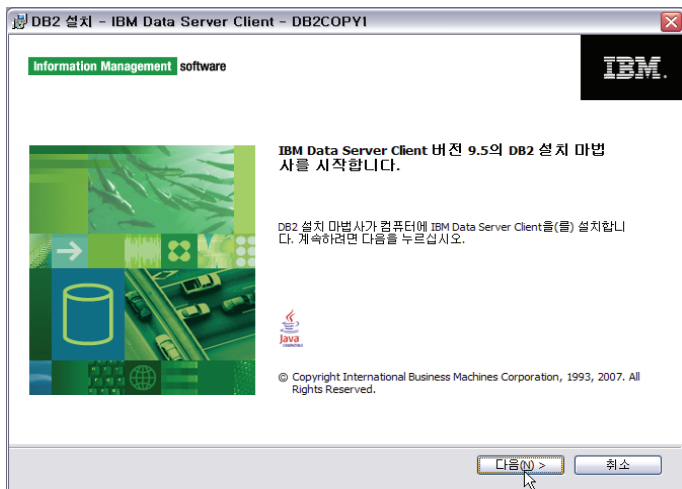
15 DB2 Client 제품 설치

Point DB2 클라이언트가 설치되는 플랫폼은 윈도우 환경이며 설치 마법사를 이용하여 설치합니다.

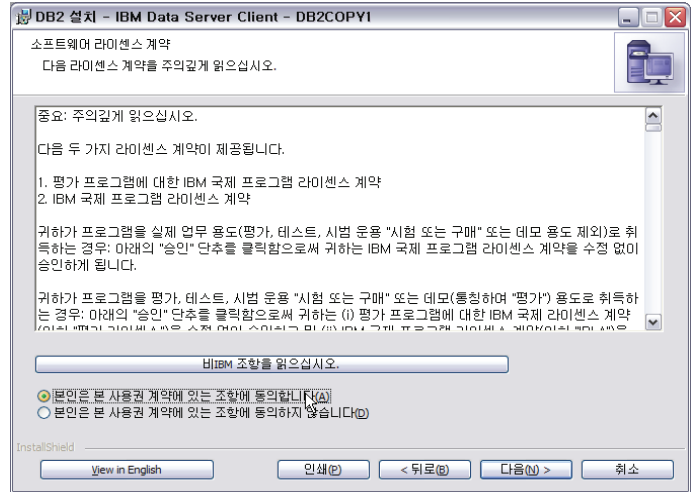
- Client 설치 이미지가 있는 디렉토리로 이동하거나 또는 DB2 설치 CD를 넣고 setup.exe 을 실행합니다. IBM Data Server Client 영역에서 제품 설치를 선택합니다.



- 아래와 같은 화면이 나타나면 "다음" 버튼을 눌러 설치를 시작합니다.



- 소프트웨어 라이선스 계약 화면이 나타나면 동의를 선택합니다.



- 설치 유형을 선택합니다. 최소 설치는 약 100~200M 공간을 차지합니다. Component를 개별적으로 선택하려면 "사용자 설치"를 선택한 후 "다음" 버튼을 클릭합니다.

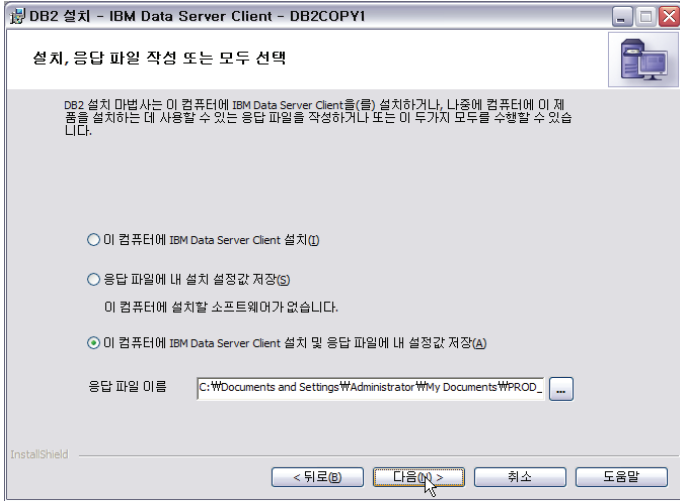


DB2 Client 제품은 <http://www-306.ibm.com/software/data/db2/9/download.html> 에서 다운로드 받으시기 바랍니다.

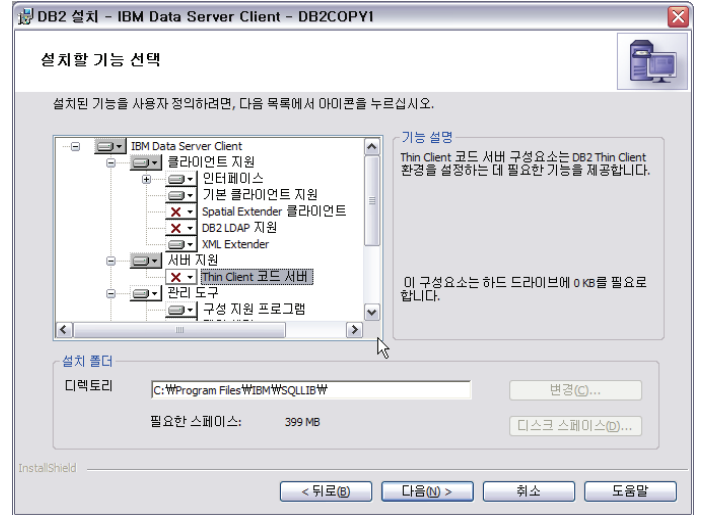
16 DB2 Client 제품 설치

Point DB2 클라이언트가 설치되는 플랫폼은 윈도우 환경이며 설치 마법사를 이용하여 설치합니다.

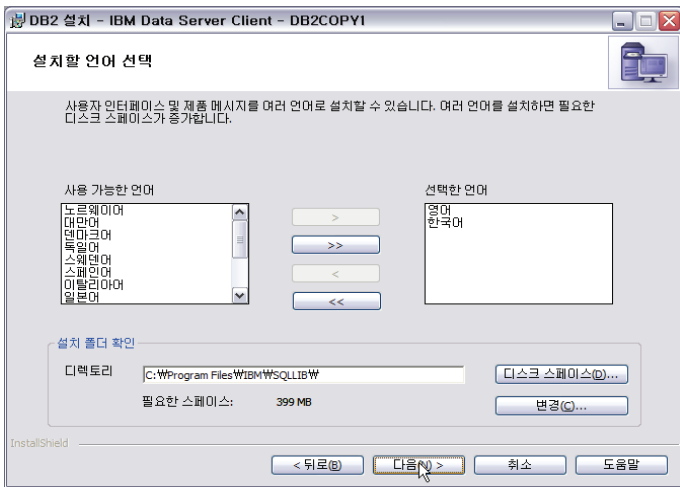
5. IBM Data Server Client를 설치할때 설치 설정값을 응답 파일에 저장하여 향후 타 시스템에 동일한 설정값으로 Client를 설치할 수 있습니다.



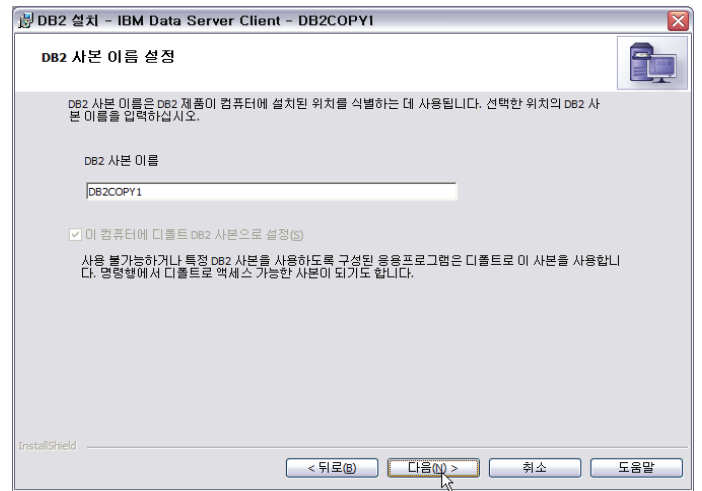
7. 설치유형을 사용자 설치로 지정하였으므로 아래와 같이 필요한 기능을 선택하여 설치합니다.



6. 설치할 언어를 선택합니다.



8. 그 후 DB2 사본 이름을 지정합니다. 이는 DB2 제품이 컴퓨터에 설치될 위치를 식별하며 DB2를 실행하는데 필요한 환경을 구성합니다.



IBM Data Server Client는 원격 및 로컬 DB2 서버에 연결하여 사용자가 명령 및 SQL 문을 실행하고 데이터베이스에 액세스하는 애플리케이션입니다. 다음은 IBM Data Server Client 종류이며 목적에 맞는 클라이언트 제품을 다운로드 받아 설치하시면 됩니다.

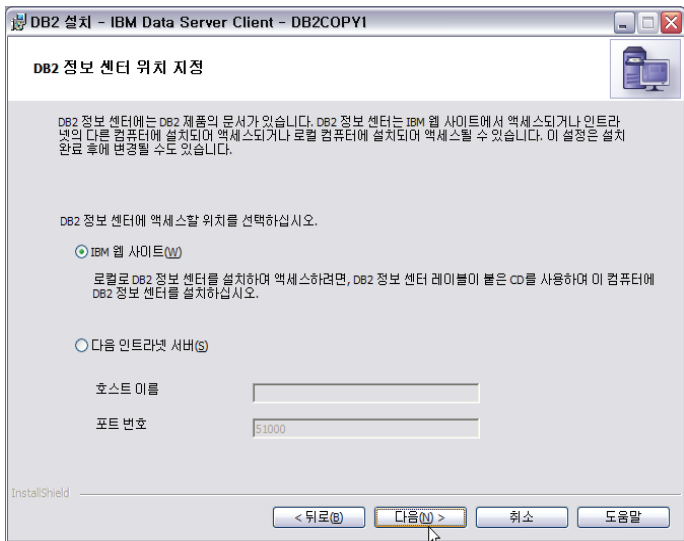
(이 제품은 V9.5 제품이며 이전 버전은 제품명이 다릅니다. 자세한 사항은 1장 제품군 비교를 참조하시기 바랍니다.)

- IBM Data Server Client
- IBM Data Server Runtime Client
- IBM Data Server Driver for ODBC, CLI 및 .NET

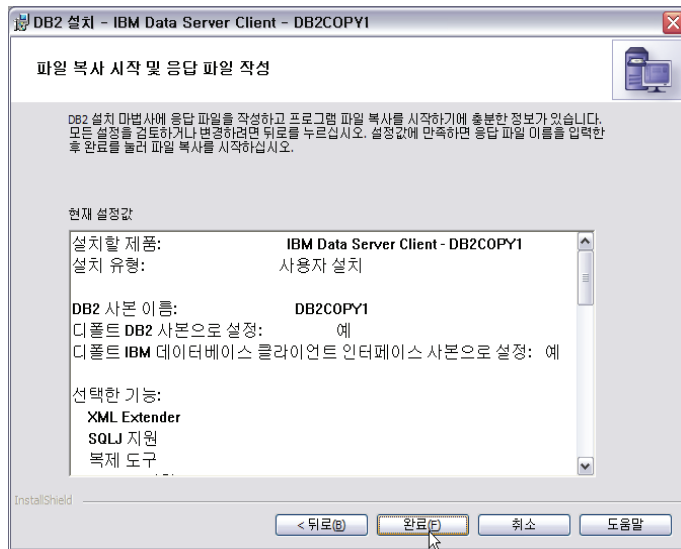
17 DB2 Client 제품 설치

Point DB2 클라이언트가 설치되는 플랫폼은 윈도우 환경이며 설치 마법사를 이용하여 설치합니다.

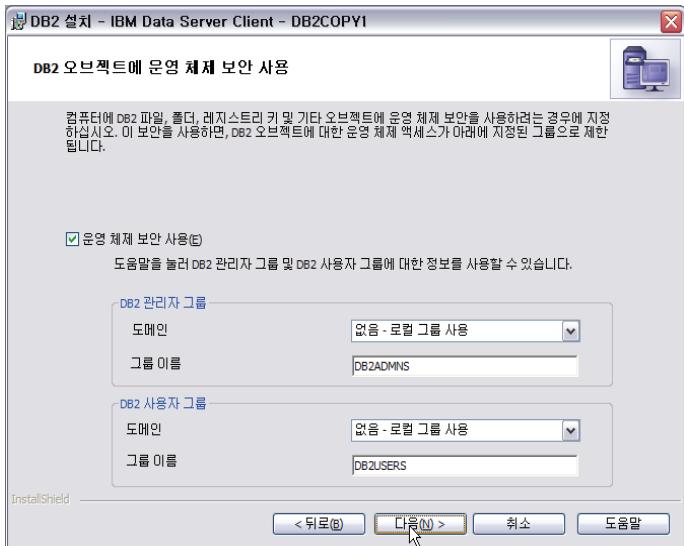
9. DB2 정보센터를 구성하면 필요시 DB2 정보를 찾아볼 수 있습니다. 디폴트는 IBM 웹사이트이며 인트라넷이 구성되어 있을 경우 그정보를 입력하면 됩니다.



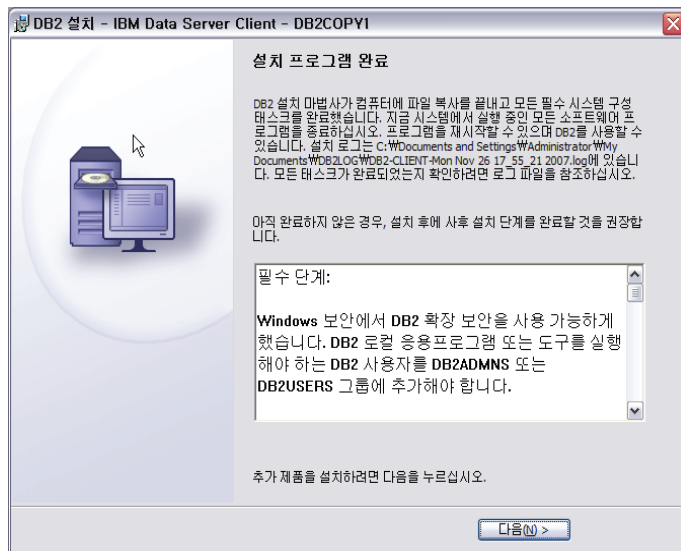
11. 이제까지 지정하였던 설정값을 확인한 후 "완료" 버튼을 클릭합니다. 설정값을 응답파일에 저장한 후 설치를 시작합니다.



10. 설치한 DB2 Client 파일, 기타 오브젝트에 대해 운영체제 보안을 사용할 경우 운영체제 보안 사용에 체크한후 관리자 그룹 정보 및 사용자 그룹 정보를 입력합니다.



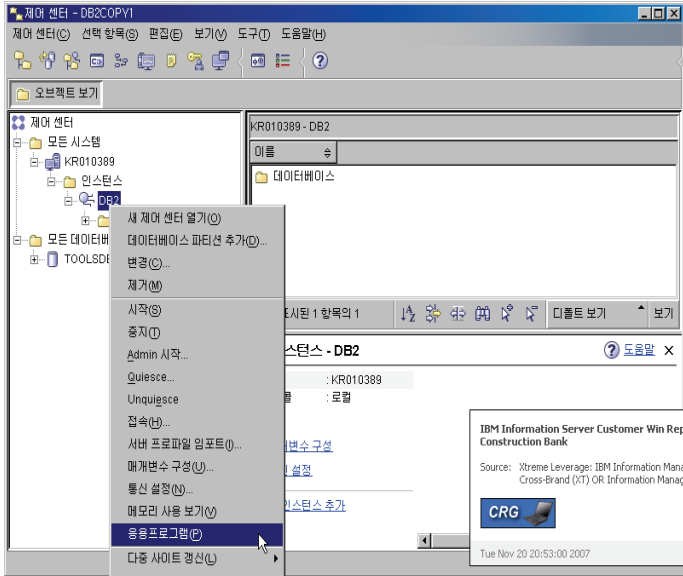
12. 설치가 완료되면 설치로그의 파일 위치 및 메시지, 그리고 사후 조치에 대한 메시지를 확인합니다. "다음" 버튼을 클릭하여 IBM Database Add Ins for Visual Studio 2005를 추가로 설치할 수 있습니다. "완료" 버튼을 클릭하여 클라이언트 설치를 종료합니다.



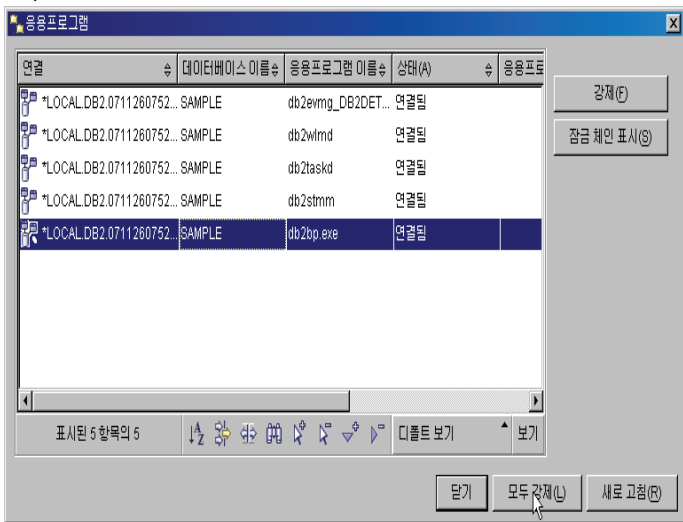
18 DB2 제품 제거 - 윈도우

Point DB2 서버 제품이 제거될 플랫폼은 윈도우 환경입니다.

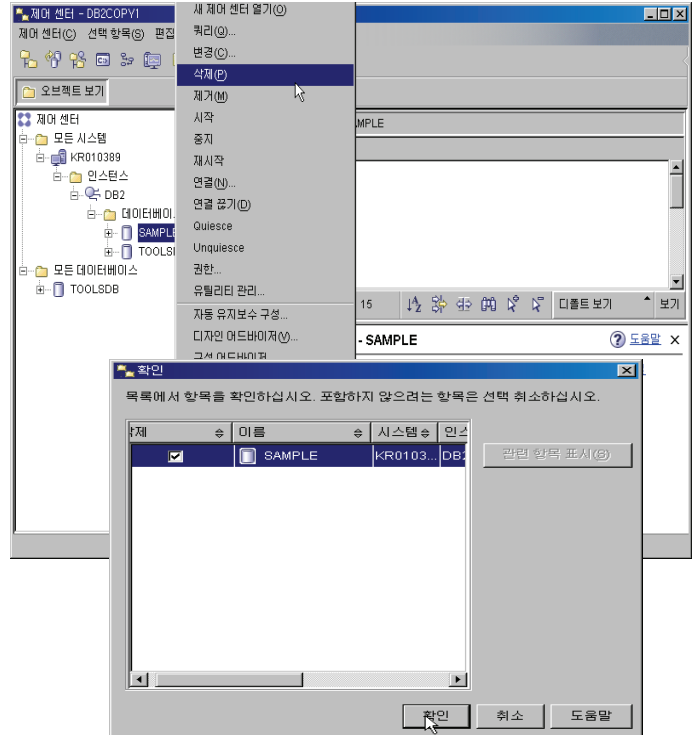
1. 시스템에 DB2 버전이 여러 개 설치되어 있고 디폴트로 지정된 버전을 제거 할 경우에는 제거를 진행하기 전에 db2swtch 명령어를 이용하여 디폴트로 사용할 DB2의 다른 버전을 선택합니다.
2. 제어센터에서 아래그림과 같이 제거할 버전에 속한 인스턴스를 선택하여 오른쪽 버튼을 클릭한후 응용프로그램을 선택합니다.



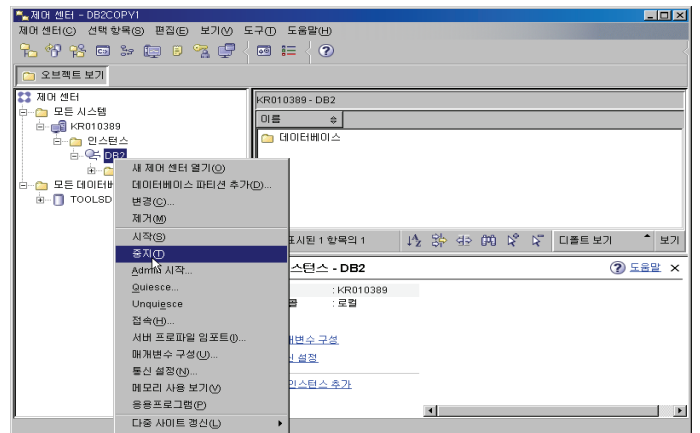
3. 아래그림과 같이 인스턴스에 접속한 응용프로그램이 나타납니다. "모두 강제" 버튼을 눌러 응용프로그램을 강제로 접속을 끊을 수 있지만 만약 응용프로그램이 트랜잭션을 수행중에 있을 경우에는 접속한 사용자 및 응용 프로그램 담당자에게 먼저 트랜잭션을 중지하고 접속을 끊도록 권유합니다. 강제로 접속을 끊을 경우에는 수행중이었던 모든 트랜잭션은 롤백됩니다.



4. 제어 센터에서 제거하고자 하는 버전의 인스턴스내에 위치한 데이터베이스를 모두 삭제(Drop) 합니다. SAMPLE 데이터베이스를 선택한후 오른쪽 버튼을 누르면 서브 메뉴가 나타납니다. 서브 메뉴중 삭제를 선택한 후 삭제 화면 창이 나타나면 아래와 같이 "확인" 버튼을 클릭합니다. 이곳에서는 DB2 인스턴스에 SAMPLE 및 TOOLSDB가 존재하므로 TOOLSDB도 위와 같은 과정으로 삭제합니다.



5. 제어센터에서 인스턴스를 선택한 후 오른쪽 버튼을 눌러 서브메뉴가 나타나면 중지를 선택합니다. 이는 인스턴스를 중지하는 명령어로 DB2 제품을 제거하기 전에 DB2 서비스 및 프로세스가 중지되어 있지 않으면 경고 메시지가 수신됩니다.



Point DB2 서버 제품이 제거될 플랫폼은 윈도우 환경입니다.

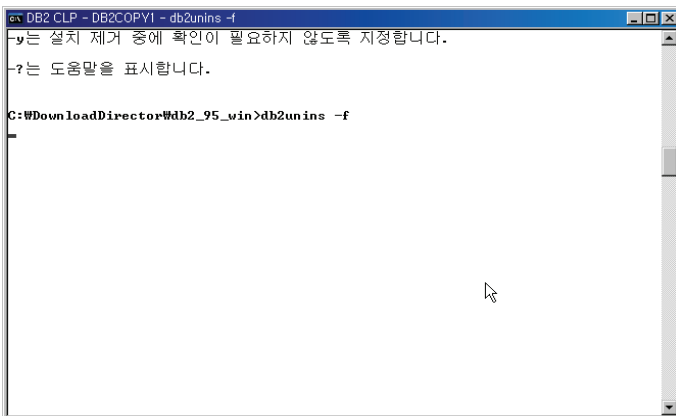
6. 다음과 같이 두가지 옵션중 선택하여 DB2 제품을 제거합니다.

- 제어판의 프로그램 추가/제거



- db2unins 명령어 :

- db2unins -p : 복수의 DB2 제품을 제거합니다.
- db2unins -f : 시스템에서 모든 DB2 사본의 설치를 강제로 제거합니다.
- db2unins -d : 현재 DB2 사본에 설치된 DB2 제품의 전체 목록을 확인합니다.



7. 제거가 완료됩니다.

Point DB2 서버 제품이 제거될 플랫폼은 AIX 입니다.

- inst95 사용자로 로그인합니다. "list applications" 명령어를 이용하여 현재 인스턴스에 접속한 애플리케이션을 확인합니다. 만약 아래와 같이 데이터베이스 매니저가 시작되지 않았다는 메시지가 나타나면 인스턴스를 시작한 후 다시 "list applications" 명령어를 수행합니다. 이 명령어는 현재 데이터베이스에 접속한 사용자 및 응용 프로그램이 무엇인지 확인해 줍니다. 접속한 사용자 및 응용프로그램 담당자에게 먼저 트랜잭션을 중지하고 접속을 끊도록 권유합니다. 강제로 접속을 끊을 경우에는 수행중이었던 모든 트랜잭션은 롤백됩니다. 이곳에서는 "force applications all" 명령어를 이용하여 접속한 모든 응용프로그램의 연결을 강제로 끊습니다. 다시한번 "list applications" 명령어를 이용하여 접속한 응용프로그램이 있는지 확인합니다.

```

sncp6303@root:/opt/IBM/db2/U9.5/adm>su - inst95
$ db2start
06/11/2008 09:30:41      0      0      SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
$ db2 connect to sample

Database Connection Information

Database server          = DB2/AIX64 9.5.0
SQL authorization ID    = INST95
Local database alias    = SAMPLE

$ db2 list applications

Auth Id  Application  Appl.  Application Id
Name     Handle      DB     # of
-----  -----  ---
INST95   db2bp       7      *LOCAL.inst95.080611003047
        SAMPLE   1

$ db2 force applications all
DB20000I The FORCE APPLICATION command completed successfully.
DB21024I This command is asynchronous and may not be effective immediately.
  
```

- "list db database" 명령어를 이용하여 제거하고자 하는 버전의 인스턴스 내에 존재하는 데이터베이스를 확인한 후 "Drop Database <DB Name>" 을 이용하여 모두 삭제합니다.

```

sncp6303@root:/opt/IBM/db2/U9.5/adm>db2 list db directory

System Database Directory

Number of entries in the directory = 1

Database 1 entry:

Database alias          = SAMPLE
Database name          = SAMPLE
Local database directory = /home/inst95
Database release level  = c.00
Comment                 =
Directory entry type    = Indirect
Catalog database partition number = 0
Alternate server hostname =
Alternate server port number =

$ db2 drop db sample_
  
```

- db2stop 명령어를 이용하여 제거하고자 하는 버전의 인스턴스를 모두 중지합니다. 다시 root 사용자로 로그인하여 DB2DIR/instance 디렉토리에 위치한 db2idrop 명령어를 이용하여 인스턴스를 삭제합니다.

```

sncp6303@root:/opt/IBM/db2/U9.5/adm>db2stop force
2008-06-11 09:37:30      0      0      SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.
$ exit
sncp6303@root:/opt/IBM/db2/U9.5/adm>cd ../instance
sncp6303@root:/opt/IBM/db2/U9.5/instance>./db2idrop inst95
  
```

- db2ls 명령어를 이용하여 설치된 DB2 디렉토리를 확인합니다. DB2DIR/install 디렉토리에서 "db2_deinstall -a" 명령어를 실행하여 현재 위치에 설치된 DB2 제품을 제거합니다. 여기서 DB2DIR은 DB2 제품을 설치할 때 지정한 위치입니다.

```

sncp6303@root:/opt/IBM/db2/U9.5/instance>db2ls

Install Path          Level  Fix Pack  Special Install Number
Install Date          Installer UID

-----
/DB2/U9.5             9.5.0.0    0
Wed Feb 13 17:12:25 2008 KORST 0
/opt/IBM/db2/U9.5     9.5.0.0    0
Wed Jun 11 08:00:26 2008 KORST 0
sncp6303@root:/opt/IBM/db2/U9.5/instance>cd ../install
sncp6303@root:/opt/IBM/db2/U9.5/install>./db2_deinstall -a
DBI1016I Program db2_deinstall is performing uninstallation. Please wait.
  
```

- 제거가 완료됩니다.

@ DB2 설치 관련 Q/A

- ❶ AIX에 DB2 Enterprise Server Edition을 설치하는데 SA MP Base Component가 설치되지 않았다는 메시지를 받았습니다. 일부 기능을 사용할 수 없다고 하는데 DB2 Server 를 사용하는데 있어서 문제가 발생하는지요? 이 제품을 반드시 설치해야 되나요?
SA MP Base Component는 티볼리 제품으로 AIX 또는 Linux를 위해 고가용성 기능을 제공합니다. 티볼리의 SA MP Base Component는 AIX 및 Linux의 IBM Data Server 제품군에 번들되어 DB2의 HA (High Availability) 기능을 한층 더 강화시켜주며 AIX 및 Linux의 IBM Data Server의 Cluster 환경을 관리합니다.

DB2 Enterprise Server Edition 설치 중 SA MP Base Component 설치가 실패할 경우 에러의 원인을 확인하여 조치한 후 향후에 설치가 가능합니다. DB2 CD 또는 이미지 디렉토리로 이동하여 하위 디렉토리인 db2/aix/tsamp로 이동하여 installSAM을 실행하면 됩니다.
- ❷ 오라클은 설치전에 하드웨어 및 소프트웨어 설치 요구사항 , 환경변수 지정 및 파일 시스템 생성, OS 사용자 지정등 설치전 준비사항이 많습니다. DB2는 설치전 준비사항이 하드웨어 및 소프트웨어 설치 요구사항외에는 없는지요?
하드웨어 및 소프트웨어 설치요구사항외에는 별다른 설정이 필요 없습니다. DB2는 설치시 환경변수 설정을 자동으로 설정합니다. 이는 DBA 및 시스템 관리자의 업무를 최소한으로 줄여줍니다.
- ❸ 설치 성공여부 및 설치 관련한 메시지를 확인하고 싶습니다. 어디에서 확인할 수 있는지요?
DB2 설치시 설치 메시지 파일의 위치는 윈도우 환경과 유닉스 환경이 상이합니다. 설치 플랫폼이 윈도우일 경우 c:\WDoments settings\Administrator\My Documents\WB2LOG\WB2-ESE-<날짜 및 시간>.log 이며 Unix 일 경우 주로 /tmp 하에 db2_install.log.<Process ID> 입니다.
- ❹ 오라클처럼 별도의 파일시스템을 생성하여 그 파일시스템에 DB2 소프트웨어를 설치하고 싶습니다. 가능한지요?
가능합니다. 앞에서 설치하실때 보신 것처럼 윈도우 및 Unix 모두 설치 디렉토리를 지정하는 설치 옵션이 있습니다. 이 곳에서 설치 디렉토리를 지정하시면 됩니다.
- ❺ 현재 사용하고 있는 시스템에는 DB2 V9.1이 설치되어 있습니다. DB2 V9.1을 그대로 사용하면서 DB2 V9.5를 설치하여 테스트해 볼 수 있는지요?
가능합니다. DB2 V9.1과 DB2 V9.5의 설치 디렉토리를 구별해 놓는다면 문제되지 않습니다. 또한 Fixpak을 바로 적용하기에는 위험이 있을 경우 별도의 디렉토리에 Fixpak을 설치하여 미리 테스트해보실 수 있습니다.
- ❻ 현재 DB2 Client V8.2가 설치되어 있습니다. 서버가 V9.5로 Upgrade 될 경우 Client를 재설치해야 되는건지요?
재설치 할 필요가 없습니다. DB2 Client v8 및 V9은 원격 DB2 서버 V9.1 , V9.5 에 액세스 가능합니다. 또한 IBM Data Server Client, IBM Data Server Runtime Client 및 IBM Data Server Driver for ODBC, CLI 및 .NET 버전 V9.5 는 DB2 V 9.1 및 DB2 UDB V8 서버에 액세스 가능합니다.

가장 최신 정보는 <http://www.ibm.com/software/data/udb/sysreq.html> 을 참조하시기 바랍니다.



Unit 04 : 용어 및 클라이언트 툴 비교


오라클 사용자를 위한 DB2 가이드

용어 비교

Command Line 툴 비교

DB2 명령행 처리기와 명령 편집기 소개



Point  오라클과 DB2의 아키텍처를 비교하기 전에 오라클과 DB2에서 사용되는 용어를 정확히 이해해야 합니다. 이 곳에서는 오라클과 DB2에서 사용되는 용어를 비교합니다.


Oracle	DB2	설명
인스턴스	인스턴스(DatabaseManager)	▪ 프로세스와 Shared Memory로 구성
데이터베이스	데이터베이스	▪ 데이터가 포함된 물리적인 구조
Data Buffers	Buffer Pool	▪ 디스크 I/O 를 줄이기 위해 테이블 공간에 있는 데이터 Block을 저장하는 메모리 공간
SGA	Database Manager와 Database Shared Memory	▪ 인스턴스 및 데이터베이스의 제어 정보를 가지는 공유 메모리 영역
Library Cache	Package Cache	▪ 실행계획과 실행 코드가 저장되는 영역
Large Pool	Utility Heap	▪ Backup, Restore, LOAD시 사용되는 메모리 영역
Data Dictionary Cache	Catalog Cache	▪ 데이터베이스의 Meta Data 정보를 저장하는 메모리로 오라클에서는 Data Dictionary, DB2에서는 Catalog 정보를 저장하는 영역
PL/SQL	SQL/PL	▪ 프로시저 및 함수를 작성하기 위해 SQL 과 함께 사용하는 구조화 프로그래밍 언어
Package	N/A	▪ PL/SQL Block을 하나로 묶어 그룹화한 오브젝트
N/A	Package	▪ static embedded SQL 애플리케이션을 사전에 컴파일하여 서버에 저장한 실행 계획
파라미터 파일	DBM Configuration File 및 Database Configuration File	▪ 인스턴스 및 데이터베이스의 구성 파라미터 값이 저장된 파일
Database Link	Federated System	▪ 현재 접속된 데이터베이스에서 네트워크상의 다른 데이터베이스에 접속을 하기 위한 방법
Redo Log Files	Transaction Log Files	▪ 에러 발생시 데이터들을 복구하기 위해 데이터베이스의 변경정보들을 가진 파일
테이블 공간	테이블 공간	▪ 데이터베이스 데이터를 저장하는 공간
Datafiles	Containers	▪ 테이블 공간을 구성하는 물리적인 파일
Segments	Objects	▪ 데이터베이스내에 생성되는 모든 객체
Extents	Extents	▪ 사용자가 지정한 수의 연속된 Block의 집합
Data blocks	Pages	▪ 데이터베이스내에서 데이터를 읽고, 쓰는 최소 단위
Data dictionary	System catalog	▪ Database의 메타 데이터를 저장하는 테이블



위의 표는 공통의 개념을 오라클과 DB2에서 다르게 사용하는 용어입니다. 이 외에도 다른 용어들이 있으나 주요 용어에 대해서만 비교하였습니다.

02 SQL 및 인터페이스 툴 비교

용어 및 클라이언트 툴 비교

Point  오라클과 DB2는 SQL언어를 이용하여 데이터를 처리합니다. 또한 SQL문을 수행하기 위해서는 SQL문을 해석할 수 있는 클라이언트 툴이 필요합니다. 이곳에서는 SQL의 정의 및 SQL 프로그램 언어, 클라이언트 툴을 간단히 비교합니다.

SQL 정의

- SQL 문은 다음과 같은 작업을 수행하는 언어입니다.
 - 테이블의 데이터를 조회합니다.
 - 테이블에 데이터를 삽입/변경/삭제 합니다.
 - 오브젝트 (테이블, 인덱스, 뷰, 스토어드 프로시저) 등을 생성/삭제/변경 합니다.
 - 데이터베이스 및 오브젝트에 대한 권한을 제어합니다.
 - DB2 와 오라클은 모두 ANSI 표준을 따르므로 SQL 문이 유사하지만 옵션에 따라서는 약간씩 달라질 수 있습니다.

오라클 PL/SQL 과 DB2 SQLPL 비교


	오라클 PL/SQL	DB2 SQLPL
정의	<ul style="list-style-type: none"> ▪ PL/SQL(Procedural Language/SQL)의 약어로 오라클 DB 환경에서 실행되는 절차적인 데이터베이스 프로그래밍 언어입니다. 	<ul style="list-style-type: none"> ▪ SQL PL(SQL Procedural Language)은 프로시저 및 함수를 작성하기 위해 SQL과 함께 사용하는 구조화 프로그래밍 언어로 복합 SQL문 및 프로시저를 작성하는 기능을 제공합니다.
표준 여부	<ul style="list-style-type: none"> ▪ 표준 SQL과 표준 3세대 언어의 일부 기능을 포함한 SQL의 확장언어. 	<ul style="list-style-type: none"> ▪ 표준 SQL/PSM(SQL Persistent Stored Modules) 언어의 서브 세트
명령문	<ul style="list-style-type: none"> ▪ 모든 SQL문 ▪ 변수 및 상수 등의 선언문 ▪ 대입문 ▪ 조건 판단문 ▪ 제어 흐름문 ▪ 반복 처리문 	<ul style="list-style-type: none"> ▪ 변수 관련 명령문 ▪ 조건문 ▪ 반복문 ▪ 제어 전달문 ▪ 오류 관리문 ▪ 결과 세트 조작문

참고)

Oracle PL/SQL과 DB2 SQLPL은 모두 서버 프로그램 작성을 위한 언어로 Oracle PL/SQL로 작성한 서버 프로그램을 DB2 SQLPL로 변환시킬 수 있습니다. 그러나 DB2 SQLPL이 표준 SQL/PSM을 따르는 반면 Oracle PL/SQL은 3세대 언어의 일부기능을 포함해 독자적으로 SQL언어를 확장하였기 때문에 DB2 SQLPL과 Oracle PL/SQL의 Syntax 차이가 있습니다.

툴 비교

	오라클	DB2
Command Line Interface	<ul style="list-style-type: none"> ▪ 오라클 SQL*Plus ▪ 오라클 iSQL*Plus 	<ul style="list-style-type: none"> ▪ DB2 명령행 처리기 ▪ DB2 명령창
그래픽 툴	<ul style="list-style-type: none"> ▪ 오라클 Enterprise Manager 	<ul style="list-style-type: none"> ▪ 제어 센터 ▪ 명령 편집기
Thrid party 툴	<ul style="list-style-type: none"> ▪ 퀘스트사의 Toad 	<ul style="list-style-type: none"> ▪ 퀘스트사의 Toad for DB2

Point  다음은 텍스트 기반으로 SQL문을 실행할 수 있는 Command line 툴입니다. 오라클에서는 Sql*Plus라 하며 DB2에서는 Command Line Processor라 합니다. 이 툴의 이용 방법에 대해 간단히 비교 합니다.

	오라클	DB2
인터페이스	sqlplus	clp (command line processor) 또는 GUI의 명령센터
실행 방법	<pre>/home/inst95> sqlplus / as sysdba SQL> select * from emp;</pre>	<p>대화식 모드와 비대화식 모드가 존재합니다. 대화식 모드는 DB2 전용 세션으로 DB2 명령어를 실행할 수 있으며 비대화식 모드는 OS 상에서 DB2 명령어를 실행할 수 있습니다. 비대화식 모드는 OS 프롬프트에서 DB2 명령어앞에 db2를 입력하고 수행하면 됩니다. 오라클은 비대화식 모드가 존재하지 않습니다.</p> <p>대화식 모드 :</p> <pre>/home/inst95> db2 db2=> select * from emp db2=></pre> <p>비대화식 모드 :</p> <pre>/home/inst95/>db2 "select * from emp" /home/inst95/>ls -la</pre>
세션 종료 방법	SQL>quit	<pre>db2=> quit</pre> <p>→데이터베이스 접속은 유지되며 DB2프롬프트 상태만 벗어납니다.</p> <pre>db2=> terminate</pre> <p>→데이터베이스 접속 및 백그라운드 프로세스가 종료됩니다.</p>
스크립트 실행 방법	SQL> @test.sql	<pre>/home/inst95> db2 -tvf <FileName></pre> <p>설명)</p> <ul style="list-style-type: none"> • -t: 명령문 구분자로 세미콜론 (;)을 사용하는 것을 지정합니다. • -v: 화면에 결과값을 보여줍니다. • -f < 파일 이름 >: 읽을 파일을 지정합니다.
CLP 명령어	<pre>SQL> help set SQL> EDIT SQL> !<OS 명령어></pre>	<pre>db2=> help [명령문] db2=> EDIT db2=>!<OS 명령어> db2=> HISTORY db2=> runcmd <History 목록 번호></pre> <p>참고) help를 입력할 경우 DB2 정보센터에 연결됩니다.</p>
도움말	SQL> ?	<pre>db2=> ? < DB2 명령어 ></pre> <p>→DB2 명령어에 대한 구문과 옵션을 확인할 수 있습니다.</p> <pre>db2=> ? <SQLCODE></pre> <p>→ SQL 문 실패 시 SQLCODE값과 에러 원인 및 해결 방안을 제시합니다.</p>

04 DB2 명령행 처리기

Point DB2 명령행 처리기를 이용하여 SQL문을 수행합니다.

1. 시작 => 모든 프로그램 => IBM DB2 => DB2COPY1(디폴트) => 명령행 도구 => 명령행 처리기를 클릭합니다. 세션이 시작되면 "db2=>" 라는 전용 프롬프트가 나타납니다. 또는 명령창에서 db2cmd를 실행하여 DB2 환경이 세팅된 창을 띄우도록 합니다. 이 창에서 db2 명령어를 실행해도 됩니다. 이 곳에서 "connect to sample" 명령어를 실행합니다.

```

c:\> DB2 CLP - DB2COPY1 - db2
C:\Documents and Settings\Administrator>db2
(c) Copyright IBM Corporation 1993,2007
DB2 Client 9.5.0용 명령행 처리기(CLP)

명령 프롬프트에서 데이터베이스 관리 프로그램 명령과 SQL문을 받
예:
  db2 => connect to sample
  db2 => bind sample.bnd

일반적인 도움말을 보려면 ?를 입력하십시오.
명령 도움말을 보려면 ?를 입력하십시오.
여기서 명령은 데이터베이스 관리 프로그램 명령의 처음 몇 키워드
있습니다. 예:
? CATALOG DATABASE: CATALOG DATABASE 명령에 대한 도움말
? CATALOG:       모든 CATALOG 명령에 대한 도움말

db2 대화식 모드에서 나가려면 명령 프롬프트에서 QUIT를 입력하십
스.
모드 외부에서는 모든 명령 앞에 'db2'를 붙여야 합니다.
현재 명령 옵션 설정값을 나열하려면 LIST COMMAND OPTIONS를 입력
하십시오.

더 자세한 도움말을 보려면 온라인 참조서를 참조하십시오.

db2 => connect to sample

데이터베이스 연결 정보
  
```

2. SAMPLE Database에 접속한 후 "list tables" 라는 명령어를 실행하여 현재 접속한 사용자의 스키마에 속하는 테이블이 존재하는지 확인합니다.

```

c:\> DB2 CLP - DB2COPY1 - db2
db2 =>
db2 => list tables

테이블/뷰                                스키마
-----
유형   작성 시간
-----
ACT                                T      2007-11-
ADEFUSR                             S      2007-11-
BUFFERPOOL_EYMON                    T      2007-12-
CL_SCHED                             T      2007-11-
CONNHEADER_EYMON                    T      2007-12-
CONNMEMUSE_EYMON                    T      2007-12-
CONN_EYMON                          T      2007-12-
CONTROL_EYMON                       T      2007-12-
  
```

3. db2 세션에서 OS 명령어를 실행하고자 할 경우에는 "!<OS 명령어>" 를 실행하면 됩니다. 간단히 아래와 같이 파일 목록을 보여주는 dir 명령어를 실행하도록 합니다.

```

c:\> DB2 CLP - DB2COPY1 - db2
db2 =>
db2 => !dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: F0C9-11EF

C:\Documents and Settings\Administrator <디렉터리>

2007-12-11 오후 04:04 <DIR>      .
2007-12-11 오후 04:04 <DIR>      ..
2007-08-15 오전 07:34 <DIR>      Bluetooth Software
2007-10-23 오후 04:29 <DIR>      Contacts
2007-05-30 오후 08:12 <DIR>      desktop
2007-11-14 오후 04:10           30 dlmgr_.pro
2007-12-18 오후 06:07 <DIR>      Favorites
2007-08-31 오전 02:56 <DIR>      IBM
2007-12-19 오후 10:08 <DIR>      My Documents
2007-12-07 오전 11:35 <DIR>      SametimeTranscripts
2007-12-11 오후 04:05           136,499,200 SAMPLE_0.DB2.NODE0000.CATN0000.2007
160454.001
2007-04-23 오후 03:03 <DIR>      WINDOWS
2007-12-20 오전 11:10 <DIR>      바탕 화면
  
```

4. DB2 구문을 잘 모를 경우 아래와 같이 "?" 와 함께 사용하여 SQL문 또는 DB2 명령어에 대해 확인할 수 있습니다. 아래에서는 "list" 명령어에 대한 구문을 확인합니다.

```

c:\> DB2 CLP - DB2COPY1 - db2
db2 =>
db2 => ? list tablespaces
LIST TABLESPACES [SHOW DETAIL]

db2 => ? list
LIST ACTIVE DATABASES [AT DBPARTITIONNUM db-partition-number | GLOBAL]
LIST APPLICATIONS [FOR DATABASE database-alias]
[AT DBPARTITIONNUM db-partition-number | GLOBAL] [SHOW DETAIL]

LIST COMMAND OPTIONS
LIST DATABASE DIRECTORY [ON drive]
LIST DATABASE PARTITION GROUPS [SHOW DETAIL]
LIST DBPARTITIONNUMS
LIST DCS APPLICATIONS [SHOW DETAIL | EXTENDED]
LIST DCS DIRECTORY
  
```

Point DB2 명령행 처리기를 이용하여 SQL문을 수행합니다.

5. "a.sql" 라는 파일에 DB2 테이블 T1, T2, T3를 생성하는 SQL문을 입력하고 저장합니다. 그리고 SAMPLE 데이터베이스에 접속한 후 아래와 같이 "-tvf" 라는 옵션을 주어 스크립트를 수행합니다.

```

c
$ cat a.sql
Create table t1 ( c1 int);
create table t2 ( c1 int);
create table t3 ( c1 int);
$ db2 -tvf a.sql
Create table t1 ( c1 int)
DB20000I The SQL command completed successfully.

create table t2 ( c1 int)
DB20000I The SQL command completed successfully.

create table t3 ( c1 int)
DB20000I The SQL command completed successfully.
$
  
```

6. 아래와 같이 "insert 문"이 실패하여 SQL0408N 에러가 발생하였습니다. "? <SQLCODE>"를 이용하여 좀 더 자세한 에러 사항을 확인할 수 있습니다.

```

c
db2 => insert into t1 values ('1')
DB21034E The command was processed as an SQL statement because
valid Command Line Processor command. During SQL processing it
SQL0408N A value is not compatible with the data type of its as
target. Target name is "C1". SQLSTATE=42821
db2 => ? SQL0408N

SQL0408N A value is not compatible with the data type of its as
target. Target name is "<name>".

Explanation:

The data type of the value to be assigned to the column, paramet
variable, or transition variable by the SQL statement is incompa
with the declared data type of the assignment target. Both must
* Numeric
* Character
* Graphic
* Dates or character
* Times or character
  
```

7. "EDIT <HISTORY 목록 번호>" 명령어를 이용하여 앞에서 수행한 "insert 문"을 다시 입력할 필요없이 필요한 부분만 수정하여 실행할 수 있습니다.

```

c:\
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 => history
1 insert into t1 values ('1')
2 ? SQL0408N
3 history
4 history
5 list tables
6 history
7 list tables
8 history
9 history
db2 => EDIT 1
* this release of the AIX Operating System.
  
```

8. "history" 명령어를 이용하여 현재까지 실행했던 명령어 목록을 조회할 수 있으며 "runcmd <목록 번호>" 를 이용하여 그 명령어를 재실행 할 수 있습니다.

```

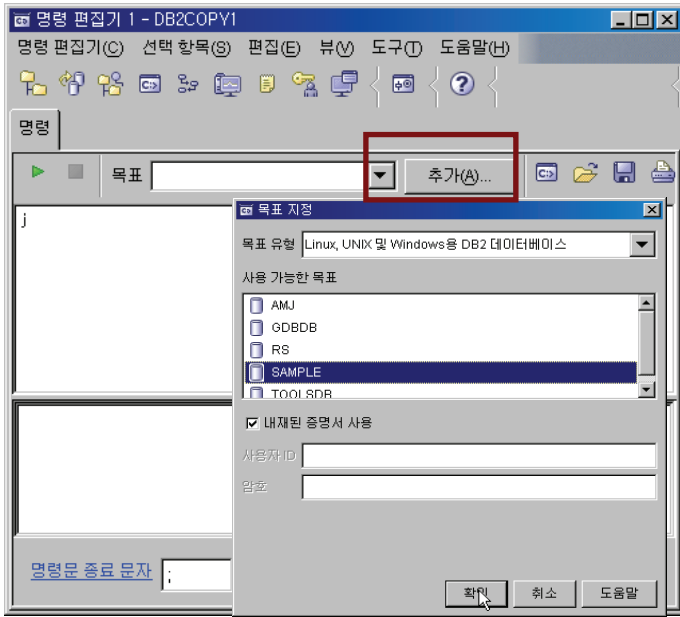
c
db2 => history
1 insert into t1 values ('1')
2 ? SQL0408N
3 history
4 history
5 list tables
6 history
db2 => runcmd 5
db2 => list tables

Table/View          Schema      Type  Creation
-----
ACT                 INST95     T     2007-11-
ADEFUSR            INST95     S     2007-11-
CL_SCHED           INST95     T     2007-11-
DEPARTMENT         INST95     T     2007-11-
DEPT               INST95     A     2007-11-
EMP               INST95     A     2007-11-
EMPACT            INST95     A     2007-11-
EMPLOYEE          INST95     T     2007-11-
  
```

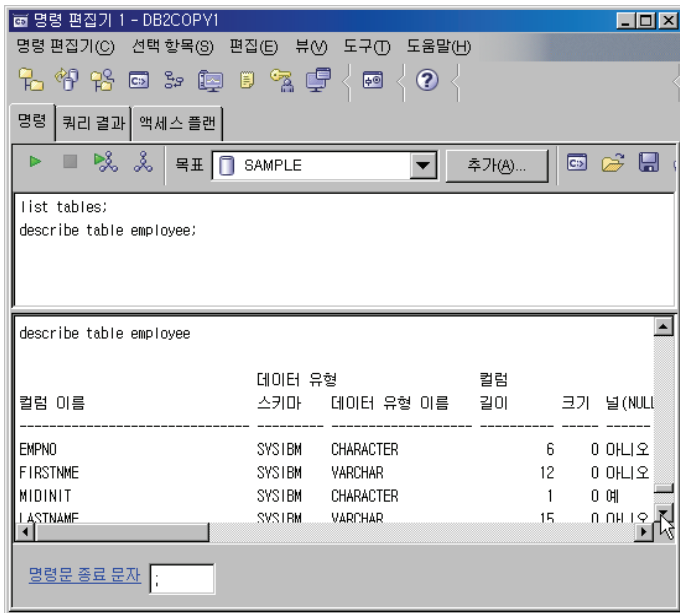

06 명령 편집기

Point DB2 명령 편집기를 이용하여 GUI 환경에서 SQL문을 쉽게 사용할 수 있습니다.

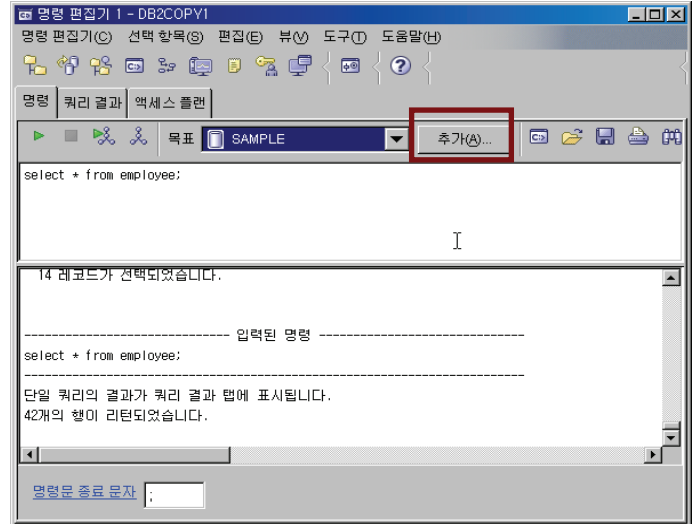
1. 시작 => 모든 프로그램 => IBM DB2 => DB2COPY1(디폴트) => 명령행 도구 => 명령 편집기를 클릭합니다. 아래와 같이 명령 편집기가 나타나면 추가 버튼을 눌러 접속할 데이터베이스를 선택하고 원격 데이터베이스일 경우 사용자 이름과 암호를 입력합니다.



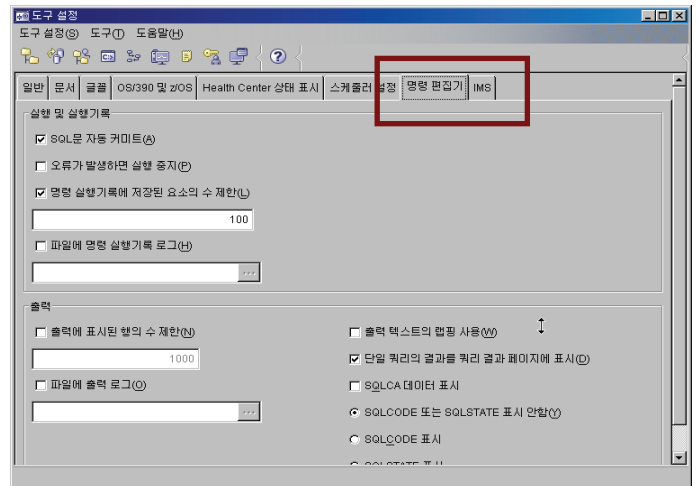
2. "list tables" 와 "describe table employee" 명령문을 실행하여 테스트 합니다. 아래 화면은 실행한 명령문의 결과값입니다.



3. 오른쪽의 아이콘을 보면 수행한 명령문을 파일에 저장할 수도 있으며 스크립트 파일을 열어 수행할 수도 있습니다.



4. 메뉴에서 도구 => 도구 설정을 선택한 후 명령 편집기 탭으로 이동하면 이곳에서 명령 편집기 환경을 설정할 수 있습니다.



📍 클라이언트 관련 Q/A

- 오라클은 Dummy 터미널에서 결과값을 보여주는 화면을 제어하는 명령어가 존재합니다. 예를 들면 "COLUMN <Column 명> FORMAT A10" 과 같은 명령어를 수행할 경우 지정된 column의 값이 10글자만 보여줄 수 있도록 조정합니다. DB2는 이러한 결과화면에 대해 설정해 주는 부분이 없지요?

안타깝게도 결과 화면에 대해 칼럼과 라인등을 제어할 수 있는 명령어는 없습니다. 그러나 실행시 -r 옵션을 사용할 경우 결과값을 파일로 저장하여 확인할 수 있습니다. 또한 DB2 Client는 작은 사이즈로 GUI 를 모두 제공하므로 명령 센터에서 확인하면 좀 더 편리하게 확인 가능합니다.



Unit 05 : 인스턴스 및 데이터베이스 비교

오라클 사용자를 위한 DB2 가이드

인스턴스 비교


DB2 인스턴스 관리

데이터베이스 비교

파라미터 비교

DB2 환경 변수 및 파라미터 설정



Point  오라클과 DB2의 아키텍처를 비교하기 전에 오라클과 DB2에서 사용되는 용어를 정확히 이해해야 합니다. 이곳에서는 오라클과 DB2에서 사용되는 용어를 비교합니다.

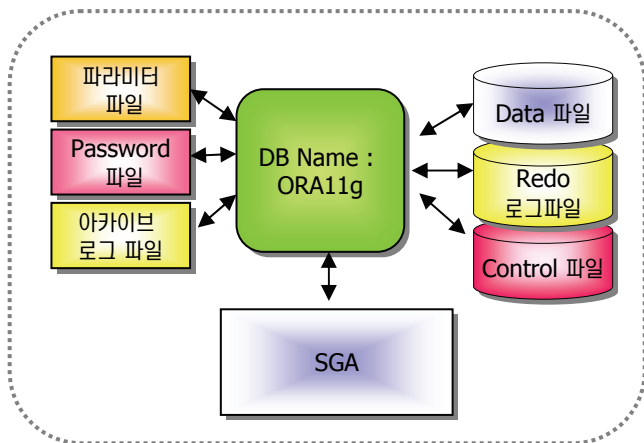
Oracle	DB2	설명
인스턴스	인스턴스(DatabaseManager)	• 프로세스와 Shared Memory 로 구성
데이터베이스	데이터베이스	• 데이터가 포함된 물리적인 구조
Data Buffers	Buffer Pool	• 디스크 I/O를 줄이기 위해 테이블 스페이스에 있는 데이터 Block을 저장하는 메모리 공간
SGA	Database Manager 와 Database Shared Memory	• 인스턴스 및 데이터베이스의 제어 정보를 가지는 공유 메모리 영역
Library Cache	Package Cache	• 실행계획과 실행 코드가 저장되는 영역
Large Pool	Utility Heap	• Backup, Restore, LOAD 시 사용되는 메모리 영역
Data Dictionary Cache	Catalog Cache	• 데이터베이스의 Meta Data 정보를 저장하는 메모리로 오라클에서는 Data Dictionary 정보를, DB2에서는 Catalog 정보를 저장하는 영역
sqlplus	command line processor(clp)	• Client 에서 데이터베이스에 접속시 사용되는 client 입니다.
PL/SQL	SQL/PL	• Procedure 및 함수를 작성하기 위해 SQL과 함께 사용하는 구조화 프로그래밍 언어
Package	N/A	• PL/SQL Block을 하나로 묶어 그룹화한 오브젝트
N/A	Package	• static embedded SQL 애플리케이션을 사전에 컴파일하여 서버에 저장한 실행 계획
파라미터 파일	DBM Configuration File 와 Database Configuration File	• 인스턴스 및 데이터베이스를 구성하기 위한 파라미터값을 가지고 있는 파일
Database Link	Federated System	• 현재 접속된 데이터베이스에서 네트워크상의 다른 데이터베이스에 접속을 하기 위한 방법
Redo Log Files	Transaction Log Files	• 에러 발생시 데이터들을 복구하기 위해 데이터베이스의 변경정보들을 가진 파일
Alert Log	Administration Notification Log 및 DB2Diag Log	• DBMS에 문제 발생시 문제 사항을 기록하는 진단 파일
Archive Redo Log	Archive Log	• 복구 시에 필요한 오프라인 파일
Autoextend	Autoresize	• 테이블 스페이스의 크기를 자동으로 재 조정 해주는 옵션
Materialized View	Materialized Query Table (MQT)	• 복잡한 쿼리, 특히 다량의 데이터 분석과 연관된 쿼리에 대한 성능 및 응답 시간을 향상 시키는 사전 집계된 데이터
테이블 스페이스	테이블 스페이스	• 데이터베이스 데이터를 저장하는 공간
Datafiles	Containers	• 테이블 스페이스를 구성하는 물리적인 파일
Segments	Objects	• 데이터베이스 내에 생성되는 모든 객체
Extents	Extents	• 사용자가 지정한 수의 연속된 Block 의 집합
Data blocks	Pages	• 데이터베이스내에서 데이터를 읽고 쓰는 최소 단위
Data dictionary	System catalog	• Database의 메타 데이터를 저장하는 테이블

02 인스턴스 비교

Point 인스턴스란 데이터베이스를 운영하는데 필요한 환경입니다. 오라클에서는 인스턴스내에 한 개의 데이터베이스만 생성이 가능하므로 대부분의 오라클 DBA 또는 사용자들은 인스턴스와 데이터베이스가 동일하다고 생각하지만 사실은 다른 개념입니다. DB2는 인스턴스와 데이터베이스의 개념이 엄격히 구별됩니다. 또한 인스턴스내에 여러 개의 데이터베이스를 생성할 수 있습니다.

오라클

Oracle SID = ORA11g

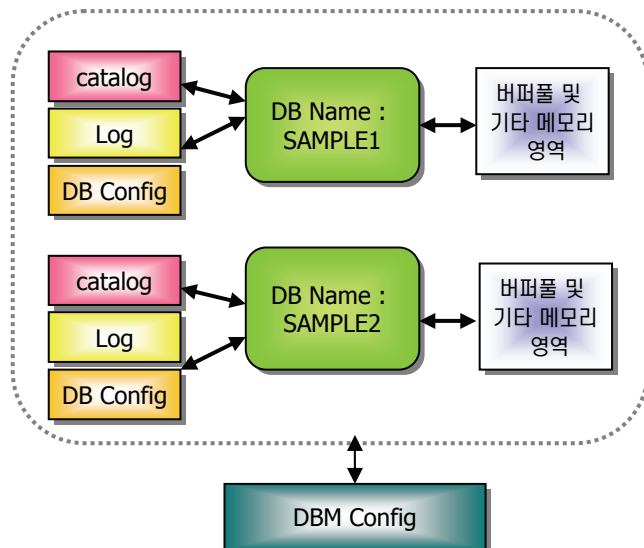


```
SET ORACLE_SID=ORA11g
SQL> Connect / as sysdba
SQL> startup
```

- 오라클에서는 인스턴스를 Oracle SID 라고도 합니다.
- 오라클에서는 하나의 인스턴스안에 하나의 데이터베이스만 생성이 가능합니다.
- 환경변수를 통해 인스턴스를 지정합니다.
- 인스턴스 및 데이터베이스의 구성 파라미터 파일인 PFILE 및 SPFILE이 존재합니다.

DB2

DB2 Instance = db2inst1




```
SET DB2INSTANCE=db2inst1
db2start
```

- DB2에서는 인스턴스를 Database Manager라고도 합니다.
- DB2에서는 하나의 인스턴스안에 여러 개의 데이터베이스를 생성할 수 있습니다.
- 환경변수 및 Registry 변수를 통해 인스턴스를 지정합니다.
- 인스턴스의 구성 파라미터 파일인 DBM Configuration File이 존재합니다.
- 각 데이터베이스별로 DB Config 파일이 존재합니다.

* 부연설명

- DB2의 인스턴스 및 데이터베이스 개념은 오라클의 DBA에게는 생소할 수 있습니다. DB2의 경우 오라클과는 달리 하나의 인스턴스안에 여러 개의 데이터베이스를 생성할 수 있으며 인스턴스를 구성하는 프로세스 및 메모리와 데이터베이스를 구성하는 프로세스 및 메모리가 구별되어 있습니다. 하지만 오라클의 경우 인스턴스는 메모리와 프로세스로 구성되어 있지만 데이터베이스의 경우는 물리적인 저장구조로 인식되어 있습니다.
- 오라클의 경우 인스턴스를 startup 하면 모든 프로세스 및 SGA가 할당 및 시작되며 데이터베이스의 관련 파일도 Mount 및 Open 되어 읽혀집니다. 하지만 DB2의 경우 인스턴스를 시작하면 인스턴스에 해당하는 프로세스와 메모리만 할당 및 시작될 뿐 데이터베이스의 관련 프로세스 및 메모리는 시작되지 않습니다. DB2는 Activate <Database Name> 라는 명령어를 통해 데이터베이스를 시작하거나 혹은 데이터베이스에 제일 먼저 접속하는 첫번째 사용자가 있을 경우 자동으로 데이터베이스를 시작합니다.

Point  오라클과 DB2 모두 인스턴스를 제어하는 명령어가 존재합니다. 여기서는 인스턴스를 제어하는 명령어를 비교합니다.

	오라클	DB2
인스턴스 생성 방법	<ul style="list-style-type: none"> 환경 변수에 ORACLE_SID 및 인스턴스 환경변수를 지정합니다. 	<ul style="list-style-type: none"> db2icrt 명령어를 통해 인스턴스를 생성합니다. * 인스턴스 생성방법은 뒷장에서 살펴 보도록 합니다.
인스턴스 제거 방법	<ul style="list-style-type: none"> 환경변수에 ORACLE_SID를 제거합니다. 	<ul style="list-style-type: none"> db2idrop 명령어를 통해 인스턴스를 제거합니다. 인스턴스를 제거 할때 데이터베이스는 제거되지 않으므로 인스턴스를 재 생성할 경우 데이터베이스를 사용할 수 있습니다. * 인스턴스 제거방법은 뒷장에서 살펴 보도록 합니다.
인스턴스 확인 방법	<ul style="list-style-type: none"> ps -ef grep pmon 이라는 OS 명령어를 통해 확인하거나 /etc/oratab 파일에서 확인합니다. 	<ul style="list-style-type: none"> db2ilist 이라는 명령어를 수행합니다. * 인스턴스 확인 방법은 뒷장에서 살펴 보도록 합니다.
인스턴스 지정 방법	<ul style="list-style-type: none"> 환경 변수에 지정합니다. <pre>SET ORACLE_SID = < Instance Name></pre>	<ul style="list-style-type: none"> 환경 변수에 지정합니다. <pre>SET DB2INSTANCE = < Instance Name></pre>
인스턴스 시작 방법	<ul style="list-style-type: none"> sqlplus에서 startup 명령어를 실행합니다. <pre>SQL> startup [NOMOUNT MOUNT OPEN]</pre>	<ul style="list-style-type: none"> DB2 명령어들의 Path가 등록되어 있을 경우 OS 명령행에서 다음 명령어를 실행합니다. <pre>/home/inst32/db2start [admin mode]</pre>
인스턴스 중지 방법	<ul style="list-style-type: none"> sqlplus에서 shutdown 명령어를 실행합니다. <pre>SQL> shutdown [NORMAL TRANSACTIONAL IMMEDIATE ABORT]</pre>	<ul style="list-style-type: none"> DB2 명령어들의 Path가 등록되어 있을 경우 OS 명령행에서 db2stop 명령어를 실행합니다. <pre>/home/inst32/db2stop [force]</pre>
버전 확인 방법	<ul style="list-style-type: none"> SQL 문을 통해 오라클의 버전 정보를 확인할 수 있습니다. <pre>SQL> select banner from v\$version</pre>	<ul style="list-style-type: none"> db2level 명령어를 통해 DB2 버전 정보를 확인할 수 있습니다. <pre>/home/db2inst1> db2level</pre>
제품 확인 방법	<ul style="list-style-type: none"> SQL 문을 통해 시스템에 설치된 오라클 제품 및 특성을 나열합니다. <pre>SQL> opatch lsinventory -all</pre>	<ul style="list-style-type: none"> db2ls 명령어를 통해 시스템에 설치된 DB2 제품 및 특성을 나열합니다. <pre>/home/db2inst1> db2ls</pre>

부연설명

- 오라클의 startup nomount 명령어를 수행하는 과정이 DB2 인스턴스의 시작 (즉, db2start 명령어 수행)에 해당되며 어떤 옵션 없이 오라클의 startup을 수행할 때의 과정은 db2start 및 Activate <Database Name> 두 과정으로 나뉘어 있다고 볼 수 있습니다.
- 오라클 shutdown시 immediate 옵션을 주면 rollback을 수행하고 현재 실행되고 있는 모든 세션을 끊은 후 shutdown을 합니다. 이에 해당하는 DB2의 옵션이 force라는 옵션입니다. db2stop을 수행할때 application이 데이터베이스에 연결되어 있을 경우에는 중지되지 않으므로 force 옵션을 사용하여 중지합니다.

Point DB2 인스턴스 생성시 윈도우와 Linux/Unix 간에 차이점이 있습니다. 플랫폼별 차이점을 유의하여 인스턴스를 생성하시기 바랍니다.

	Unix/Linux	윈도우
인스턴스 명령어	<ul style="list-style-type: none"> db2icrt -u < FencedID > < InstName > 	<ul style="list-style-type: none"> db2icrt < InstName >
인스턴스 Owner	<ul style="list-style-type: none"> 인스턴스 생성 전에 OS Platform에 인스턴스 Owner가 될 사용자가 반드시 존재해야 합니다. 사용자 이름은 생성하고자 하는 인스턴스 이름과 동일해야 합니다. 인스턴스 생성시 인스턴스 Owner가 OS 에서 생성되지 않았을 경우 인스턴스 생성이 실패합니다. 	<ul style="list-style-type: none"> 인스턴스 Owner가 될 OS 사용자를 지정해야 합니다. 인스턴스 Owner 이름과 인스턴스 이름이 동일할 필요는 없습니다. 인스턴스를 생성한 사용자가 인스턴스의 소유자가 됩니다.
Fenced User ID 지정 여부	<ul style="list-style-type: none"> Fenced User ID를 반드시 지정해야 합니다. Fenced User ID에 대해서는 부연 설명을 참고하시기 바랍니다. 	<ul style="list-style-type: none"> Fenced User ID를 반드시 지정할 필요는 없습니다.
생성 절차	<ol style="list-style-type: none"> root 로 로그인하거나 또는 root 권한으로 변경합니다. <ul style="list-style-type: none"> /home> su - root 인스턴스 Owner 가 될 사용자 및 그 사용자가 속할 그룹 및 Fenced User 와 Group 을 선택합니다. 지정한 사용자 및 그룹이 OS 에 없을 경우 새로 작성합니다. (AIX의 경우 예를 들었습니다.) <ul style="list-style-type: none"> /> mkgroup db2iadm /> mkuser -a pgrp=db2iadm db2inst1 /> mkgroup db2fadm /> mkuser -a pgrp=db2fadm db2fenc1 DB2가 설치된 디렉토리로 이동한 후 (AIX의 디폴트 위치는 /opt/IBM/db2/V9.5 입니다.) 하위 디렉토리인 instance 로 이동합니다. db2icrt 명령어를 수행하여 인스턴스를 생성합니다. <ul style="list-style-type: none"> /> db2icrt -u db2fenc1 db2inst1 	<ol style="list-style-type: none"> 로컬 Administrator 권한이 있는 사용자로 로그인합니다. DB2가 설치된 디렉토리로 이동한 후 (윈도우의 디폴트 위치는 c:\wprogram files\IBM\wsqllib 입니다.) 하위 디렉토리인 bin 으로 이동하거나 db2cmd 명령어를 이용하여 DB2 환경이 모두 적용된 CLP 창을 엽니다. db2icrt 명령어를 수행하여 인스턴스를 생성합니다. <ul style="list-style-type: none"> c:/> db2icrt -u db2fenc1 db2inst1



- Fenced ID는 사용자가 작성한 함수 또는 Stored Procedure를 별도의 메모리 및 프로세스를 두어 DB2 Engine이 운영되는 메모리 및 프로세스와 분리함으로써 버그 발생시 DB2 Engine에 영향을 미치지 않도록 합니다.
- FencedID는 분리된 함수 또는 Stored Procedure를 실행할때 사용되는 사용자라서 OS 사용자를 지정해야 합니다.
- 사용자 작성 함수 또는 Stored Procedure를 작성 및 사용하지 않을 경우에는 인스턴스 소유자와 동일해도 됩니다.

05 DB2 인스턴스 생성

Point DB2 인스턴스 생성시 윈도우와 Linux/Unix 간에 차이점이 있습니다. 이곳에서는 예제를 통해 인스턴스를 생성하도록 합니다.

윈도우

1. 로컬 Administrator 권한이 있는 사용자로 로그인합니다.
2. 명령창에서 dbcmd 명령어를 이용하여 CLP 창을 엽니다. CLP는 DB2 환경이 적용된 클라이언트로 DB2 명령어 및 SQL 문을 실행할 수 있습니다.
3. db2icrt 명령어를 이용하여 인스턴스를 생성합니다.

```

C:\Program Files\IBM\SQLLIB\BIN>db2icrt inst95
DB20000I DB2ICRT 명령이 완료되었습니다.

C:\Program Files\IBM\SQLLIB\BIN>
  
```

Unix/Linux

1. root 로 로그인하거나 또는 root 권한으로 변경합니다.
/home> su - root
2. 인스턴스 Owner가 될 사용자 및 그 사용자가 속할 그룹 및 Fenced User와 Group 을 선택합니다.
3. 지정한 사용자 및 그룹이 OS 에 없을 경우 새로 작성합니다. (AIX의 경우 예를 들었습니다.)
/> mkgroup db2iadm
/> mkuser -a pgrp=db2iadm db2inst1
/> mkgroup db2fadm
/> mkuser -a pgrp=db2fadm db2fenc1
4. DB2가 설치된 디렉토리로 이동한 후 (AIX 의 디폴트 위치는 /opt/IBM/db2/V9.5 입니다.) 하위 디렉토리인 인스턴스로 이동합니다.
5. db2icrt 명령어를 수행하여 인스턴스를 생성합니다.
/> db2icrt -u db2fenc1 db2inst1


```

dtterm
창(w) 편집(E) 옵션(O) 도움말(H)

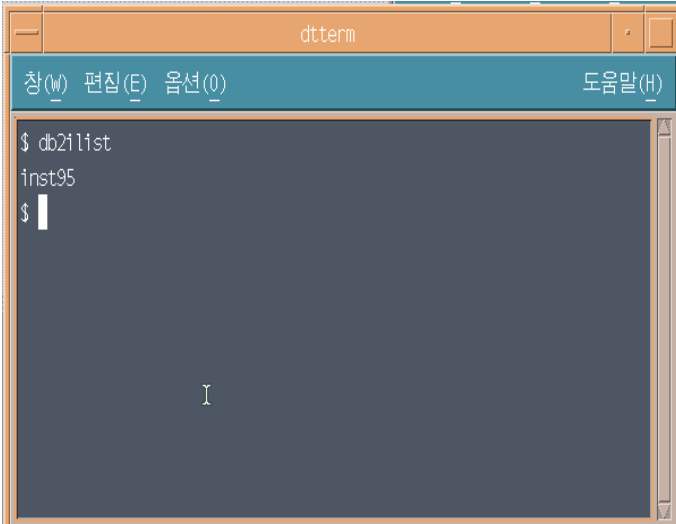
spcp5206@root:/opt/IBM/db2/V9.5/instance>./db2icrt -u fenc95 inst95
DBI1070I Program db2icrt completed successfully.

spcp5206@root:/opt/IBM/db2/V9.5/instance>
  
```

06 DB2 인스턴스 제거

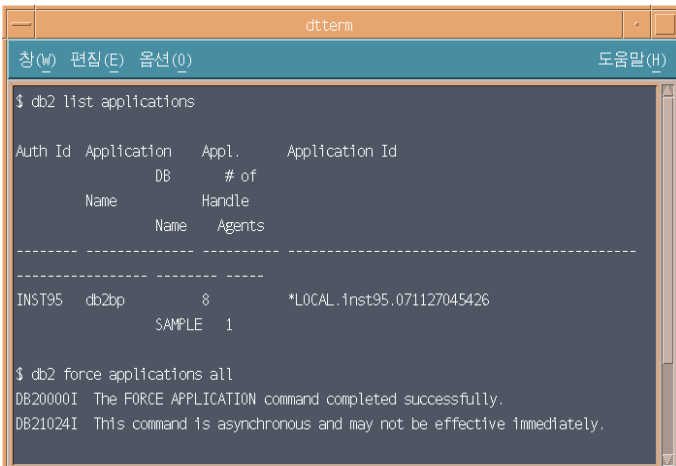
Point  인스턴스 생성과는 달리 인스턴스 제거는 윈도우와 Linux/Unix 다르지 않습니다. 아래와 같이 동일한 절차에 따라 인스턴스를 제거합니다. 아래는 Unix/Linux 환경에서 인스턴스를 제거하는 예제입니다.

1. db2ilist 명령어를 이용하여 시스템내에 존재하는 인스턴스를 확인한 후 제거할 인스턴스를 선택합니다.



```
dtterm
창(W) 편집(E) 옵션(O) 도움말(H)
$ db2ilist
inst95
$
```

2. "list applications" 명령어를 이용하여 현재 인스턴스에 접속한 애플리케이션을 확인합니다. 접속한 사용자 및 응용프로그램 담당자에게 먼저 트랜잭션을 중지하고 접속을 끊도록 권유합니다. 강제 적으로 접속을 끊을 경우에는 수행중이었던 모든 트랜잭션은 롤백됩니다. 이 곳에서는 "force applications all" 명령어를 이용하여 접속한 모든 응용프로그램의 연결을 강제로 끊습니다. 다시한번 "list applications" 명령어를 이용하여 접속한 응용프로그램이 없는지 확인합니다.

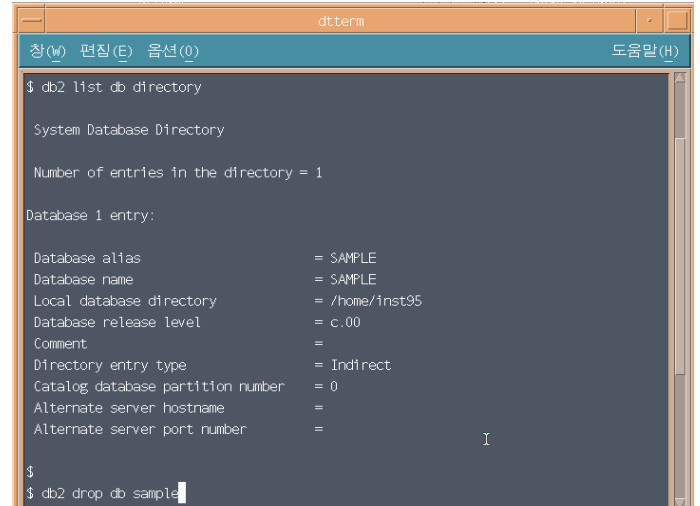


```
dtterm
창(W) 편집(E) 옵션(O) 도움말(H)
$ db2 list applications

Auth Id  Application  Appl.  Application Id
        DB      # of
        Name      Handle
        Name      Agents
-----
INST95  db2bp      8      *LOCAL_inst95_071127045426
        SAMPLE  1

$ db2 force applications all
DB20000I The FORCE APPLICATION command completed successfully.
DB21024I This command is asynchronous and may not be effective immediately.
```

3. "list db database" 명령어를 이용하여 제거하고자 하는 버전의 인스턴스 내에 존재하는 데이터베이스를 확인합니다. "drop database <DB Name>" 을 이용하여 모두 삭제합니다.



```
dtterm
창(W) 편집(E) 옵션(O) 도움말(H)
$ db2 list db directory

System Database Directory

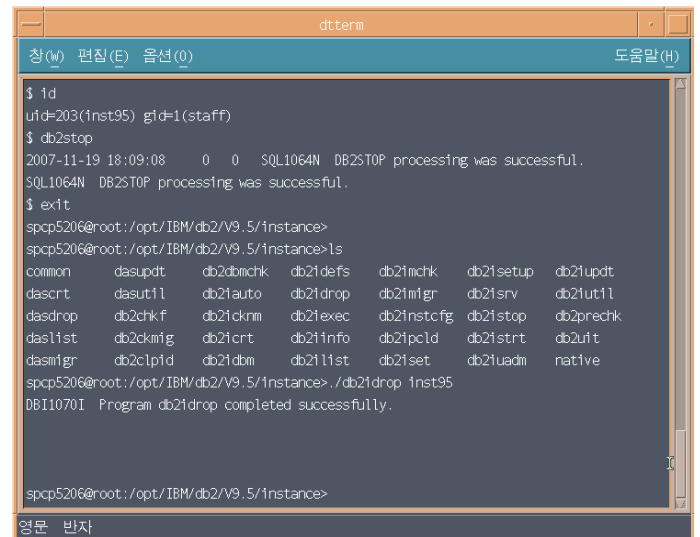
Number of entries in the directory = 1

Database 1 entry:

Database alias          = SAMPLE
Database name          = SAMPLE
Local database directory = /home/inst95
Database release level  = c.00
Comment                 =
Directory entry type    = Indirect
Catalog database partition number = 0
Alternate server hostname =
Alternate server port number =

$
$ db2 drop db sample
```

4. db2stop 명령어를 이용하여 제거하고자 하는 버전의 인스턴스를 모두 중지합니다. 혹은 접속을 끊지 않은 응용프로그램이 있을 경우 db2stop에 force 옵션을 두어 강제로 연결을 끊고 인스턴스를 중지합니다. DB2DIR/instance 디렉토리에 위치한 db2idrop 명령어를 이용하여 인스턴스를 삭제합니다.



```
dtterm
창(W) 편집(E) 옵션(O) 도움말(H)
$ !d
uid=203(inst95) gid=1(staff)
$ db2stop
2007-11-19 18:09:08 0 0 SQL1064N DB2ST0P processing was successful.
SQL1064N DB2ST0P processing was successful.
$ exit
sppc5206@root:/opt/IBM/db2/V9.5/instance>
sppc5206@root:/opt/IBM/db2/V9.5/instance>ls
common  dasupdt  db2dbmchk  db21defs  db21mchk  db21setup  db21updt
dasrct  dasutl1  db21auto  db21drop  db21migr  db21srv  db21util
dasdrop  db2chkf  db21cknm  db21exec  db21nstcfg  db21stop  db2prechk
daslist  db2ckmfg  db21crt  db21info  db21pcld  db21strt  db2uit
dasmigr  db2clpid  db21dbm  db21list  db21set  db21uadm  native
sppc5206@root:/opt/IBM/db2/V9.5/instance>./db2idrop inst95
DBI1070I Program db2idrop completed successfully.

sppc5206@root:/opt/IBM/db2/V9.5/instance>
```



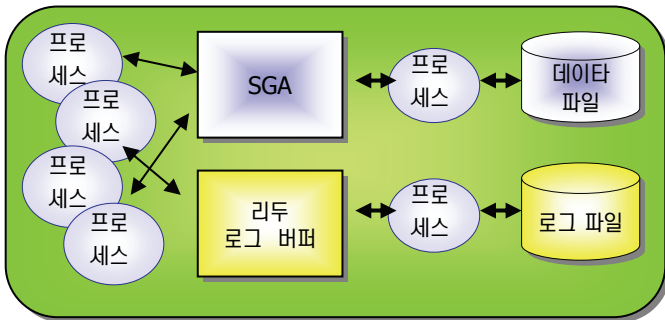
- 제거할 인스턴스내에 있는 데이터베이스를 시스템내의 다른 인스턴스에서 사용할 수 있습니다. 예를 들어 시스템 내에 test95 인스턴스와 prod95 인스턴스가 존재하고 test95 내에 AMJ라는 데이터베이스가 있다고 가정할 경우 더 이상 test95 인스턴스가 필요치 않지만 AMJ라는 데이터베이스는 계속 사용하고자 한다면 다음과 같은 절차를 따라 AMJ 데이터베이스를 재사용할 수 있습니다.

1. 데이터베이스를 drop 하지 않고 인스턴스를 중지한 후 db2idrop 명령어를 이용하여 인스턴스를 제거합니다.
2. catalog database 명령어를 이용하여 Prod95 인스턴스에서 AMJ 데이터베이스를 카탈로그 합니다.
/home/prod95>db2 catalog db amj on /home

Point 앞장에서 설명한 것처럼 오라클 데이터베이스가 물리적인 파일들로 구성된 데 반해 DB2의 데이터베이스는 프로세스 및 메모리, 물리적인 파일들로 구성되어 있습니다. 여기서는 데이터베이스의 구성 및 생성에 대해 비교합니다.

오라클

Oracle SID = ORA11g



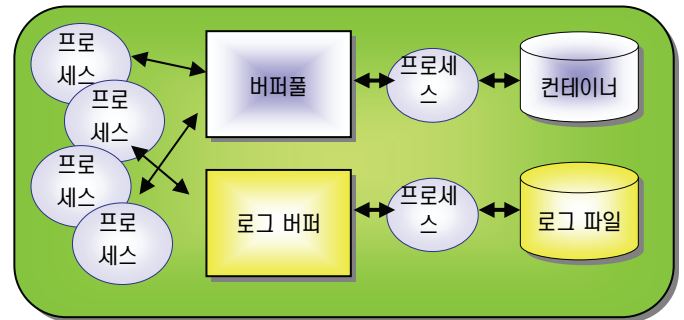
```
sqlplus > CREATE DATABASE ORA11G
LOGFILE
GROUP 1 ( '~/ORADATA/log/redo01.log')
SIZE 100M,
GROUP 2( '~/ORADATA/log/redo02.log')
SIZE 100M
MAXLOGFILES 5
MAXLOGMEMBERS 5
DATAFILE '$HOME/ORADATA/data/;
```

```
sqlplus > conn / as sysdba;
```

- 오라클 데이터베이스는 물리적인 파일들로 구성됩니다.
- SYSDBA 권한이 필요합니다.
- 데이터베이스 생성 방법 :
 - 오라클 설치시 데이터베이스 생성이 가능합니다.
 - Oracle Database Configuration Assistant에 의해 생성이 가능합니다.
 - "CREATE DATABASE ~" 명령어에 의해 생성이 가능합니다.

DB2

DB Name = SAMPLE




```
db2=> CREATE DATABASE SAMPLE
AUTOMATIC STORAGE YES ON C:,D
```

```
db2=> UPDATE DB CFG FOR SAMPLE USING
LOGPRIMARY 10
LOGSECOND 5
LOGFILSIZ 2000 NEWLOGPATH C:\wdb2_logs
LOGARCHMETH1 DISK:c:\warch_logs
```

```
db2=> CONNECT TO SAMPLE
```


- DB2 데이터베이스는 메모리와 프로세스, 물리적인 파일들로 구성됩니다.
- 데이터베이스를 생성하기 위해서는 SYSADM, 또는 SYSCTRL 권한이 필요합니다.
- 데이터베이스 생성 방법 :
 - DB2 설치시 데이터베이스 생성이 가능합니다.
 - GUI 툴인 제어센터에서 데이터베이스 생성이 가능합니다.
 - "CREATE DATABASE~" 명령어에 의해 생성이 가능합니다.

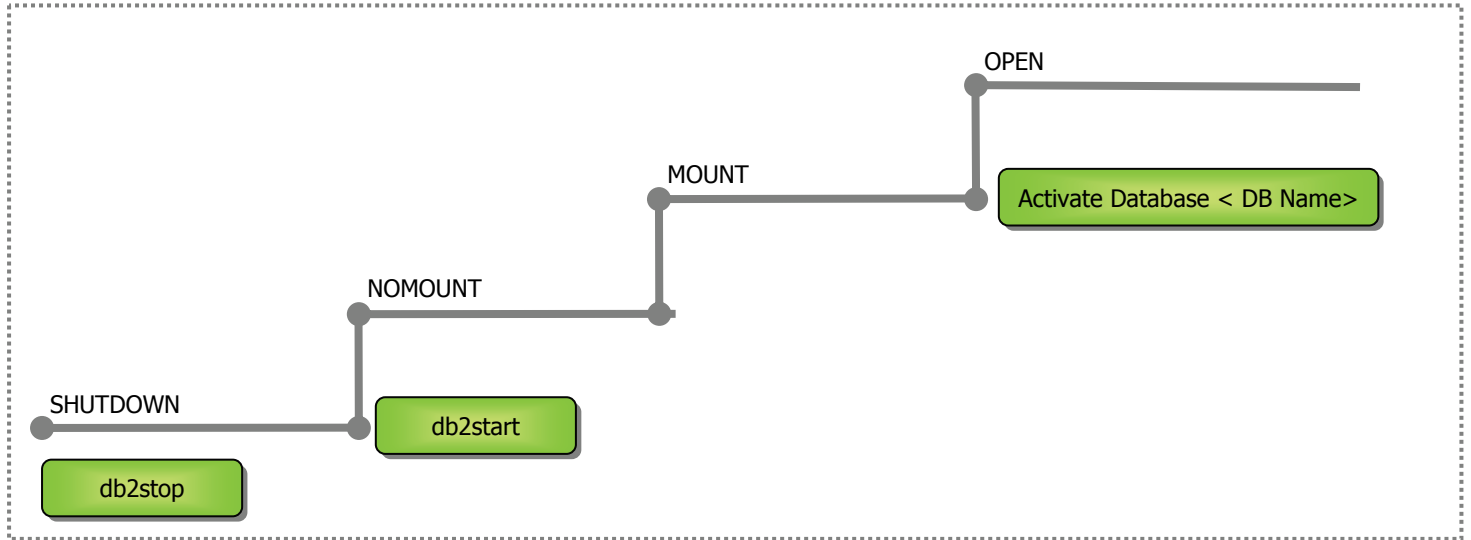
08 데이터베이스 주요 파일 비교

Point  오라클 데이터베이스를 구성하는 파일군과 DB2 데이터베이스를 구성하는 파일군은 유사하나 다음과 같이 약간의 차이점이 있습니다. 유의하여 파일을 비교하시기 바랍니다.

오라클	DB2
<ul style="list-style-type: none"> ▪ 파라미터 파일 (init.ora 파일) <ul style="list-style-type: none"> ▪ 인스턴스 및 데이터베이스를 구성하는 파라미터 값이 포함됩니다. 	<ul style="list-style-type: none"> ▪ DB Configuration file <ul style="list-style-type: none"> ▪ 데이터베이스를 구성하는 파라미터값이 포함됩니다.
<ul style="list-style-type: none"> ▪ 패스워드 파일 	<ul style="list-style-type: none"> ▪ 존재하지 않습니다. <ul style="list-style-type: none"> ▪ DB2는 OS의 사용자를 사용하여 권한을 부여합니다.
<ul style="list-style-type: none"> ▪ archived redo log file <ul style="list-style-type: none"> ▪ ARCHIVEDLOG Mode 일 경우 redo log file이 overwrite 되기 전에 redo log file을 복사하여 보존합니다. ▪ 데이터베이스에 문제 발생시 복구를 위해 사용됩니다. 	<ul style="list-style-type: none"> ▪ archived log file <ul style="list-style-type: none"> ▪ Archiving 로깅 으로 구성될 경우 redo log file을 overwrite 하지 않고 새로운 로그 파일을 생성합니다. ▪ 데이터베이스에 문제 발생시 복구를 위해 사용됩니다.
<ul style="list-style-type: none"> ▪ Control File : <ul style="list-style-type: none"> ▪ 바이너리 파일로 물리적인 데이터베이스의 현재 상태를 정의합니다. ▪ 데이터베이스의 integrity를 유지합니다. ▪ CREATE DATABASE ~ 명령어 수행시 control file의 위치와 사이즈를 지정할 수 있습니다. ▪ DB_CREATE_ONLINE_LOG_DEST_N 지정함으로써 OMF 를 이용하여 디폴트 control file 위치를 지정할 수 있습니다. 	<ul style="list-style-type: none"> ▪ 다음과 같이 여러 개의 제어 파일이 존재합니다. <ul style="list-style-type: none"> ▪ SQLBP.1 및 SQLBP.2 : 버퍼풀 정보를 포함하고 있으며 백업 목적으로 두개의 파일이 존재합니다. ▪ SQLSPCS.1 및 SQLSPCS.2 : 테이블 스페이스 정보가 존재합니다. ▪ SQLOGCTL.LFH.1 및 SQLOGCTL.LFH.2 : 로그를 제어하는 파일로 active log에 대한 정보가 포함되며 백업 목적으로 두개의 파일이 존재합니다. ▪ SQLDBCON 및 SQLDBCONF : database configuration 파일로 database 를 구성하는 파라미터 값이 포함되며 SQLDBCONF 가 깨질 경우 SQLDBCON이 사용됩니다. ▪ db2rhis.asc 및 db2rhist.bak : 백업 및 복구, 테이블 로드, reorg 등에 대한 history 정보가 포함된 파일입니다. ▪ DB2가 자동으로 제어 파일들을 생성 및 관리합니다.
<ul style="list-style-type: none"> ▪ Data File <ul style="list-style-type: none"> ▪ Tablespace를 구성하는 물리적인 파일입니다. ▪ 데이터베이스 생성시 UNDO Tablespace, System Tablespace, Temporary Tablespace를 지정합니다. ▪ DB_CREATE_FILE_DEST를 지정함으로써 OMF를 이용하여 디폴트 data file 위치를 지정할 수 있습니다. 	<ul style="list-style-type: none"> ▪ container <ul style="list-style-type: none"> ▪ 테이블 스페이스를 구성하는 물리적인 파일입니다. ▪ 데이터베이스 생성시 Catalog Tablespace, User Space, Temporary Tablespace가 자동으로 생성되며 사용자 선택에 따라 위치 및 관리형식등 변경할 수 있습니다.
<ul style="list-style-type: none"> ▪ Redo Log file : <ul style="list-style-type: none"> ▪ 데이터의 변경내역을 기록한 파일로 Crash Recovery를 위해 사용되는 파일입니다. 	<ul style="list-style-type: none"> ▪ Log File <ul style="list-style-type: none"> ▪ 데이터의 변경내역을 기록한 파일로 Crash Recovery를 위해 사용되는 파일입니다.

09 인스턴스 및 데이터베이스 시작/중지 비교

Point  오라클과 DB2 모두 인스턴스를 제어하는 명령어가 존재합니다. 여기서는 인스턴스 및 데이터베이스를 제어하는 명령어를 비교합니다.



시작 방법 비교

오라클	DB2	부연 설명
SQL> startup nomount	db2=> db2start	인스턴스를 시작합니다. 프로세스가 시작되며 메모리가 할당됩니다.
SQL> startup nomount SQL> alter database open 또는 SQL> startup	db2=> db2start db2=> activate database < DB 명>	인스턴스를 시작하며 사용자들이 데이터베이스에 접속할 수 있도록 허용합니다.
SQL> startup restrict 또는 sqlplu> alter system enable RESTRICTED SESSION	db2=> db2start admin mode	제한된 사용자만이 데이터베이스에 접속할 수 있습니다. 오라클의 경우 RESTRICTED SESSION 권한이 있는 사용자만이 접속이 가능하며 DB2의 경우 SYSADM, SYSMANT, SYSCTRL 권한을 가진 사용자만이 접속이 가능합니다.

중지 방법 비교

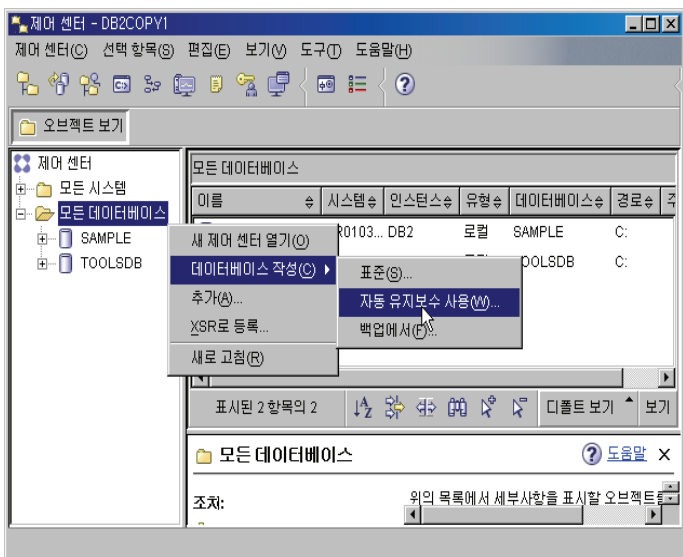
오라클	DB2	부연 설명
SQL> shutdown 또는 SQL> shutdown normal	db2> db2stop	인스턴스를 중지합니다. 데이터베이스에 접속한 사용자가 있을 경우 중지되지 않습니다.
SQL> shutdown immediate	db2> db2stop force	데이터베이스에 접속한 application의 연결을 모두 강제 종료하고 인스턴스를 중지합니다.

10 제어센타를 이용한 DB2 데이터베이스 생성

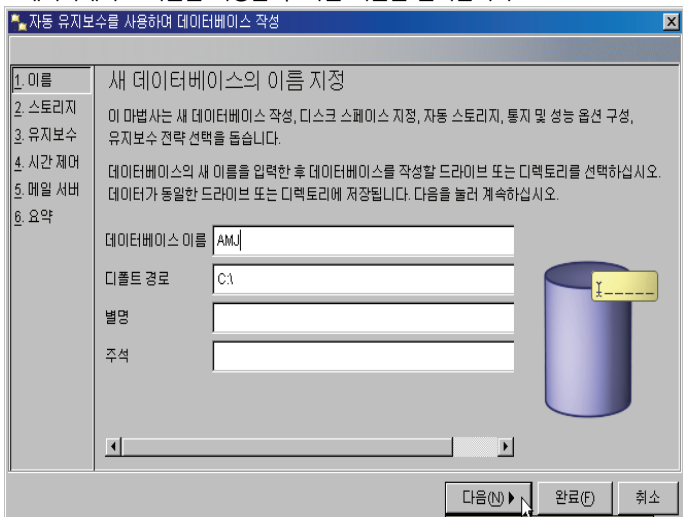
인스턴스 및 데이터베이스 비교

Point 제어센타를 이용하여 DB2 데이터베이스를 생성합니다.

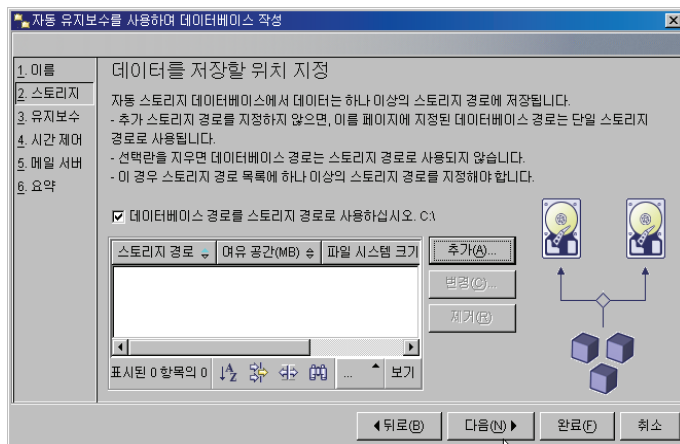
1. 제어센타에서 모든 데이터베이스를 선택한 후 오른쪽 마우스를 클릭합니다. 서브 메뉴가 나타나면 데이터베이스 작성을 선택합니다. 아래 그림과 같이 "표준", "자동 유지 보수 사용", "백업에서" 3가지 옵션을 선택할 수 있습니다. 표준은 사용자가 수동으로 테이블 스페이스에 대한 스토리지를 관리하는 모드이며 "자동 유지보수 사용"은 DBMS가 자동으로 테이블 스페이스에 대한 스토리지를 관리하는 모드입니다. 마지막으로 백업은 백업이미지로부터 데이터베이스를 생성하는 모드입니다. 이곳에서는 자동 유지보수 사용으로 데이터베이스를 생성합니다.



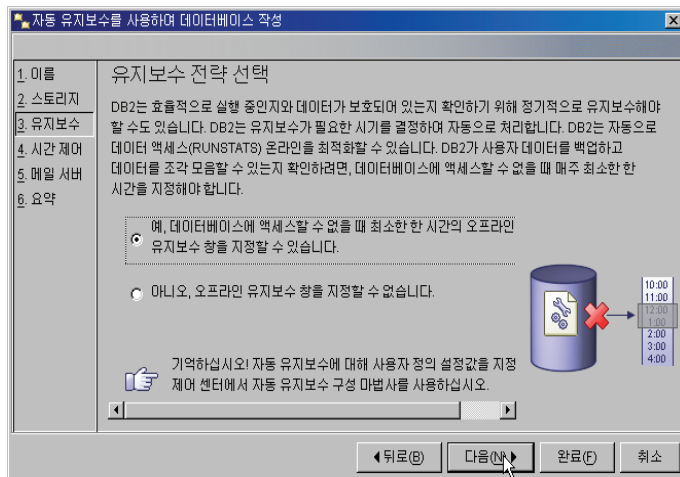
2. 데이터베이스 이름을 지정한 후 다음 버튼을 클릭합니다.



3. 데이터를 저장할 위치를 지정합니다. 추가 스토리지를 지정하지 않을 경우 단일 스토리지 경로로 사용됩니다. 다음 버튼을 클릭합니다.



4. DB2의 성능 및 안정성을 위해 정기적으로 데이터 백업 및 조각 모음이 필요하며 데이터를 조각 모음하려면 일정 시간 데이터베이스를 오프라인으로 할 필요가 있습니다. 그러나 만약 계속 데이터베이스를 사용해야 한다면 오프라인 유지보수 창을 지정할 수 없다고 지정하면 됩니다.

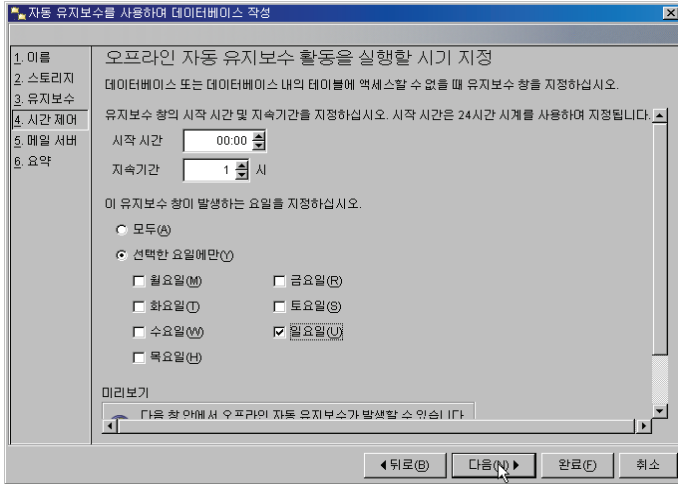


11 제어센타를 이용한 DB2 데이터베이스 생성

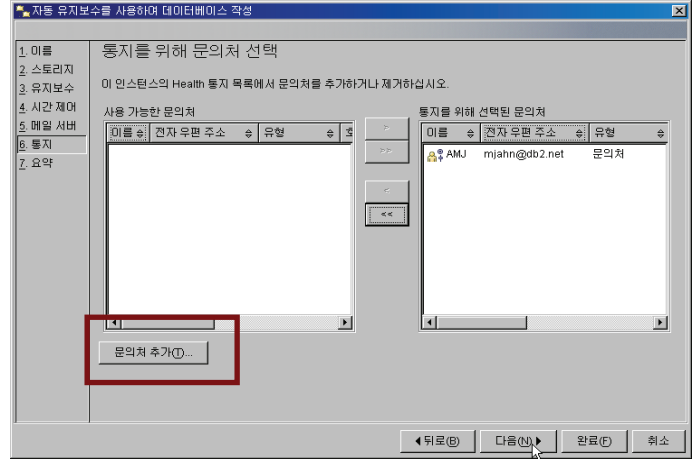
인스턴스 및 데이터베이스 비교

Point 제어센타를 이용하여 DB2 데이터베이스를 생성합니다.

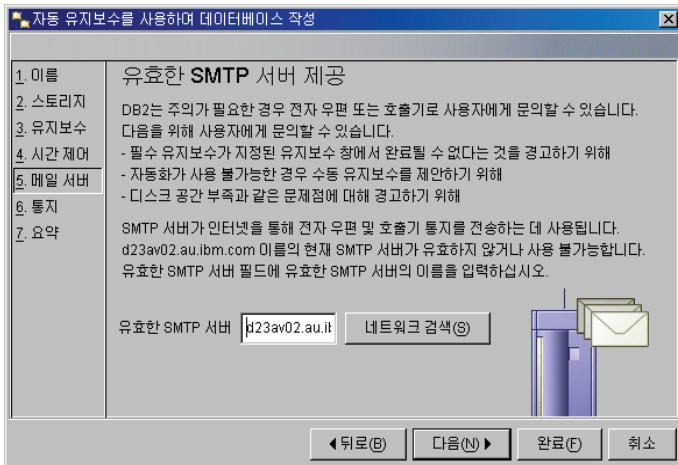
5. 오프라인 유지보수 지정시 아래와 같이 오프라인 유지보수를 수행할 시간 및 요일을 지정할 수 있습니다.



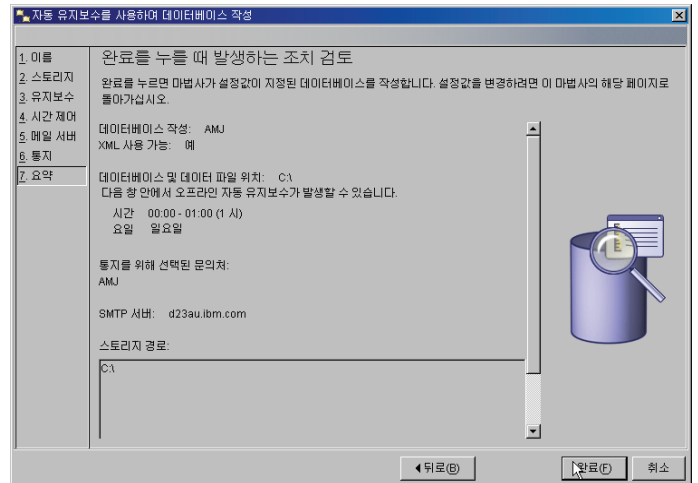
7. 데이터베이스 설치시 또는 Health Center에서 데이터베이스에 문제 발생 할 경우 아래 등록된 문의처로 메시지를 보냅니다. 이러한 문의처 정보를 등록하시기 바랍니다.



6. 아래 그림과 같이 생성하는 데이터베이스에 문제 발생시 SMTP 서버를 이용하여 사전에 등록된 문의처에 메시지를 전송하도록 설정합니다.



8. 지정되었던 내용을 확인한 후 완료 버튼을 클릭하면 데이터베이스가 생성됩니다.



12 명령어를 이용한 데이터베이스 생성

Point 명령어를 이용하여 DB2 데이터베이스를 생성합니다. 예제는 AIX 플랫폼에서 데이터베이스를 생성하는 예제입니다.

Create DB 구분

● CREATE DATABASE — < DB Name> — CREATE Database 옵션 —>

CREATE Database 옵션 :

● AUTOMATIC STORAGE — [YES | NO] — ON — [path /drive] —>

● CATALOG TABLESPACE — 테이블 스페이스 옵션 —>

● TEMPORARY TABLESPACE — 테이블 스페이스 옵션 —>

● USER TABLESPACE — 테이블 스페이스 옵션 —>

● AUTOCONFIGURE — USING — Input 키워드 파라미터 — Value — APPLY — DB ONLY — DB AND DBM — NONE —>

참고 : 이외에도 다양한 옵션이 지정되어 있으므로 도움말을 참고하시기 바랍니다.

Input 키워드 파라미터

파라미터	범위	디폴트 값	설명
workload_type	simple, mixed, complex	mixed	Simple 워크로드는 I/O 작업이 많고 대부분 트랜잭션 위주이며 complex 워크로드는 CPU 작업이 많고 쿼리 위주입니다. 업무를 파악하여 적당한 워크로드를 선택합니다.
tpm	1-200 000	60	분당 트랜잭션을 지정합니다.
num_remote_apps	0-5 000	100	접속되는 원격 응용프로그램의 수를 지정합니다.

참고 : 이외에도 다양한 옵션이 지정되어 있으므로 도움말을 참고하시기 바랍니다.


예제

```
/home/inst95 > db2
db2 => create db amj automatic storage yes on /db2_data1,/db2_data2
DB20000I The CREATE DATABASE command completed successfully.
Db2=> list db directory → 내가 접속할 수 있는 데이터베이스 목록을 보여줍니다.
```



- 오라클의 HR Schema가 가지고 있는 Sdb2sampl 명령어를 이용하여 sample 데이터베이스를 생성할 수 있습니다. SAMPLE 데이터베이스에는 EMPLOYEE, DEPARTMENT 테이블 등이 존재하며 SAMPLE DATA가 삽입되어 있으므로 SAMPLE 애플리케이션 등을 테스트 할 수 있습니다.

13 환경 변수 및 파라미터 비교


Point  오라클과 DB2는 OS 환경변수와 제품내에서 제공하는 구성 파라미터를 이용하여 DBMS 환경을 구성합니다. 이곳에서는 오라클과 DB2를 구성하는 환경 변수 및 구성 파라미터 파일을 비교합니다.

환경 변수 비교


	오라클	DB2
환경 변수	<ul style="list-style-type: none"> OS 환경변수를 이용하여 데이터베이스 환경을 제어합니다. ORACLE_SID, ORACLE_BASE 을 OS 환경변수에 설정합니다. 	<ul style="list-style-type: none"> OS 환경변수 및 DB2 Registry 변수를 이용하여 데이터베이스 환경을 제어합니다. db2set 이라는 명령어를 통해 DB2 Registry 변수를 설정합니다. OS 환경변수로는 DB2PATH, DB2INSTANCE 가 있습니다.

구성 파라미터 비교

	오라클	DB2
인스턴스 및 데이터베이스 구성 파라미터	<ul style="list-style-type: none"> PFFILE : <ul style="list-style-type: none"> init<SID>.ora 라는 텍스트 파일에 database initialization parameters 값을 저장합니다. database initialization parameters 값이 설정되어 있으며 OS Editor 에서 편집이 가능한 텍스트 파일입니다. 인스턴스가 기동될때 pfile을 읽어서 데이터베이스에 적용합니다. SPFILE : <ul style="list-style-type: none"> 위부터 도입된 방식으로 database initialization parameters를 spfile<SID>.ora 라는 바이너리 파일에 저장합니다. 데이터베이스에서 관리하는 방식입니다. 인스턴스가 기동될 때 pfile이 아닌 spfile을 참고하여 데이터베이스에 적용됩니다. 파라미터 설정시 pfile이나 spfile 중 어떤 방식으로든 적용 가능합니다. 	<ul style="list-style-type: none"> Database Manager Configuration 또는 Database Configuration 이라는 바이너리 파일에 인스턴스 및 데이터베이스 구성 파라미터 값을 저장하며 Database Configuration 파일에 데이터베이스 구성 파라미터 값을 저장합니다. 오라클의 SPFILE 에 해당합니다.
구성 파라미터 확인 방법	SQL> show parameter [Parameter Name]	db2=> get dbm cfg db2=> get db cfg for < Database Name>
구성 파라미터 변경 방법	SQL> alter system set < Parameter Name> = < Value > [SCOPE=[MEMORY SPFILE BOTH]]	db2=> update dbm cfg using < Parameter Name> < Value > db2=> update db cf g for < Database Name> using <Parameter Name> < Value >
구성 파라미터 Default 설정 방법	<ul style="list-style-type: none"> Default 값으로 변경합니다 SQL> alter system reset < Parameter Name> [SCOPE= MEMORY SPFILE BOTH]	<ul style="list-style-type: none"> Default 값으로 변경합니다 db2=> reset dbm cfg db2=> reset db cfg for < Database Name>
구성 파라미터 적용 방법	SQL> startup /* 이 경우 SPFILE 을 적용 */ 또는 SQL>startup PFILE=initora10g.ora	/home/inst32/db2start

Point  인스턴스와 데이터베이스를 구성하는 파라미터 값은 오라클의 경우 init<SID>.ora 파일에 저장되며 DB2 에서는 Database Manager Configuration 파일과 Database Configuration File 에 저장됩니다.


Oracle 10g init.ora 파라미터	DB2 9.5 구성 파라미터	설명
control_files	일치하는 파라미터 없음	<ul style="list-style-type: none"> 오라클: <ul style="list-style-type: none"> 하나이상의 control file을 지정하는 옵션입니다. DB2: <ul style="list-style-type: none"> SQLBP.1, SQLSPCS.1, SQLOG*, SQLDBCON, db2rhist.asc 등의 파일이 control 파일과 유사한 일을 수행합니다.
db_nk_cache_size	일치하는 파라미터 없음	<ul style="list-style-type: none"> 오라클: <ul style="list-style-type: none"> nK 버퍼에 대한 캐시 크기를 지정하는 파라미터입니다. DB2: <ul style="list-style-type: none"> page size 별로 사용자가 원하는 pagesize의 버퍼풀 생성이 가능합니다. <p>* 자세한 사항은 테이블 스페이스에서 다루도록 합니다.</p>
db_block_checking	일치하는 파라미터 없음	<ul style="list-style-type: none"> 오라클: <ul style="list-style-type: none"> 트랜잭션 관리 블록의 손상여부를 확인할지에 대해 지정하는 파라미터입니다. DB2: <ul style="list-style-type: none"> Consistency 비트를 이용하여 page level 의 integrity를 유지시킵니다. 이것은 오라클의 checksum page checking 보다 성능이 우수합니다.
db_block_checksum	일치하는 파라미터 없음	<ul style="list-style-type: none"> 오라클: <ul style="list-style-type: none"> 읽거나 기록한 모든 데이터 블록에 대해 DBWn, ARCH, SQL*Loader가 block checksum을 계산 또는 확인할지 지정하는 파라미터입니다. DB2: <ul style="list-style-type: none"> Consistency 비트를 이용하여 page level의 integrity를 유지시킵니다. 이것은 오라클의 checksum page checking 보다 성능이 우수합니다.
db_create_online_log_dest_n	newlogpathn (db cfg 파라미터)	<ul style="list-style-type: none"> 오라클: <ul style="list-style-type: none"> control file 및 log file의 Default 위치를 지정하는 파라미터입니다. DB2: <ul style="list-style-type: none"> 로그 파일의 위치를 변경할때 변경하고자 하는 위치를 지정합니다.
db_block_size	pagesize (db cfg 파라미터)	<ul style="list-style-type: none"> 오라클: <ul style="list-style-type: none"> 데이터베이스 생성시 설정되며 이후에는 변경할 수 없습니다. DB2: <ul style="list-style-type: none"> 데이터베이스가 생성시 또는 테이블 스페이스 생성시 pagesize를 지정할 수 있으며 PageSize를 지정하지 않을 경우 데이터베이스 생성시 지정하였던 값을 Default로 사용합니다.
db_files	maxtotfilop (dbm cfg 파라미터)	<ul style="list-style-type: none"> DB2 및 오라클: <ul style="list-style-type: none"> 열 수 있는 데이터베이스의 최대 개수입니다.
db_keep_cache_size	일치하는 파라미터 없음	<ul style="list-style-type: none"> 오라클: <ul style="list-style-type: none"> 재활용될 가능성이 높은 Block을 고정적으로 저장하는데 사용하는 파라미터로 KEEP 버퍼 캐시 크기를 지정합니다. DB2: <ul style="list-style-type: none"> 메모리에 데이터 페이지를 보존하기 위해서 메모리에 보존하고자 하는 테이블에 대한 버퍼풀을 따로 생성할 수 있습니다.
db_recycle_cache_size	일치하는 파라미터 없음	<ul style="list-style-type: none"> 오라클: <ul style="list-style-type: none"> 재활용될 가능성이 낮은 Block을 즉시 제거하는데 사용되는 파라미터로 Recycle 버퍼 캐시 크기를 지정합니다. DB2: <ul style="list-style-type: none"> DB2의 버퍼풀은 단순히 LRU 알고리즘을 사용하는 것이 아니라 이보다 좀 더 향상된 알고리즘을 사용하므로 Recycle Pool이 필요치 않습니다.

Point  인스턴스와 데이터베이스를 구성하는 파라미터 값은 오라클의 경우 init<SID>.ora 파일에 저장되며 DB2에서는 Database Manager Configuration 파일과 Database Configuration File에 저장됩니다.

Oracle 10g init.ora 파라미터	DB2 9.5 구성 파라미터	설명
db_writer_processes	num_iocleaners (db cfg 파라미터)	<ul style="list-style-type: none"> 오라클 : <ul style="list-style-type: none"> 버퍼 캐쉬에서 Dirty Page를 데이터 파일에 기록하는 프로세스의 수를 지정하는 파라미터입니다. DB2 : <ul style="list-style-type: none"> 버퍼풀에서 Dirty Page를 Container에 기록하는 프로세스의 수를 지정하는 파라미터입니다.
dml_locks	매핑되는 파라미터 없음 (locklist 와 maxlock 참조)	<ul style="list-style-type: none"> 오라클 : <ul style="list-style-type: none"> 모든 사용자에게 의해 획득된 테이블 잠금의 최대 개수를 지정하는 파라미터입니다. DB2 : <ul style="list-style-type: none"> DB2 lock은 메모리에 저장되며 lock의 크기 정보는 Database Configuration 파일에 지정됩니다.
java_pool_size	java_heap_sz (dbm cfg 파라미터)	<ul style="list-style-type: none"> DB2 및 오라클 : <ul style="list-style-type: none"> java app 용 메모리 크기를 지정하는 파라미터입니다.
local_listener	svcname (dbm cfg 파라미터) db2comm (registry 변수)	<ul style="list-style-type: none"> 오라클 : <ul style="list-style-type: none"> Listener에 의해 사용되는 포트 및 프로토콜을 정의합니다. DB2 : <ul style="list-style-type: none"> Listener에 의해 사용되는 포트 및 프로토콜을 정의합니다. SVCENAME 파라미터는 포트를 정의하며 Registry 변수인 DB2COMM은 프로토콜을 정의합니다. <p>* 자세한 사항은 제 5장 네트워크 설정에서 다루도록 합니다.</p>
log_archive_dest	logarchmeth1	<ul style="list-style-type: none"> 오라클 : <ul style="list-style-type: none"> 데이터베이스를 아카이브 모드로 변경하는 파라미터중 하나로 아카이브 파일의 위치를 지정합니다. DB2 : <ul style="list-style-type: none"> 데이터베이스를 아카이브 모드로 변경하는 파라미터중 하나로 파라미터 값을 디스크로 설정시 그 아카이브 파일의 위치를 지정할 수 있으며 또한 옵션을 두어 타 벤더 또는 TSM으로 로그 파일을 아카이브할 수 있습니다.
log_buffer	logbufsz (db cfg 파라미터)	<ul style="list-style-type: none"> DB2 및 오라클 : <ul style="list-style-type: none"> 로그 버퍼의 크기를 지정합니다.
sessions	maxcagents (dbm cfg 파라미터)	<ul style="list-style-type: none"> DB2 및 오라클 : <ul style="list-style-type: none"> 동시에 접속한 애플리케이션의 수를 지정합니다.
sort_area_size	sortheap	<ul style="list-style-type: none"> 오라클 : <ul style="list-style-type: none"> PGA 내의 정렬을 위해 사용될 메모리 크기를 지정합니다. DB2 : <ul style="list-style-type: none"> 개인 정렬시 사용될 private memory의 최대 크기 또는 공유 정렬시 사용될 shared memory의 최대 크기를 지정합니다.

16 DB2 환경 변수 및 주요 레지스트리 변수

인스턴스 및 데이터베이스 비교

Point  이곳에서는 DB2 의 주요 환경 변수 및 레지스트리 변수를 알아보도록 합니다.

환경 변수 및 Registry 변수

DB2		
시스템 환경	DB2INSTANCE	<ul style="list-style-type: none"> Default로 사용할 인스턴스를 지정합니다.
	DB2_CLP_EDITOR	<ul style="list-style-type: none"> Client인 CLP (오라클의 sqlplus 에 해당) 에서 "EDIT" 명령어 수행시 사용될 에디터 프로그램을 지정합니다. 디폴트로 윈도우의 경우 notepad , Unix의 경우 vi 에디터가 지정됩니다.
	DB2_CLP_HISTSIZ E	<ul style="list-style-type: none"> CLP시 실행된 명령어를 몇 개까지 저장할 것인지를 지정합니다.
Registry 변수	DB2COMM	<ul style="list-style-type: none"> 데이터베이스 서버에서만 지정되는 변수입니다. 이 변수는 프로토콜을 지정함으로써 인스턴스 구동시 구동될 통신 프로세스를 지정합니다. 즉 TCPIP 지정시 TCPIP 를 담당하는 통신 프로세스가 구동됩니다. 이 변수를 지정하지 않을 경우 어떠한 통신 프로세스도 구동되지 않으므로 Client의 요청을 처리할 수 없습니다.

Registry 변수 제어 명령어 및 예제

명령어	설명
db2set -lr	<ul style="list-style-type: none"> 지원되는 Registry 변수를 나열합니다. 예) C:W> db2set -lr
db2set -all	<ul style="list-style-type: none"> 설정된 Registry 변수값을 모두 나열합니다. 예) C:W>db2set -all <pre>[e] DB2PATH=C:WProgram FilesWIBMWSQLLIB [i] DB2ACCOUNTNAME=KR010389Wdb2admin [i] DB2INSTOWNER=KR010389 [i] DB2PORTRANGE=60000:60003 [i] DB2COMM=TCPIP [g] DB2_EXTSECURITY=YES [g] DB2SYSTEM=KR010389 [g] DB2PATH=C:WProgram FilesWIBMWSQLLIB [g] DB2INSTDEF=DB2 [g] DB2ADMINSERVER=DB2DAS00</pre>
db2set <Registry 변수 > = <Value >	<ul style="list-style-type: none"> Registry 변수값을 지정합니다. 예) C:W> db2set DB2COMM=TCPIP

17 제어센타를 이용한 파라미터 구성

Point 제어센타를 이용하여 Database Manager Configuration File 및 Database Configuration File 에 저장된 파라미터 (매개변수)를 확인하고 수정합니다.

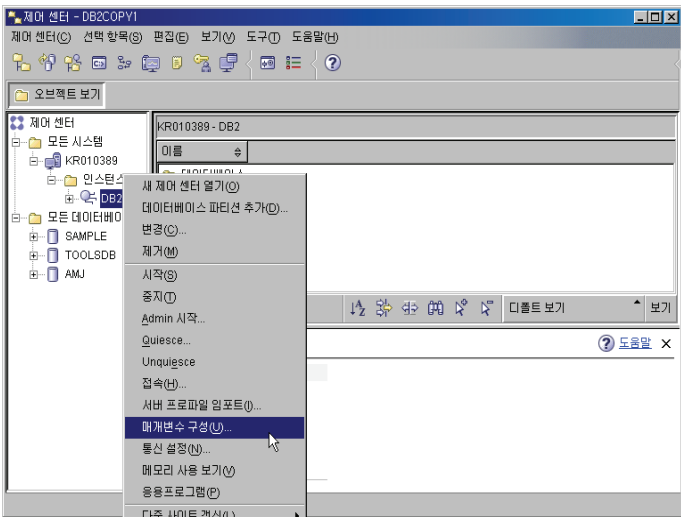
1. 시작 => 실행 에서 실행창에 "db2cmd" 명령어를 입력하여 DB2 CLP 창을 띄운 후에 DB2 서버를 시작한 후 SAMPLE 데이터베이스를 생성합니다. 만약 앞장에서 sample 데이터베이스를 생성하였을 경우 이 단계는 생략하여도 됩니다.

```
C:\W> db2start
2007-10-30 18:01:30 0 0 SQL1063N DB2START 처리
가 완료되었습니다.
SQL1063N DB2START 처리가 완료되었습니다.

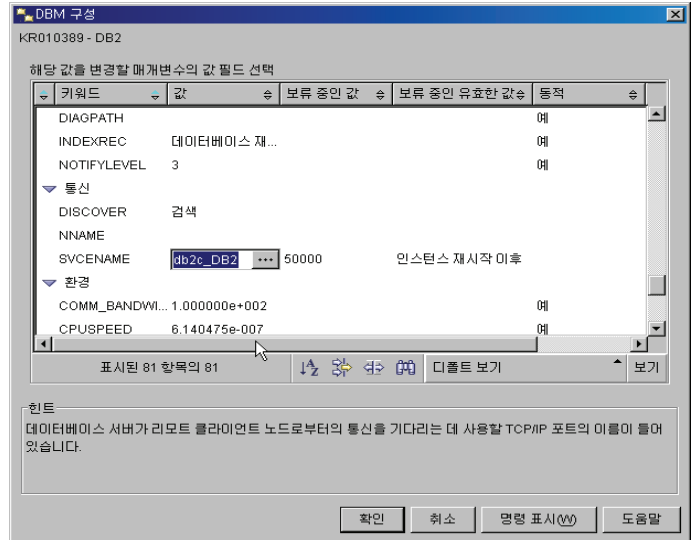
C:\W> db2sampl -sql
C:\W>db2sampl

Creating database "SAMPLE"...
Connecting to database "SAMPLE"...
Creating tables and data in schema "KR010389"...
'db2sampl' processing complete.
```

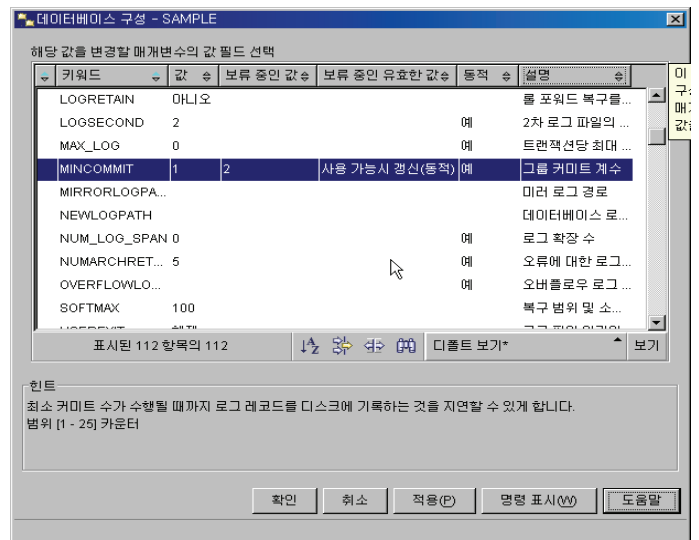
2. 시작 => 모든 프로그램 => IBM DB2 => DB2COPY1(디폴트) => 일반 관리 도구 => 제어 센터를 선택한 후 제어센터가 나타나면 "기본", "고급", "사용자 정의" 보기 중 원하는 보기를 선택합니다. 하지만 사용 가능한 모든 오브젝트를 표시하기 위해서는 "고급" 보기를 선택해야 합니다.
3. 보기를 선택한 후 모든 시스템 => < 시스템 이름 > => 인스턴스 => DB2를 선택한 후 오른쪽 버튼을 눌러 오른쪽 버튼을 눌러 아래 그림과 같이 디폴트 서버 메뉴가 나타나면 매개변수 구성을 선택합니다.



4. Database Manager Configuration에 저장된 파라미터 값을 확인하거나 변경합니다. 아래와 같이 파라미터 값 변경시 변경 값이 적용되는 시점에 대해 보여줍니다. 아래 SVCENAME 파라미터는 인스턴스 재시작후 변경된 값 50000 이 적용된다는 것을 나타냅니다. 다른 파라미터에 대해서도 확인합니다. 파라미터에 따라 변경 값이 적용되는 시점이 각기 다르므로 이 점 확인하시기 바랍니다.



6. 다시 오른쪽 패널에서 모든 데이터베이스 => SAMPLE 데이터베이스를 선택한 후 오른쪽 버튼을 눌러 서버 메뉴가 나타나면 매개변수 구성을 선택합니다.
7. Database Configuration에 저장된 파라미터 값을 확인하거나 변경합니다. 아래와 같이 매개변수 값 변경시 변경 값이 적용되는 시점에 대해 확인 가능하며 매개변수에 대한 설명도 확인할 수 있습니다. 아래 MINCOMMIT은 1에서 2로 변경되었으며 동적으로 바로 변경값이 적용된다는 것을 보여줍니다.



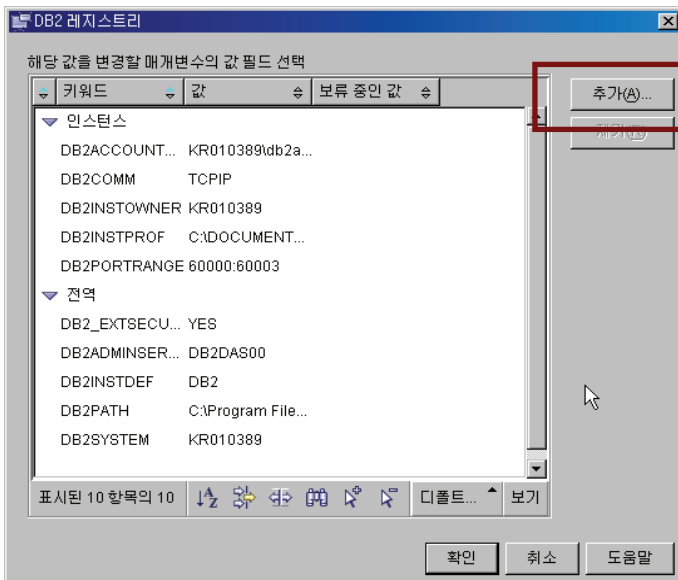
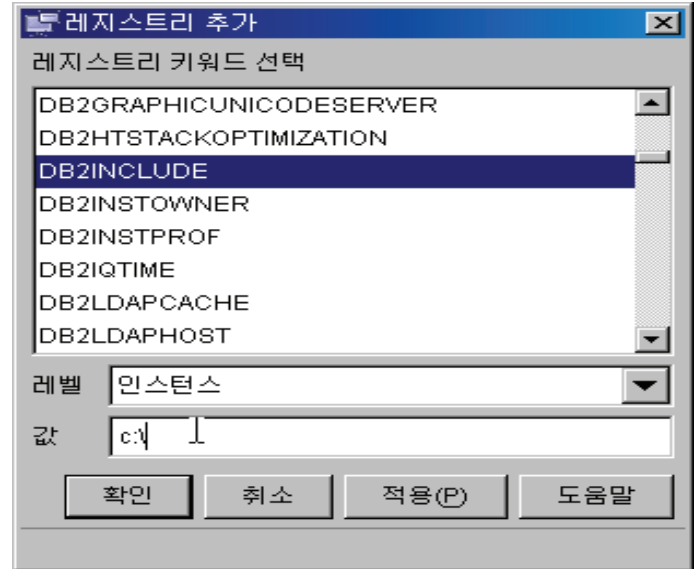
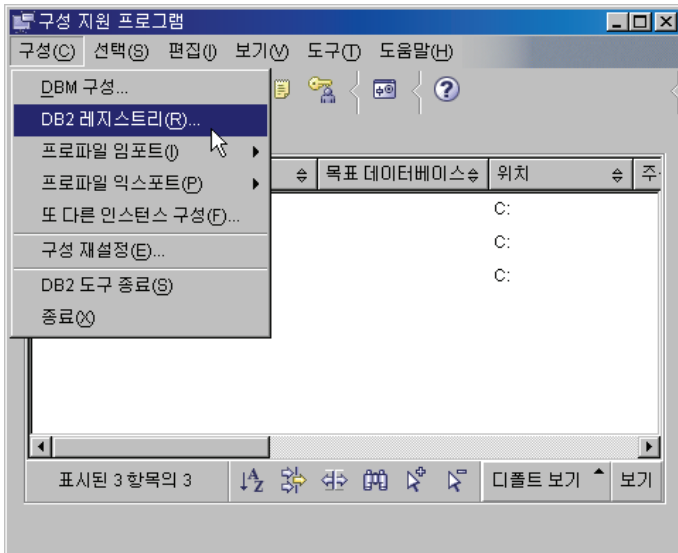
참고) DB2에서는 파라미터를 매개변수라고도 합니다.

18 구성 지원 프로그램을 이용한 레지스트리 구성

인스턴스 및 데이터베이스 비교

Point 구성지원 프로그램을 이용하여 레지스트리를 구성합니다.

1. 시작 => 모든 프로그램 => IBM DB2 => DB2COPY1(디폴트)=> 설정 도구 => 구성 지원 프로그램을 선택합니다.
2. 메뉴에서 구성을 클릭한 후 DB2 레지스트리를 선택합니다.
3. 아래 그림과 같이 Registry 값을 확인할 수 있으며 Registry를 설정할 수도 있습니다.
4. 추가 버튼을 누른 후 DB2INCLUDE라는 레지스트리 변수를 설정합니다.



19 명령어를 이용한 파라미터 구성

Point 명령어를 이용하여 Database Manager Configuration File 및 Database Configuration File 에 저장된 파라미터 (매개변수)를 확인하고 수정합니다.

1. "get dbm cfg" 명령어를 이용하여 파라미터(매개변수) 값을 확인합니다.

```
C:\WLAB> db2 get dbm cfg | more
```

데이터베이스 관리 프로그램 구성

노드 유형 = 로컬 및 리모트 클라이언트가
있는 Enterprise Server Edition

데이터베이스 관리 프로그램 구성 릴리스 레벨 = 0x0c00

전체 열린 최대 파일 수(MAXTOTFILOP) = 16000
CPU 속도(밀리초/명령어)(CPUSPEED) = 3.700030e-007
통신 대역폭(MB/초)(COMM_BANDWIDTH) =
1.000000e+002

현재 사용 중인 최대 데이터베이스 수(NUMDB) = 8
페더레이티드 데이터베이스 시스템 지원(FEDERATED) = NO
트랜잭션 프로세서 모니터 이름(TP_MON_NAME) =

2. "update dbm cfg" 명령어를 이용하여 파라미터 값을 수정합니다.

```
C:\WLAB> db2 update dbm cfg using SVCENAME 50000
```

DB20000I UPDATE DATABASE MANAGER
CONFIGURATION 명령이 완료되었습니다.

3. "get db cfg for <DB Name>" 명령어를 이용하여 파라미터 값을 확인합니다.

```
C:\WLAB> db2 get db cfg for amj | more
```

데이터베이스 amj에 대한 데이터베이스 구성

데이터베이스 구성 릴리스 레벨 = 0x0c00
데이터베이스 릴리스 레벨 = 0x0c00

데이터베이스 지역 = KR
데이터베이스 코드 페이지 = 1208
데이터베이스 코드 세트 = UTF-8
데이터베이스 국가/지역 코드 = 82
데이터베이스 조합 시퀀스 = IDENTITY
대체 조합 조합 시퀀스(ALT_COLLATE) =
Number 호환성 = OFF
Varchar2 호환성 = OFF
데이터베이스 페이지 크기 = 4096

동적 SQL 쿼리 관리(DYN_QUERY_MGMT) = DISABLE

이 데이터베이스에 대한 발견 지원(DISCOVER_DB) = ENABLE

액세스 제한 = NO
디폴트 쿼리 최적화 클래스(DFT_QUERYOPT) = 5
병렬 처리 등급(DFT_DEGREE) = 1


4. "update db cfg for <DB Name>" 명령어를 이용하여 파라미터 값을 수정합니다.

```
C:\WLAB> db2 update db cfg for amj using LOGFILSIZ 2048
```

DB20000I UPDATE DATABASE CONFIGURATION 명령이
완료되었습니다.



- get dbm cfg 명령어를 이용하여 매개변수값을 확인하면 위그림과 같이 매개변수 이름이 보이지 않는 부분이 나타납니다. 예를 들면 database manager configuration release level 은 매개변수 이름이 보이지 않고 값만 존재합니다. 이는 사용자가 수정할 수 없는 값으로 단지 사용자에게 정보용으로 보여주는 데이터입니다.
- get db cfg for <DB Name> 명령어를 이용하여 매개변수값을 확인하면 위그림과 같이 매개변수 이름이 보이지 않는 부분이 나타납니다. 예를 들면 database code page , database code set, database country 등은 매개변수 값만 보여줍니다. 이와 같은 설정값들은 데이터베이스 생성시 지정된 값들로 데이터베이스 생성 후에 사용자가 변경할 수 없습니다. 그러나 데이터베이스 생성시 지정된 모든 값을 변경할 수 없는 것이 아니라 이와같은 특정 값만 변경할 수 있습니다.

Point  인스턴스 및 데이터베이스 제어 명령문을 비교합니다.

설명	Oracle	DB2
데이터베이스 생성	Create Database ~ 시스템 테이블 스페이스 및 Temp 테이블 스페이스가 디폴트로 생성됨.	Create Database~ builds system, temp and user default tablespace
인스턴스 시작	startup pfile = ...	db2start
인스턴스 중지	shutdown [abort ...]	db2stop [force]
구성 파라미터 변경	alter system set < Parameter Name> = <Value> [SCOPE=[MEMORY SPFILE BOTH]]	update dbm cfg using < Parameter Name> < Value > update db cf g for < Database Name> using <Parameter Name> < Value >
구성 파라미터 Default 설정	<ul style="list-style-type: none"> Default 값으로 변경합니다 alter system reset < Parameter Name> [SCOPE= MEMORY SPFILE BOTH]	<ul style="list-style-type: none"> Default 값으로 변경합니다 reset dbm cfg reset db cfg for < Database Name>
구성 파라미터 적용	startup /* 이 경우 SPFILE 을 적용 */ 또는 startup PFILE=initora10g.ora	/home/inst32/db2start
데이터베이스 제거	Drop Database ~ "including contents"를 지정하지 않으면 데이터 파일은 그대로 남음. 파티션 테이블에 대한 파일은 제거되지 않음.	Drop Database ~ 테이블 및 테이블 스페이스, 컨테이너 모두 제거

인스턴스 및 데이터베이스 관련 Q/A

- 인스턴스를 ATX 에서 생성하는데 DBI1131E “User ID inst1is invalid ” 라는 에러가 발생합니다. 어떻게 해야되는건가요?

Unix/Linux Platform에서는 반드시 OS에 인스턴스 Owner가 될 사용자가 존재해야 합니다. 또한 생성할 인스턴스 이름도 인스턴스 Owner 이름과 동일합니다. OS에 사용자가 존재하는지 확인한후 존재하지 않을 경우 새 사용자를 생성한 후 인스턴스를 생성하시기 바랍니다.
- 오라클은 pfile인 init<SID>.ora 파일에 설정값을 저장하거나 또는 ‘create pfile from spfile’ 명령어를 이용하여 향후 다른 시스템에 같은 파라미터를 적용할 수 있습니다. DB2는 현재 시스템의 동일한 환경 및 매개변수 값을 적용시키고자 할때 편리한 방법이 있는지요?

오라클의 PFILE은 파라미터값만 저장할뿐 환경변수 값은 저장하지 않으며 접속할 데이터베이스 목록이나 네트워크 정보도 listener.ora 및 tnsnames.ora 파일에 별도로 관리해야 합니다. DB2에서는 GUI 툴인 구성지원 프로그램 혹은 db2cfexp 명령어를 이용하여 환경 변수, registry 및 접속할 데이터베이스 목록, database manager 매개변수 값을 모두 하나의 파일에 export 할 수 있으며 이 파일을 다른 인스턴스 또는 다른 시스템에서 import 하여 동일한 환경을 구축할 수 있습니다.
- Database Configuration File에 존재하는 매개변수를 변경하였는데 바로 적용되지 않는 듯 합니다. 적용되는 시점이 언제인지 어떻게 확인할 수 있는지요?

제어센터에서 변경할 경우 “보류중인 유효한 값” 칼럼에 적용시점에 대해서 나타냅니다. 만약 명령어를 이용하여 변경할 경우 변경 후에 메시지가 나타나 언제 적용되는지 알려줍니다. 예를 들어 클라이언트와의 통신을 위해 포트를 변경하였을 경우에는 인스턴스를 재 시작해야 변경된 포트값으로 DB 서버가 listen 합니다.
- 데이터베이스를 생성하는데 SQL1005N 오류가 발생합니다. 어떻게 해야되는지요?

인스턴스내에 동일한 이름의 데이터베이스가 존재합니다. “list db directory” 를 이용하여 접속할 수 있는 데이터베이스 목록을 확인한 후 동일한 이름이 있을 경우 다른 이름으로 생성하도록 합니다.



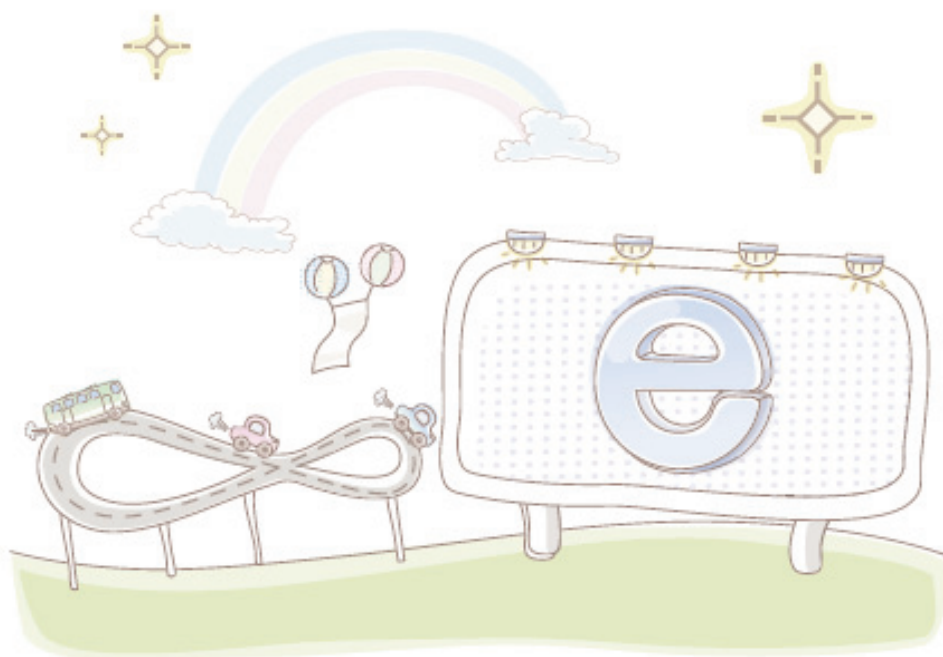
Unit 06 : 네트워크 설정 비교

오리플 사용자를 위한 DB2 가이드

서버 네트워크 설정 비교

원격 데이터베이스 설정 비교

DB2 네트워크 설정



01 서버 네트워크 설정 비교

Point 클라이언트의 접속 요청을 처리하기 위해서는 데이터베이스 서버에 네트워크를 설정해야 합니다. 이곳에서는 오라클과 DB2의 서버측 네트워크 설정방법 및 관리방법을 비교하도록 합니다.

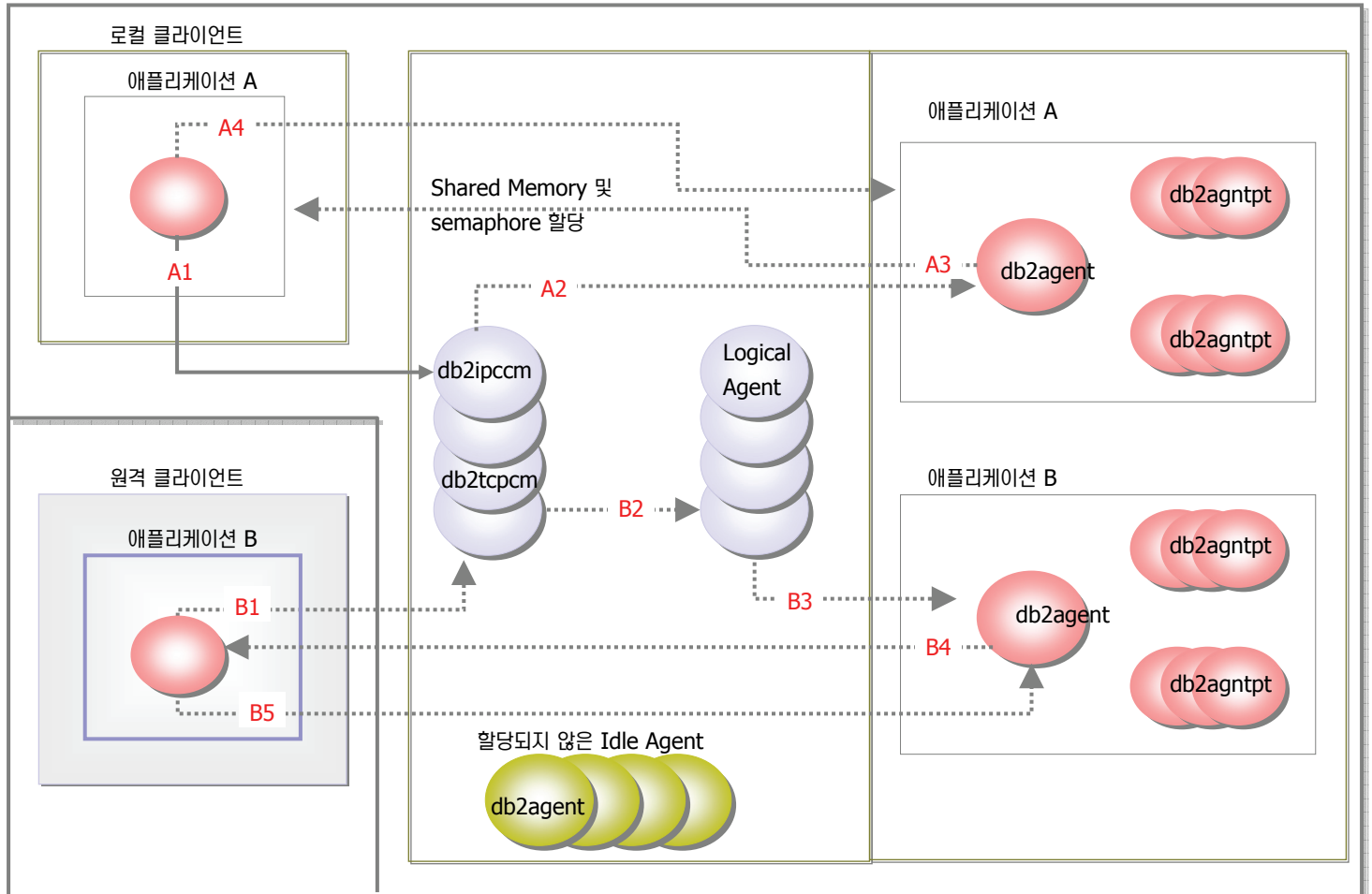
	오라클	DB2
네트워크	<ul style="list-style-type: none"> Oracle Net 서비스는 클라이언트 또는 Middle-Tier 응용프로그램에서 오라클 서버로 연결 요청을 담당합니다. 	<ul style="list-style-type: none"> DB2는 인스턴스 구동시 서버에서 설정한 프로토콜에 맞는 프로세스를 자동으로 구동시킵니다. 예를 들면 클라이언트와 서버간의 프로토콜을 TCPIP로 설정시 db2tccm 프로세스가 자동으로 시작됩니다.
네트워크 담당 프로세스	<ul style="list-style-type: none"> Oracle Net에 포함된 Listener 프로세스가 담당합니다. 	<ul style="list-style-type: none"> DB2에서는 사용하는 프로토콜 (TCPIP, IPC, IPX 등)에 따라 각 해당 프로세스가 담당합니다. <ul style="list-style-type: none"> db2tccm- TCPIP 통신 담당 db2ipccm -IPC 통신담당 (로컬컴퓨터) db2snacm -APPC 통신담당
서버 네트워크 설정	<ul style="list-style-type: none"> 서버 네트워크 설정 방법에는 다음과 같이 두가지 방식이 존재합니다. <ul style="list-style-type: none"> Listener.ora 파일에 프로토콜 및 포트 번호를 설정합니다. Net Manager 또는 Net Configuration Assitant를 이용하여 네트워크를 설정합니다. 	<ul style="list-style-type: none"> 서버 네트워크 설정 방법에는 다음과 같이 두가지 방식이 존재합니다. <ul style="list-style-type: none"> DBMS Configuration File에 포트 번호를 설정하고 Resistry Profile에 프로토콜을 설정합니다. 제어센타를 통해 네트워크를 설정합니다.
네트워크 관리 프로그램	<ul style="list-style-type: none"> Lsnrctl 이라는 유틸리티를 통해 Listener 프로세스를 제어합니다. <p>예) 시작: LSNRCTL> start 중지: LSNRCTL> stop</p>	<ul style="list-style-type: none"> DB2 Network을 제어하는 유틸리티는 존재하지 않습니다. 그러나 네트워크 담당 프로세스 구동 여부를 확인하기 위해 "db2_local_ps" 명령어를 이용하여 네트워크 담당 프로세스를 확인할 수 있습니다. 인스턴스 구동시 네트워크 담당 프로세스가 함께 구동되므로 특별히 구동 및 관리할 필요가 없습니다. <p>예) db2start → 인스턴스 구동 명령어로 네트워크 프로세스가 함께 구동됩니다.</p>

* 부연설명

오라클 서버의 네트워크 설정 파일인 listener.ora 파일은 Text 파일로 모든 Editor 프로그램에서 열어 보실 수 있으며 편집/수정이 가능합니다. 그러나 DB2에서 네트워크 설정 파일은 Binary 파일로 명령어를 통해서만 읽기/수정이 가능합니다. 이는 DB2 일반 사용자 혹은 잘 모르는 사용자가 함부로 네트워크 설정값을 건드리지 못하도록 사전에 방지하기 위해서입니다.

02 DB2 접속 프로세스 절차

Point 클라이언트의 접속 요청 처리 절차는 아래와 같습니다. 로컬 클라이언트 및 원격 클라이언트의 요청 절차는 사용하는 프로토콜이 다르므로 처리절차가 다릅니다.




로컬 서버 네트워크 절차

- A1. 로컬 클라이언트는 db2ipccm 프로세스를 통해 통신을 설정합니다.
- A2. db2ipccm은 향후 로컬 클라이언트의 애플리케이션 요청을 담당할 coordinator agent가 될 db2agent EDU에 요청을 보냅니다.
- A3. db2agent는 로컬 클라이언트 응용프로그램과 db2 coordinator agent의 통신을 위해 shared memory와 semaphore를 할당 후 로컬 클라이언트 응용프로그램에 연락합니다.
- A4. 로컬 클라이언트 애플리케이션 A는 데이터베이스에 접속합니다.

원격 서버 네트워크 절차

- B1. 원격 클라이언트는 db2tcpm 프로세스를 통해 TCP/IP 통신을 설정합니다. 만약 다른 프로토콜을 설정하였다면 그 프로토콜을 담당하는 프로세스를 통해 통신을 설정합니다.
- B2. db2tcpm은 논리적인 agent에 접속요청을 보냅니다.
- B3. Logical Agent는 향후 원격 클라이언트의 요청을 담당할 coordinator agent가 될 db2agent에 접속요청을 전달합니다.
- B4. coordinator agent는 원격 클라이언트 애플리케이션에 연락합니다.
- B5. 원격 클라이언트 애플리케이션 B는 데이터베이스에 접속합니다.

Point  클라이언트에서 데이터베이스 서버에 접속하려면 서버 주소 및 Listen 포트, 그리고 접속하고자 하는 데이터베이스 이름을 설정해야 합니다. 이 곳에서는 오라클과 DB2의 클라이언트 네트워크 설정방법 및 관리방법을 비교하도록 합니다.

	오라클	DB2
클라이언트 네트워크 설정 방법	<ul style="list-style-type: none"> ▪ 필수 정보 <ul style="list-style-type: none"> ▪ 사용 프로토콜 ▪ 서버 주소 ▪ 포트 번호 ▪ 인스턴스 이름 (혹은 DB Name) ▪ 클라이언트 네트워크 설정 방법에는 다음과 같이 두가지 방식이 존재합니다. <ul style="list-style-type: none"> ▪ Net Manager 또는 Net configuration Assistant를 통해 네트워크를 설정합니다. ▪ tnsnames.ora 파일에 필수 정보를 입력하여 설정합니다. 	<ul style="list-style-type: none"> ▪ 필수 정보 <ul style="list-style-type: none"> ▪ 사용 프로토콜 ▪ 서버 주소 ▪ 포트 번호 ▪ DB Name ▪ 클라이언트 네트워크 설정 방법에는 다음과 같이 두가지 방식이 존재합니다. <ul style="list-style-type: none"> ▪ 구성 지원 프로그램을 통해 네트워크를 설정합니다. ▪ "catalog" 명령어를 이용해 클라이언트 네트워크를 설정합니다.
클라이언트 네트워크 설정 확인 방법	<ul style="list-style-type: none"> ▪ 클라이언트 네트워크 설정 확인 방법에는 다음과 같이 두가지 방식이 존재합니다. <ul style="list-style-type: none"> ▪ Net Configuration Assistant 를 통해 확인합니다. ▪ tnsnames.ora 파일을 열어 확인합니다. 	<ul style="list-style-type: none"> ▪ 클라이언트 네트워크 설정 확인 방법에는 다음과 같이 두가지 방식이 존재합니다. <ul style="list-style-type: none"> ▪ 구성 지원 프로그램을 통해 네트워크 설정을 확인합니다. ▪ "list node directory" 와 "list db directory" 명령어를 통해 네트워크 정보를 확인합니다.

참고) 클라이언트의 네트워크 설정방법은 편지를 보낼때 받는 사람의 아파트 주소 (서버 주소와 유사) 그 아파트의 몇호 (포트) , 그리고 받을 사람 (DB Name) 을 지정하는 것과 같은 원리입니다. 접속할 서버 및 그 서버에 설치된 프로그램 중 요청을 보낼 프로그램은 포트번호로 찾으실 수 있습니다.

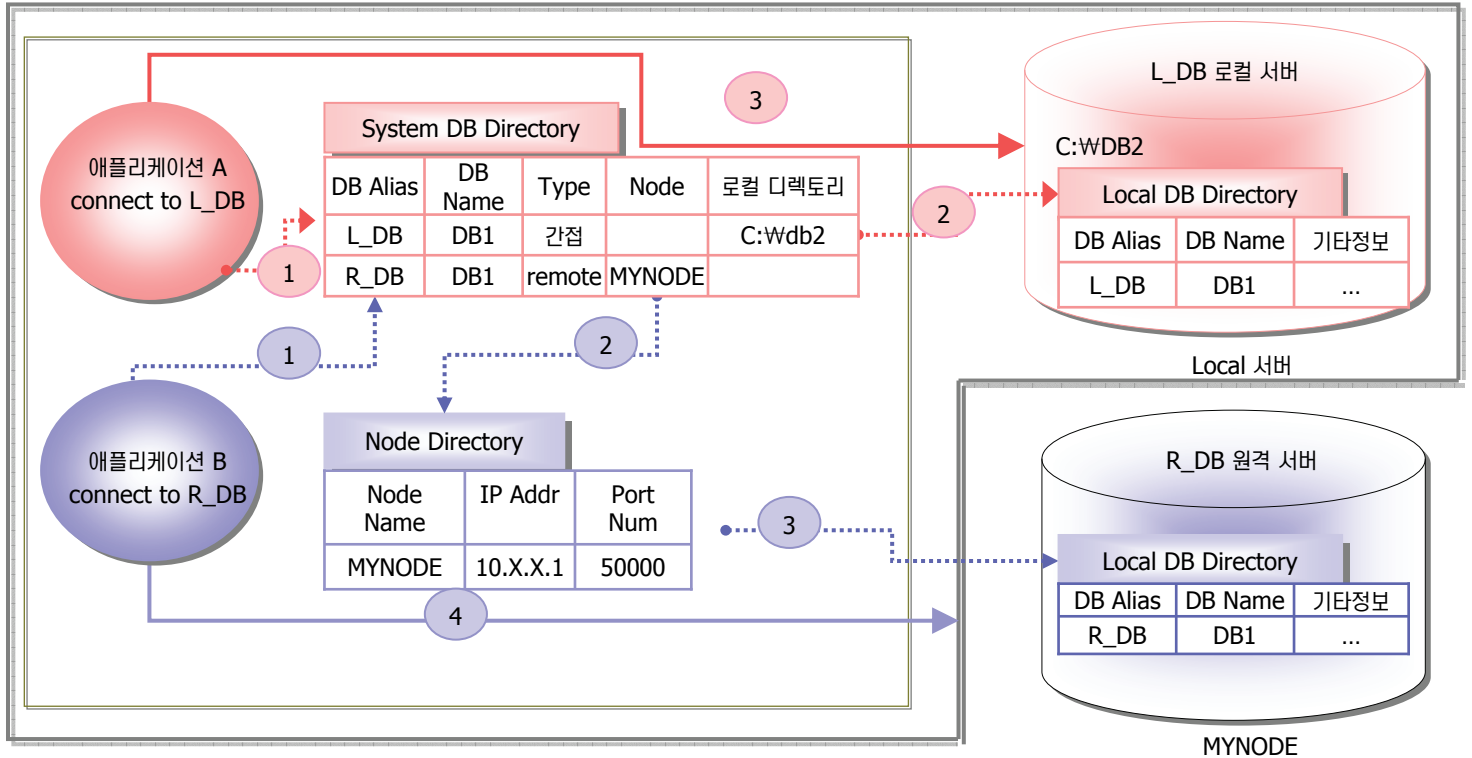
부연설명

오라클 클라이언트의 네트워크 설정 파일인 tnsnames.ora 파일은 Text 파일로 모든 Editor 프로그램에서 열어 볼 수 있으며, 편집/수정이 가능합니다. 그러나 DB2 클라이언트에서 네트워크 설정은 Catalog 명령어를 통해 sqllib/sqlnodir/sqlnodir 파일과 sqllib/sqlbdbir/sqlbdbir Binary파일에 네트워크 정보를 등록합니다. 이는 DB2를 잘 모르는 사용자가 함부로 네트워크 설정을 수정하지 못하게 하기 위해서입니다.

DB2에서는 sqllib/sqlnodir 파일을 직접 언급하지 않고 node directory라 하며 마찬가지로 sqllib/sqlbdbir/sqlbdbir 파일도 db directory라 합니다.

04 DB2 클라이언트의 접속 처리 절차

Point DB2 클라이언트의 접속은 아래와 같이 로컬 데이터베이스에 접속할 경우와 원격 데이터베이스에 접속할 경우 참조하는 파일이 다릅니다. 로컬 데이터베이스 서버 및 원격 데이터베이스 서버의 접속 처리절차를 확인합니다.



참고)

- 간접 : 같은 머신에 존재하는 로컬 DB 서버를 의미합니다.
- Remote : 원격에 존재하는 DB 서버를 의미합니다.
- System db directory, local db directory, node db directory는 다음 페이지에서 다루도록 합니다.

로컬 클라이언트 접속 절차

로컬 클라이언트의 접속 절차는 다음과 같습니다.

1. System DB Directory 파일에서 접속하고자 하는 데이터베이스의 정보가 있는지 확인한 후 (오라클에서는 tnsnames.ora 파일을 확인하는 경우임) 데이터베이스의 존재 여부 및 Type을 확인합니다.
2. Type이 간접일 경우에는 Local 서버 정보를 저장한 local db directory를 읽어 DB 위치를 파악하여 클라이언트에게 정보를 보냅니다.
3. 로컬 클라이언트와 접속합니다.

원격 클라이언트 접속 절차

원격 클라이언트의 접속 절차는 다음과 같습니다.

1. System DB Directory 파일에서 접속하고자 하는 데이터베이스의 정보가 있는지 확인한 후 (오라클에서는 tnsnames.ora 파일을 확인하는 경우임) 데이터베이스의 존재 여부 및 Type을 확인합니다.
2. Type이 원격일 경우 node directory를 찾아 시스템 정보와 포트 정보를 확인합니다.
3. 원격 데이터베이스에 접속하여 원격 서버에 존재하는 local db directory를 읽어 DB 위치를 파악하여 클라이언트에게 정보를 보냅니다.
4. 원격 클라이언트와 접속합니다.

Point DB2 클라이언트에서 서버에 접속하기 위해서는 접속하고자 하는 서버의 정보가 필요하며 이러한 서버 정보를 관리하는 목록을 디렉토리라 합니다. 디렉토리는 바이너리 파일로 DB2 명령어인 `catalog` 명령어를 이용하여 제어합니다.

디렉토리 개념

- 디렉토리는 데이터베이스 서버 위치 정보, 파일 위치, 데이터베이스가 `listen` 하고 있는 포트 정보 등을 관리하는 디렉토리로 다음과 같이 3가지 유형이 존재합니다.
 - Local DB Directory
 - 데이터베이스가 존재하는 로컬 서버의 디렉토리에 이 목록이 생성됩니다.
 - “Create Database ~ ” 명령어 수행시 자동으로 이 디렉토리에 데이터베이스 정보가 등록되며 “drop Database~ ” 명령어 수행시 데이터베이스 정보가 삭제됩니다.
 - 데이터베이스 접속시 이용되며 DB2 클라이언트만 설치된 시스템에는 이 디렉토리가 존재하지 않습니다.
 - 데이터베이스 이름, 별명, 타입, 그리고 위치정보가 저장됩니다.
 - “list db directory on <데이터베이스가 존재하는 PATH 또는 Drive>” 로 Local DB directory를 확인할 수 있으며, “on <데이터베이스가 존재하는 PATH 또는 Drive>” 생략시 system db directory를 확인합니다.
 - System DB Directory
 - DB2 클라이언트가 접속할 로컬 데이터베이스 서버 및 원격 데이터베이스 서버의 정보가 존재합니다.
 - 로컬 서버의 경우 데이터베이스 생성시 자동으로 데이터베이스 정보가 등록됩니다.
 - 데이터베이스 이름, 별명, 타입 그리고 원격 서버일 경우 노드이름이 포함되어 있습니다.
 - 데이터베이스가 로컬 서버일 경우 Local Database Directory에 데이터베이스의 위치가 저장됩니다.
 - 데이터베이스가 원격 서버일 경우 Node Directory에 데이터베이스가 존재하는 시스템 정보가 저장됩니다.
 - 클라이언트에서 원격 서버에 접속시 반드시 존재해야 하는 디렉토리입니다.
 - `sqldbdir` 파일이 System DB Directory의 물리적 바이너리 파일입니다
 - “list db directory” 명령어로 system db Directory를 확인할 수 있습니다.
 - Node Directory
 - DB2 클라이언트가 접속할 원격 데이터베이스 서버의 시스템 정보가 존재합니다.
 - 원격 서버의 IP Addr, Listen Port 정보가 존재합니다.
 - 클라이언트에서 원격 서버에 접속시 반드시 존재해야 하는 디렉토리입니다.
 - `sqllib/sqlnodir/sqlnodir` 파일이 Node Directory의 물리적 Binary파일입니다.
 - “list node directory” 명령어로 node Directory를 확인할 수 있습니다.

Catalog / Uncatalog 명령어

- “System DB Directory” 및 “Node Directory”에 Node 혹은 DB 정보를 등록/삭제하는 명령어입니다.
- 데이터베이스의 존재 유무, 시스템의 존재 여부, 설정된 포트의 Listen 여부는 “catalog” 명령어 수행시 체크하지 않고 접속 요청시에 체크합니다.

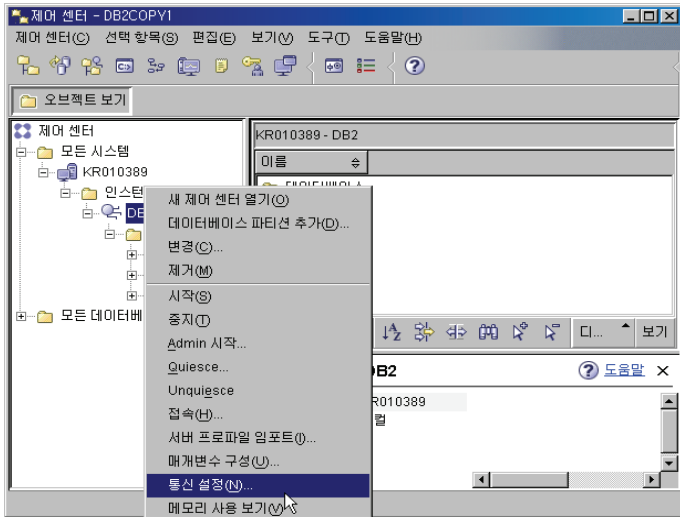


로컬 데이터베이스가 생성될때 system db directory와 local db directory에 데이터베이스 정보를 자동으로 등록하므로 데이터베이스가 생성된 로컬 클라이언트에서는 접속 설정이 필요치 않습니다. 그러나 원격 데이터베이스 서버 접속시에는 클라이언트측에서 접속정보가 필요하므로 Catalog 명령어를 통해 System 정보와 DB 정보를 “System db directory”와 “node directory”에 등록해야 합니다. 또한 로컬 데이터베이스 서버를 원격 서버인 것처럼 등록도 가능합니다.

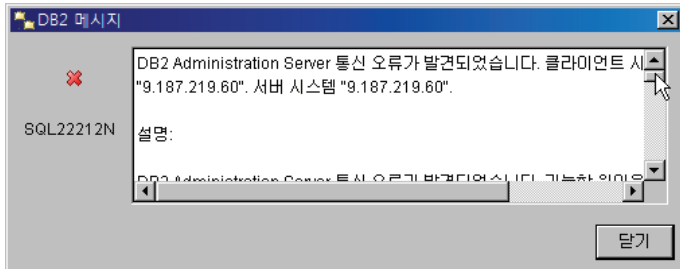
06 제어센타를 이용한 서버 네트워크 설정

Point 제어센타를 이용하여 서버 네트워크를 설정합니다.

1. 시작 => 모든 프로그램 => IBM DB2 => DB2COPY1(디폴트) => 일반 관리 도구 => 제어센타를 실행합니다. 아래와 같이 제어센타가 나타나면 모든 시스템 => <시스템 이름> => 인스턴스 => DB2를 선택한 후 오른쪽 버튼을 클릭하여 "통신 설정"을 선택합니다.



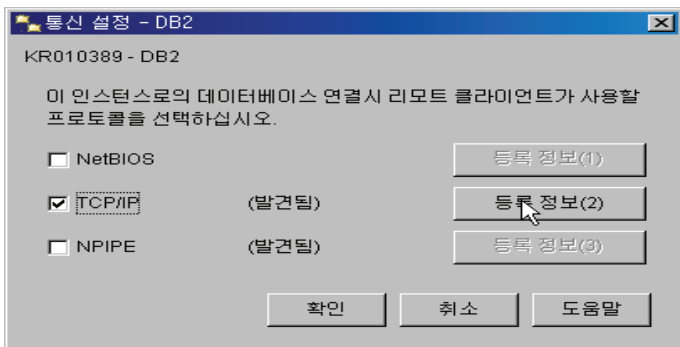
2. 아래와 같이 SQL22212N 에러가 발생하면 명령창에서 "db2admin start" 를 이용하여 Admin instance를 실행하도록 합니다.



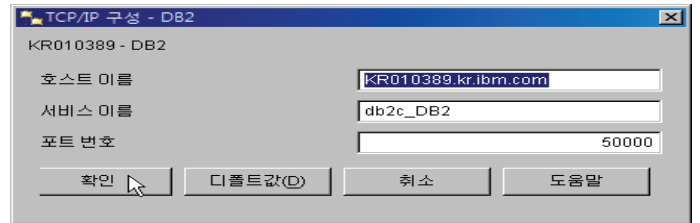
C:\WLAB> db2admin start

SQL4406W DB2 Administration Server가 시작되었습니다.

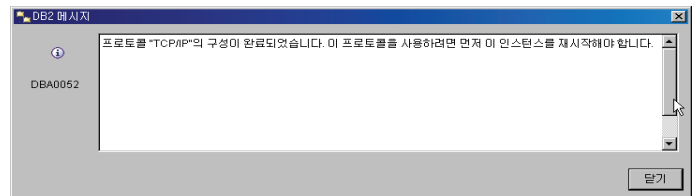
3. 아래와 같이 통신할 프로토콜을 지정한후 등록 정보를 클릭합니다.



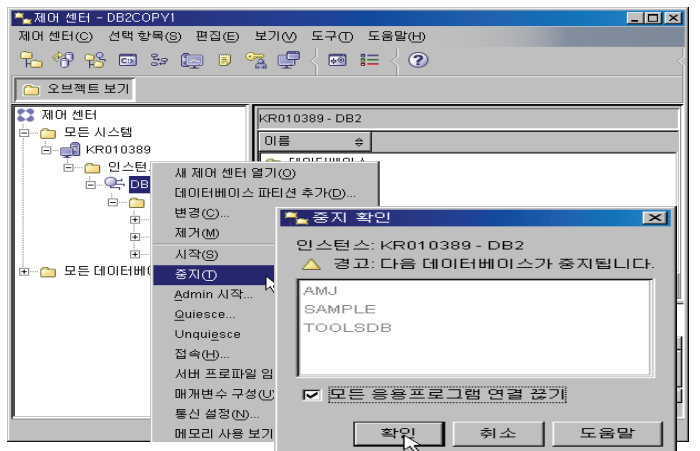
4. 호스트 이름을 확인한 후 Listen 할 포트 번호와 그 포트 번호에 대한 서비스 이름을 지정합니다.이 곳에서는 디폴트 값을 사용하도록 합니다.



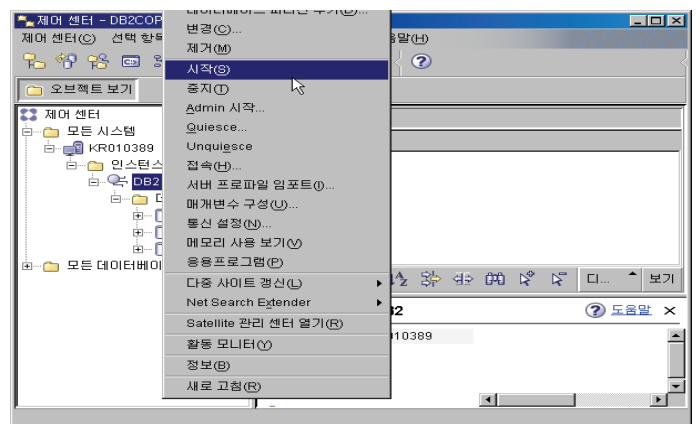
5. 이 프로토콜을 사용하려면 먼저 인스턴스를 재시작해야 된다는 메시지를 확인할 수 있습니다. 확인 버튼을 클릭합니다.




6. 오른쪽 패널에서 DB2를 다시 선택한 후 오른쪽 버튼을 눌러 중지지를 클릭합니다. 강제로 모든 애플리케이션 연결을 끊을 경우 아래와 같이 "모든 응용 프로그램 연결 끊기"를 선택하고 확인을 클릭합니다.



7. DB2 인스턴스 중지가 성공적으로 완료되었다는 메시지를 확인한 후 다시 인스턴스를 선택하여 시작합니다.



Point  명령어를 이용하여 서버 네트워크를 설정합니다.

1. 앞에서 설정한 DB2COMM을 지운 후에 이번에는 명령어를 이용하여 Profile Registry에 DB2 서버와 클라이언트간에 통신할 프로토콜을 명령어를 이용하여 재설정합니다.

```
C:\WLAB> db2set -all
[i] DB2COMM=TCPIP
C:\WLAB> db2set DB2COMM=
C:\WLAB> db2set DB2COMM=TCPIP
C:\WLAB> db2set -all
[i] DB2COMM=TCPIP
```

2. Database Manager Configuration에 DB2 서버에서 Listen하고 있을 포트를 설정합니다. 아래 예제처럼 설정된 포트가 있는지 "get dbm cfg" 명령어를 통해 확인한 후 설정된 값이 없거나 변경할 경우 "update dbm cfg"를 사용하여 원하는 포트로 설정합니다.

```
c:\WLAB> db2 get dbm cfg | find "SVCENAME"
TCP/IP 서비스 이름(SVCENAME) = db2c_DB2
c:\WLAB> db2 update dbm cfg using SVCENAME 50000
DB20000I UPDATE DATABASE MANAGER CONFIGURATION 명령이 완료되었습니다.
SQL1362W 즉각적인 수정을 위해 제출된 하나 이상의 매개변수가 동적으로 변경되지 않았습니다. 클라이언트 변경사항은 다음에 응용프로그램이 시작되거나 TERMINATE 명령이 발행된 이후에 적용됩니다. 서버 변경사항은 다음에 DB2START 명령을 수행할 때 적용됩니다.
```

3. 위와 같이 프로토콜과 사용할 포트를 설정한 후 DB2 서버가 설정된 새로운 포트와 프로토콜로 클라이언트 요청을 받아들일 수 있도록 인스턴스를 중지한 후 다시 시작합니다.

```
C:\WLABWUNIT5WNET> db2stop [ force ]
2007-10-30 19:30:05 0 0 SQL1064N DB2STOP 처리가 완료되었습니다
SQL1064N DB2STOP 처리가 완료되었습니다.
C:\WLABWUNIT5WNET> db2start
2007-10-30 19:30:05 0 0 SQL1064N DB2START 처리가 완료되었습니다
SQL1064N DB2START 처리가 완료되었습니다.
```

4. DB2 서버 네트워크 설정은 모두 완료되었습니다.



Database Manager Configuration File에서 포트를 설정할때 OS의 services 파일을 확인한 후 포트를 설정해야 합니다. 시스템의 다른 프로그램이 사용하고 있는 포트를 확인하여 충돌하지 않도록 체크합니다. services 파일에 Listen할 포트번호와 서비스 이름을 등록한후 Configuration 파일에 "update dbm cfg using SVCENAME < 서비스이름> "" 명령어를 이용하여 설정합니다.

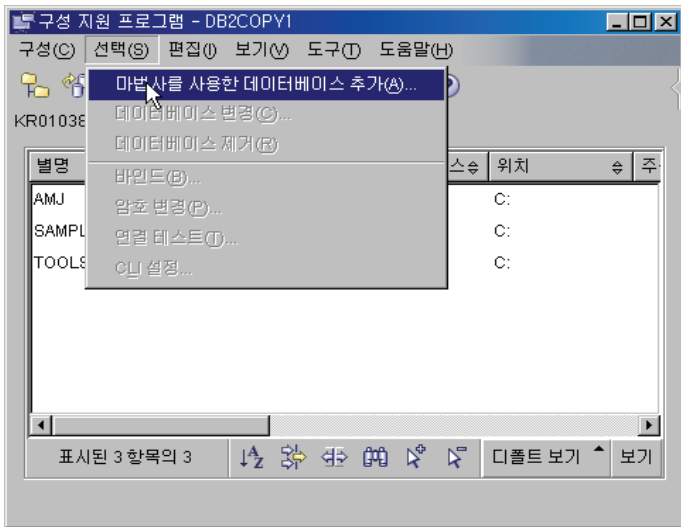
참고로 Windows의 경우 C:\WINDOWS\system32\drivers\etc 에 services 파일이 존재하며 Unix/linux의 경우 /etc에 services 파일이 존재합니다.

08 구성 지원 프로그램을 이용한 클라이언트 설정

네트워크 설정 비교

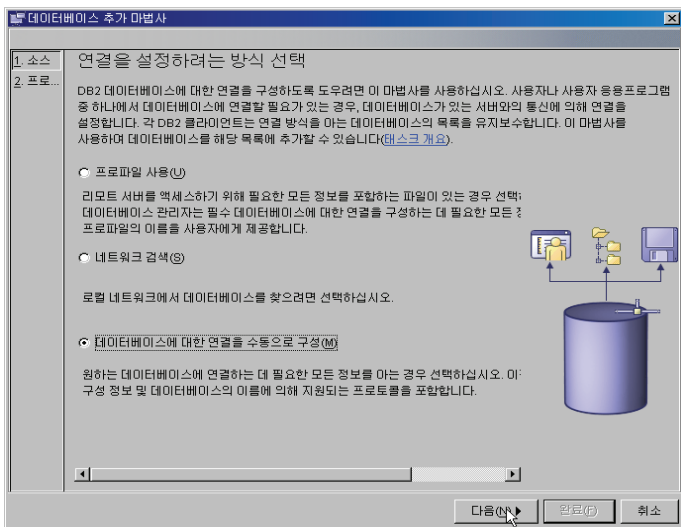
Point 구성 지원 프로그램을 이용하여 클라이언트 접속을 설정합니다.

1. 시작 => 모든 프로그램 => IBM DB2 => DB2COPY1(디폴트) => 설정 도구 => 구성 지원 프로그램을 실행한 후 아래와 같은 화면이 나타나면 메뉴에서 선택 => 마법사를 사용한 데이터베이스 추가 버튼을 클릭합니다.

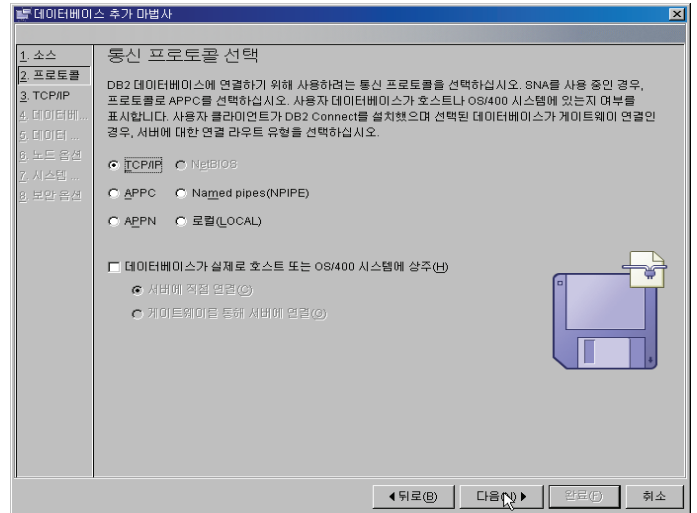


2. "연결 설정 방식"을 선택합니다. 선택방식에는 다음과 같이 3가지 방식이 존재합니다. 상황에 맞는 방식을 선택합니다. 이 곳에서는 IP 주소와 Port 번호를 알고 있으므로 수동 구성을 선택합니다.

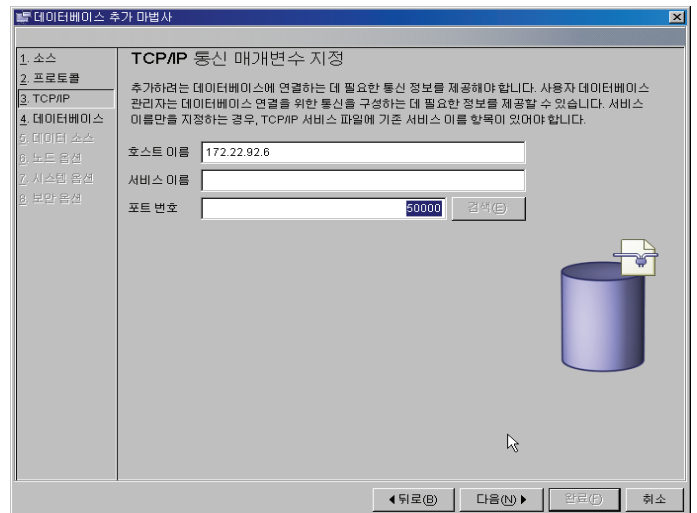
- 프로파일 선택: 다른 서버에서 export 한 프로파일이 존재할 경우
- 네트워크 검색: 네트워크 검색을 통해 로컬 네트워크의 데이터베이스를 자동 검색
- 데이터베이스에 대한 연결을 수동으로 구성: 시스템 IP 주소 및 포트 번호를 알고 있을 경우



3. 클라이언트와 서버간에 통신할 프로토콜을 지정합니다. TCP/IP를 선택하고 "다음" 버튼을 클릭합니다.



4. 서버의 호스트 이름 또는 IP 주소와 포트번호를 지정합니다. 이때 호스트 이름을 입력할 경우 호스트 이름은 OS의 hosts 파일에 IP 주소가 등록되어 있어야 합니다.



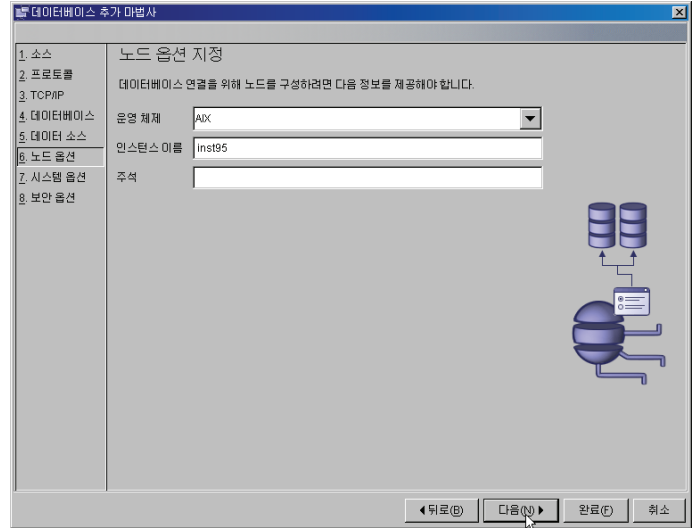
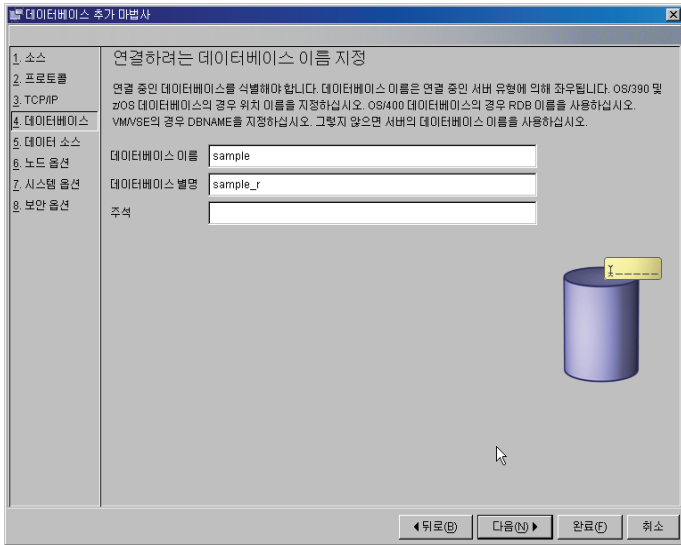
09 구성 지원 프로그램을 이용한 클라이언트 설정

네트워크 설정 비교

Point 구성 지원 프로그램을 이용하여 클라이언트 접속을 설정합니다.

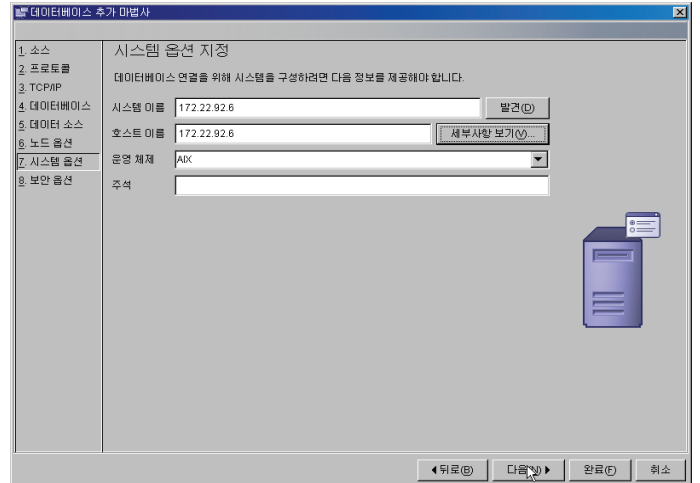
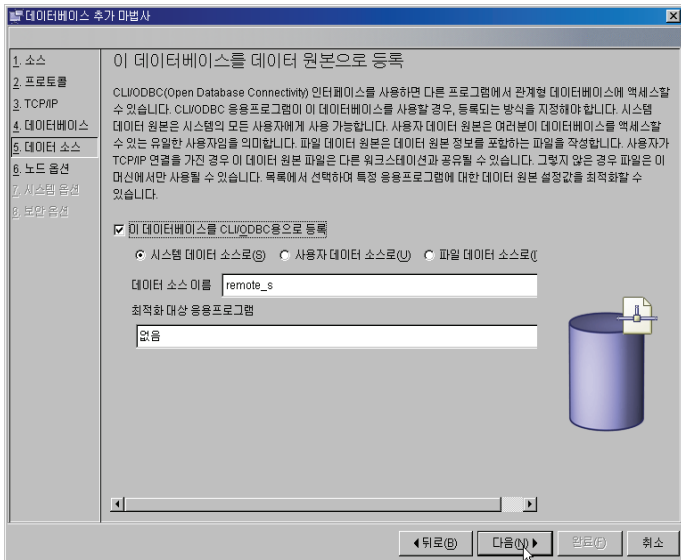
5. 연결하려는 데이터베이스 이름을 설정합니다. 이때 데이터베이스 이름은 접속할 데이터베이스 이름을 지정하고 데이터베이스 별명은 다른 데이터베이스와 구별하기 위해 유일한 이름을 지정해야 합니다. 이곳에서는 로컬에 이미 SAMPLE 별명을 가진 데이터베이스가 존재하므로 다른 데이터베이스와 구별짓기 위해 데이터베이스 별명을 다른 유일한 이름으로 지정합니다.

7. 원격 서버의 운영 체제 및 인스턴스 이름을 지정합니다. 이때 인스턴스 이름은 다른 인스턴스와 구별하기 위해 임의로 작성해도 됩니다.



6. CLI/ODBC 애플리케이션이 데이터베이스에 접속할 수 있도록 CLI/ODBC 등록합니다.

8. 시스템 옵션 지정을 확인한 후 "다음" 버튼을 클릭합니다.

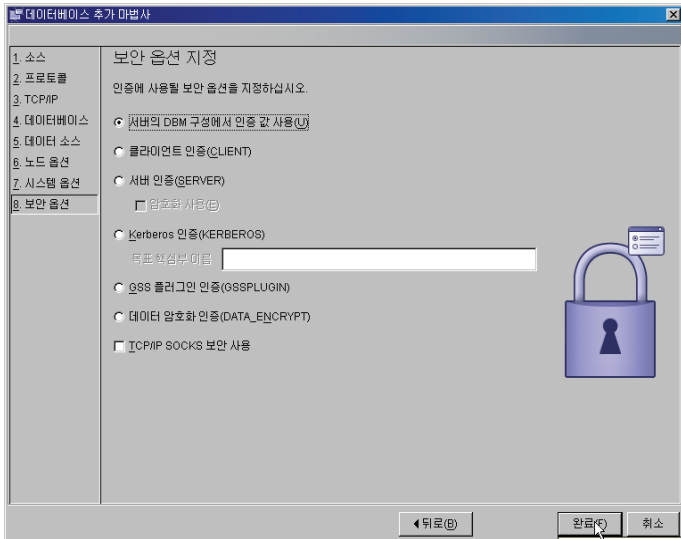


10 구성 지원 프로그램을 이용한 클라이언트 설정

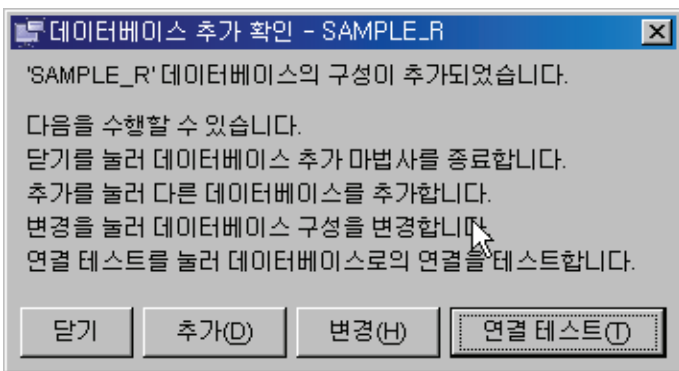
네트워크 설정 비교

Point 구성 지원 프로그램을 이용하여 클라이언트 접속을 설정합니다.

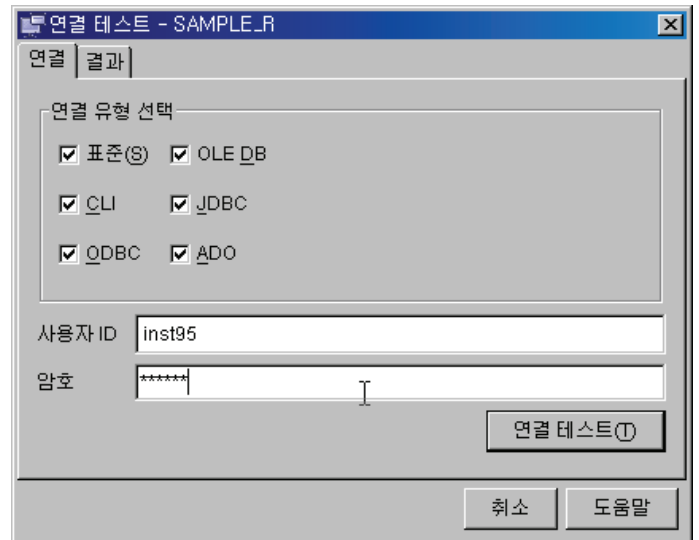
9. 보안 옵션을 지정하여 인증이 어느 시점에서 이루어 질지 지정합니다. 데이터베이스 접속 인증을 서버측에서 할 것인지 아니면 클라이언트 측에서 할 것인지 또는 Kerberos와 같은 Third-Party Security 프로그램을 이용할 것인지 지정합니다. 이곳에서는 Default 인 "서버 DBM 구성에서 인증값 사용"을 선택합니다.



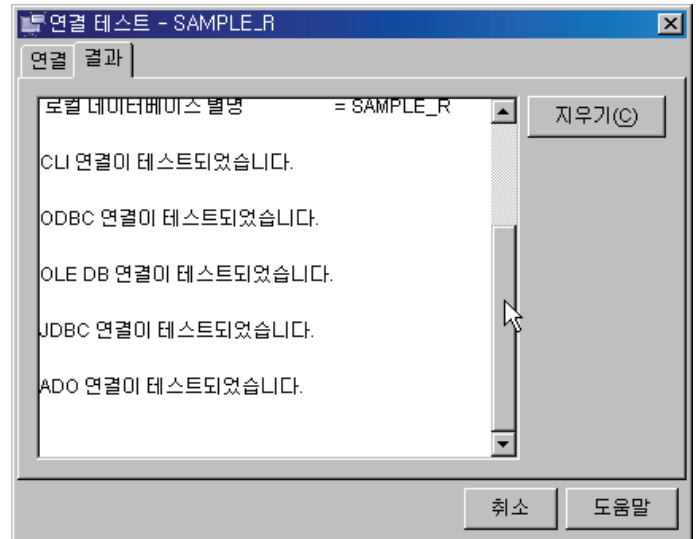
10. 완료버튼을 누르면 다음과 같이 데이터베이스 추가 확인창이 나타나며 연결 테스트를 수행할 수 있습니다.



11. 테스트할 연결 유형을 선택한 후 사용자 ID와 암호를 입력합니다. 현재 이곳에서는 "서버의 DBM 구성에서 인증값 사용"을 선택하였고 서버의 DBM 구성은 "서버 인증"이므로 서버 OS의 사용자 ID와 암호를 입력해야 합니다.



12. 연결 결과가 성공일 경우, 아래와 같이 테스트가 완료되었다는 메시지를 확인할 수 있습니다.



11 CATALOG 명령어를 이용한 네트워크 설정

네트워크 설정 비교

Point 명령어를 이용하여 서버 네트워크를 설정합니다.

1. 일단 앞페이지의 구성 지원 프로그램을 통해 등록된 Node Directory 를 "list node directory" 명령어를 이용하여 확인한 후 uncatlog 명령어를 이용하여 노드를 제거합니다. 이 때 노드 이름은 "list node directory"에서 보여주는 이름을 사용합니다

```
C:> db2
db2 => list node directory
```

```
Node Directory
Node name          = NODENEO
Directory entry type = LOCAL
Protocol           = TCPIP
Hostname           = 172.22.71.2
Service name       = 50000
```

```
db2 => uncatlogp node NODNEO
```

```
DB20000I UNCATALOG NODE 명령이 완료되었습니다.
DB21056W 디렉토리 변경사항은 디렉토리 캐시를 새로 고쳐야 적용됩니다.
```

2. catalog 명령어를 이용하여 Node Directory에 서버와 Listen 할 포트를 등록합니다.

```
db2 => catalog tcpip node <Node Name> remote < Server 주소> server < Port 번호 혹은 port 이름>
```

```
DB20000I CATALOG NODE 명령이 완료되었습니다.
DB21056W 디렉토리 변경사항은 디렉토리 캐시를 새로 고쳐야 적용됩니다.
```

3. 위에서 등록된 노드 정보를 "List Node Directory" 명령어로 확인합니다.

참고) 로컬 서버의 경우 바로 local db directory에서 데이터베이스 정보를 읽어오므로 이 과정이 필요치 않습니다.

4. Catalog 명령어를 통해 System DB Directory에 DB정보를 등록합니다. 이때 등록하고자 하는 DB가 어떤 노드에 위치하는지 입력해야 하며 System DB Directory에 등록하는 데이터베이스의 이름은 같더라도 별명은 유일해야 합니다. 이는 DB2 접속 명령어는 DB Alias를 이용하여 데이터베이스에 접속하기 때문입니다.

```
db2 => catalog db < DB Name> as < Unique alias name >
at node < Node Name>
```

```
DB20000I CATALOG DB 명령이 완료되었습니다.
DB21056W 디렉토리 변경사항은 디렉토리 캐시를 새로 고쳐야 적용됩니다.
```

5. 위에서 등록된 DB 정보는 "List DB Directory" 명령어로 다음과 같이 확인할 수 있습니다.

참고) "System DB Directory"에는 로컬 서버 정보와 Remote 서버 정보가 모두 포함되어 있습니다.

```
db2 => list db directory
```

```

:
:
데이터베이스 별명 = SAMPLE
노드 이름          = NODENEO
디렉토리 항목 유형 = 리모트
```

6. 클라이언트에서 데이터베이스 서버에 접속합니다.

```
db2 => connect to <DB 별명> user < User ID> using
< Password >
```

7. 앞에서 등록된 DB 정보를 Uncatalog 명령어를 통해 "System DB Directory"에서 삭제할 수 있습니다. 더 이상 접속할 필요가 없는 데이터베이스에 대한 정보를 삭제할 경우 사용할 수 있습니다.

```
db2 => uncatlog node <Node Name>
```

8. 앞에서 등록된 노드정보를 Uncatalog 명령어를 통해 "Node Directory"에서 삭제할 수 있습니다. 더 이상 접속할 필요가 없는 노드에 대한 정보를 삭제할 경우 사용할 수 있습니다.

```
db2 => uncatlog db <DB Name>
```

네트워크 관련 Q/A

- ① “SQL30081N 통신오류가 발생되었습니다.” 라는 에러가 발생합니다. 어떻게 해결해야 되는지요?

먼저 ping 명령어로 네트워크에 문제가 없는지 확인합니다. 그리고 DB2 데이터베이스가 시작되었는지 확인합니다. 만약 네트워크에 문제가 없고 DB2가 시작되었다면 db2set -all 명령어를 이용하여 DB2COMM Registry 변수가 사용하고 있는 프로토콜로 설정이 되어 있는지 확인합니다. 또한 “get dbm cfg” 명령어를 이용하여 SVCENAME이 적절한 포트로 설정이 되어 있는지 확인한 후 재접속 합니다. 만약 DB2COMM 및 SVCENAME이 사용할 프로토콜 및 포트로 설정이 되어 있지 않다면 수정한 후 DB2를 재시작합니다.

- ② Connect시 접속 권한은 어떻게 되는건지요?

로컬 클라이언트에서 로컬 데이터베이스 서버로 접속시에는 OS에 접속한 사용자는 이미 운영체제에서 인증을 받은 것으로 인식하고 로컬 사용자를 이용해 로컬 데이터베이스에 접속합니다. 이런 이유로 로컬 데이터베이스 접속시에는 사용자 이름과 암호를 입력하지 않습니다. 그러나 원격 데이터베이스 서버에 접속시에는 “DBM CFG”의 AUTHENTICATION 파라미터를 어떻게 설정했는 지에 따라 달라집니다. Default 로는 “SERVER” 로 설정되는데 이것은 서버측에서 사용자를 인증하겠다는 의미입니다. 즉 connect 시 데이터베이스 서버가 설치된 OS 사용자와 암호를 입력해야 합니다. 자세한 사항은 Security 부분에서 다시 다루도록 합니다.

- ③ “list db directory” 명령어로 확인한 결과 생성할 데이터베이스가 없는데도 불구하고 데이터베이스를 생성하는데 “이미 데이터베이스가 존재한다”라는 메시지와 함께 데이터베이스 생성이 실패합니다. 원인이 무엇인지요?

“drop database” 명령어를 이용하여 데이터베이스를 삭제하지 않고 uncatalog 명령어를 이용하여 system db directory에 데이터베이스 정보만 삭제할 경우 이런 에러가 발생합니다. 즉 system db directroy에는 데이터베이스 정보가 없고 local db directory에만 데이터베이스 정보가 있는 경우입니다. “catalog” 명령어를 이용하여 system db directory 에 데이터베이스 정보를 입력하면 해결이 가능합니다. 참고로 Local db directory는 catalog 및 uncatalog 명령어를 이용하여 등록 및 삭제가 불가능하며 데이터베이스 생성시 자동으로 등록되며 데이터베이스 drop시 자동으로 디렉토리에서 삭제됩니다



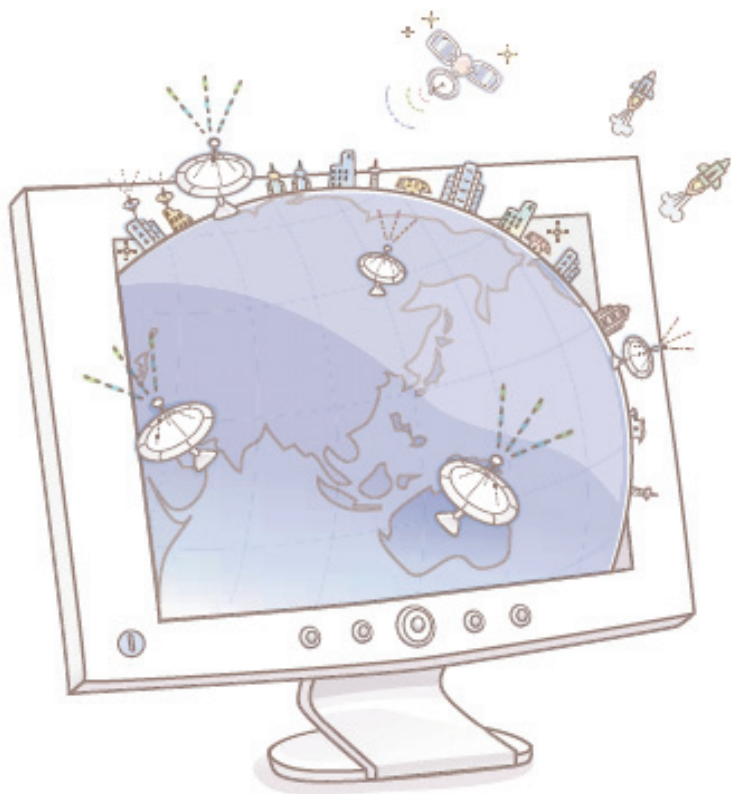
Unit 07 : 테이블 스페이스 비교

오라클 사용자를 위한 DB2 가이드


물리적 저장 모델 비교

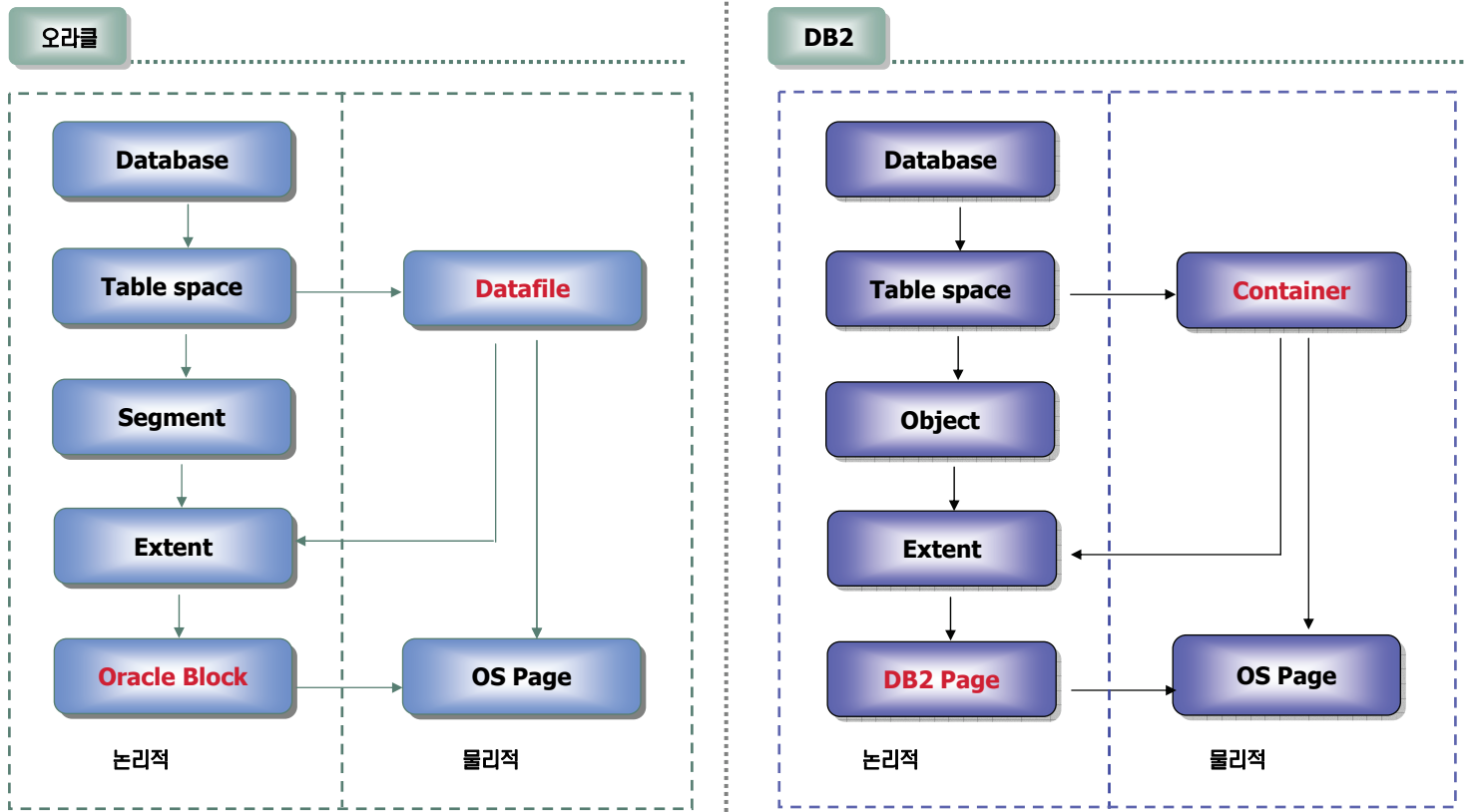
테이블 스페이스 관리 비교

DB2 테이블 스페이스 생성 및 관리




01 물리적 저장 모델 비교


Point  아래는 오라클과 DB2의 계층적 물리적, 논리적 구조를 비교한 그림입니다.



Oracle	DB2	설명
<ul style="list-style-type: none"> 테이블 스페이스 	<ul style="list-style-type: none"> 테이블 스페이스 	<ul style="list-style-type: none"> 테이블 스페이스는 테이블을 저장하는 논리적 개념입니다. 테이블 스페이스를 구성하는 물리적인 개념은 DB2에서는 Container이며 오라클에서는 데이터 파일입니다.
<ul style="list-style-type: none"> 세그먼트 <ul style="list-style-type: none"> 테이블, 인덱스, 롤백 	<ul style="list-style-type: none"> 오브젝트 <ul style="list-style-type: none"> 테이블, 인덱스 	<ul style="list-style-type: none"> 오라클의 세그먼트를 DB2에서는 오브젝트라 명명합니다. 물리적 디스크 공간을 차지하는 테이블 혹은 인덱스가 해당됩니다.
<ul style="list-style-type: none"> Extent 	<ul style="list-style-type: none"> Extent 	<ul style="list-style-type: none"> Block 또는 페이지가 연속적으로 할당된 공간입니다. 세그먼트 또는 오브젝트 크기 증가시 Extent 크기로 연속적인 공간이 할당됩니다.
<ul style="list-style-type: none"> Block (2k, 4k, 8k, 16k, 32k) 	<ul style="list-style-type: none"> 페이지 (4k, 8k, 16k, 32k) 	<ul style="list-style-type: none"> 가장 작은 저장 공간 단위입니다. 오라클의 경우 한 행이 한 Block에 저장되지 않을 경우 체인현상과 Update시 PCT_FREE 용량 초과로 마이그레이션이 발생되어 성능이 저하되며 디스크 깨짐 현상이 간혹 발생합니다. DB2의 경우 성능 저하를 야기시키는 체인현상 및 마이그레이션 현상을 막기 위해 한 행은 반드시 한 페이지에 저장이 되도록 설계되었습니다.
<ul style="list-style-type: none"> Datafile 	<ul style="list-style-type: none"> Container 	<ul style="list-style-type: none"> 테이블 스페이스를 구성하는 물리적 파일입니다 .

Point  데이터베이스의 데이터는 물리적 공간에 저장됩니다. 오라클에서는 이러한 물리적 저장 공간을 데이터 파일이라 하며 DB2에서는 컨테이너라 합니다. 테이블 스페이스는 이러한 물리적 저장공간의 집합에 대한 논리적 개념입니다.

	오라클	DB2
정의	<ul style="list-style-type: none"> Datafile의 집합입니다. 	<ul style="list-style-type: none"> Container의 집합입니다.
테이블 스페이스 관리 방식	<ul style="list-style-type: none"> Locally managed Tablespace <ul style="list-style-type: none"> 비트맵을 사용하여 Extent를 관리합니다. Free extent가 테이블 스페이스안에서 관리됩니다. Data dictionary managed Tablespace <ul style="list-style-type: none"> 8.15 이전에 사용했던 관리방식으로 테이블 스페이스 정보가 Dictionary 테이블에 저장되어 관리됩니다. 테이블 스페이스 관리시 자원의 경합이 많이 발생되어 성능 저하를 초래합니다. 	<ul style="list-style-type: none"> System managed Space (SMS) <ul style="list-style-type: none"> 운영체제가 테이블 스페이스를 관리하는 방식입니다. 컨테이너로 디렉토리가 사용되며 운영체제의 파일시스템이 스토리지 할당 및 관리를 제어합니다. Database Managed Space (DMS) <ul style="list-style-type: none"> 데이터베이스가 테이블 스페이스를 관리하는 방식입니다. Container로 file과 raw device가 이용되어집니다. 오라클의 Locally Managed Tablespace와 유사합니다. Automatic storage table spaces <ul style="list-style-type: none"> DB2가 자동으로 스토리지를 관리하는 방식으로 테이블 스페이스의 타입에 따라 SMS 또는 DMS로 자동 생성하며 컨테이너를 지정할 필요가 없습니다
테이블 스페이스 타입	<ul style="list-style-type: none"> Temporary <ul style="list-style-type: none"> 데이터 정렬 및 Join과 같은 SQL문을 처리할때, 임시 테이블 또는 인덱스를 저장하는데 사용되며 데이터 정렬 및 Join 테이블 스페이스입니다. 또한 데이터 정렬을 위해 사용되며 사용자 당 하나의 Temporary 테이블 스페이스가 지정되어야 합니다. Permanent <ul style="list-style-type: none"> 오브젝트 및 인덱스의 모든 데이터를 저장하는 공간으로 테이블 스페이스 생성시 타입을 지정하지 않을 경우 디폴트로 생성되는 타입입니다. Undo <ul style="list-style-type: none"> UNDO 정보를 저장하기 위해 데이터베이스가 사용하는 UNDO 테이블 스페이스입니다. 	<ul style="list-style-type: none"> Regular <ul style="list-style-type: none"> 오브젝트 및 인덱스의 모든 데이터를 저장하는 공간입니다. Temporary <ul style="list-style-type: none"> 시스템 Temporary <ul style="list-style-type: none"> 정렬 및 join 같은 SQL문 및 기타 작업 실행시 데이터베이스 Manager가 사용하는 영역입니다. User Temporary <ul style="list-style-type: none"> 세션의 전역 임시 테이블의 데이터를 임시로 저장하는 영역입니다. Large <ul style="list-style-type: none"> Regular처럼 모든 데이터를 저장하나 시스템 관리 유형이 DMS일 경우에만 생성이 가능하며 한 페이지에 저장되는 행의 수가 255개를 초과하는 테이블을 저장할 수 있습니다. <p>참고) DB2는 UNDO 정보를 로그 버퍼 또는 로그 파일에 저장하므로 UNDO 정보를 위한 별도의 UNDO 테이블 스페이스는 존재하지 않습니다.</p>
디폴트로 생성되는 테이블 스페이스	<ul style="list-style-type: none"> SYSTEM <ul style="list-style-type: none"> Data Dictionary 및 데이터베이스를 관리하기 위한 관리 테이블이 저장됩니다. TEMP <ul style="list-style-type: none"> 디폴트 Temporary 테이블 스페이스입니다. UNDOTBS1 <ul style="list-style-type: none"> UNDO 정보를 저장하기 위한 디폴트 테이블 스페이스입니다. USERS <ul style="list-style-type: none"> SYS, SYSTEM 이외의 사용자가 생성한 오브젝트 및 데이터를 저장하는 디폴트 테이블 스페이스입니다. EXAMPLE <ul style="list-style-type: none"> 데이터베이스를 생성할때 생성되는 샘플 스키마내의 오브젝트들을 저장하는 테이블 스페이스입니다. 	<ul style="list-style-type: none"> SYSCATSPACE <ul style="list-style-type: none"> Catalog Table 및 데이터베이스를 관리하기 위한 관리 테이블들이 저장됩니다. TEMPSPACE1 <ul style="list-style-type: none"> 디폴트 Temporary 테이블 스페이스입니다. USERSPACE1 <ul style="list-style-type: none"> 사용자가 생성한 오브젝트 및 데이터를 저장하는 디폴트 테이블 스페이스입니다. 테이블 생성시 테이블 스페이스를 지정하지 않을 경우 디폴트로 이 테이블 스페이스를 사용합니다

Point  테이블 스페이스 생성, 변경 및 삭제, 그리고 테이블 스페이스의 정보를 조회하는 명령어를 비교한 표입니다.

테이블 스페이스 명령어 비교

	오라클	DB2
테이블 스페이스	<ul style="list-style-type: none"> CREATE TABLESPACE문을 이용하여 테이블 스페이스를 생성합니다. ALTER TABLESPACE문을 이용하여 다음과 같은 테이블 스페이스의 속성을 변경합니다. <ul style="list-style-type: none"> 데이터파일 추가 / 크기 재조정 / 삭제 가능. 테이블 스페이스 상태 변경. 로깅 및 Threshold 변경. . 	<ul style="list-style-type: none"> CREATE TABLESPACE문을 이용하여 테이블 스페이스를 생성합니다. 다음과 같은 테이블 스페이스의 속성을 변경합니다. <ul style="list-style-type: none"> DMS 테이블 스페이스의 경우 컨테이너 추가 / 크기 재조정 / 삭제 가능. PREFETCHSIZE, 버퍼풀, 오버헤드, TRANSFERRATE 설정 값 조정 가능. 테이블 스페이스의 파일 시스템 캐싱에 대한 설정값 조정 가능
테이블 스페이스 정보조회	<ul style="list-style-type: none"> v\$TABLESPACE인 다이내믹 뷰와 DBA_TABLESPACES 덱서너리 테이블을 조회하여 확인할 수 있습니다. 	<ul style="list-style-type: none"> SYSIBMADM.TBSP_UTILIZATION 카탈로그 테이블을 조회하면 테이블 스페이스 정보를 확인할 수 있습니다.

예문

오라클

테이블 스페이스 생성 :

```
SQL> CREATE TABLESPACE amj_ts
      DATAFILE '/tbs/amj_ts1.dbf' SIZE 10M;
```

테이블 스페이스 변경 :

```
SQL> ALTER TABLESPACE amj_ts
      add DATAFILE '/tbs/amj_ts2.dbf' SIZE 10M;
```

테이블 스페이스 제거 :

```
SQL> DROP TABLESPACE amj_ts;
```

테이블 스페이스 정보 조회 :

```
SQL> SELECT ts#,name FROM v$TABLESPACE;
SQL>SELECT tablespace_name,
      extent_management,
      status, contents, block_size,
      allocation_type,
      segment_space_management
      FROM DBA_TABLESPACES;
```

데이터 파일 정보 조회 :

```
SQL> SELECT file#,status,block_size
      FROM v$DATAFILE;
SQL> SELECT tablespace_name,
      file_name,file_id, status,increment_by
      FROM dba_data_files
      WHERE tablespace_name =<테이블 스페이스 명>;
```

DB2

테이블 스페이스 생성 :

```
db2 => CREATE TABLESPACE amj_ts MANAGED BY
      DATABASE USING (FILE '/tbs/amj_ts1.dbf' 300M)
```

테이블 스페이스 변경 :

```
db2 => ALTER TABLESPACE amj_ts
      ADD ( file '/tbs/amj_ts2.dbf' 300M)
```

테이블 스페이스 제거 :

```
db2 => DROP TABLESPACE amj_ts
```

테이블 스페이스 정보 조회 :

```
db2 => list tablespaces [ show detail ]
db2 => SELECT tbsp_id as id,
      substr(tbsp_name,1,20), tbsp_type,
      tbsp_content_type, tbsp_state
      FROM SYSIBMADM.TBSP_UTILIZATION
```

컨테이너 정보 조회 :

```
db2 => list tablespace containers for <Tablespace
      ID>
      [ show detail ]
db2 => SELECT substr(tbsp_name,1,20),
      int(tbsp_id) as tid,
      substr(container_name,1,25) as cont_name,
      int(container_id) as cid, container_type,
      int(total_pages) as tot_pages
      FROM SYSIBMADM.CONTAINER_UTILIZATION
      WHERE tbsp_id = <테이블 스페이스 ID>
```

Point DB2 테이블 스페이스 관리방식은 DMS와 SMS가 있습니다. 이곳에서는 두 관리방식의 차이점 및 장점에 대해 살펴보도록 합니다.

DMS Tablespaces

```
CREATE TABLESPACE dms PAGESIZE 8K
MANAGED BY DATABASE
USING ( file '/db2/tbs/dms01' 10M,
        file '/db2/tbs/dms02' 10M)
EXTENTSIZE 16
```

물리적 컨테이너의 위치 (파일 및 RAW DEVICE) :

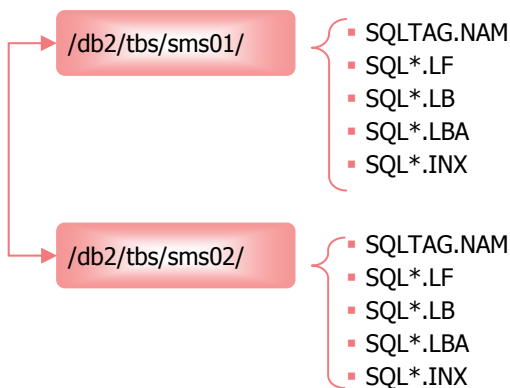


- dms01 및 dms02는 파일이며 데이터 정보 및 제어 정보가 모두 파일안에 포함됨.

SMS Tablespaces

```
CREATE TABLESPACE sms PAGESIZE 8K
MANAGED BY SYSTEM
USING( '/db2/tbs/sms01',
        '/db2/tbs/sms02')
EXTENTSIZE 16
```

물리적 컨테이너의 위치 (디렉토리) :



- SQLTAG.NAM : 이 디렉토리가 사용중임을 표시하는 파일.
- SQL*.LF : LONG VARCHAR, LONG VARGRAPHIC 데이터가 포함됨.
- SQL*.LB : BLOB, CLOB, 또는 DBCLOB 데이터가 포함됨.
- SQL*.LBA : SQL*.LB 파일의 할당 정보 및 Free 정보가 포함됨.
- SQL*.INX : 인덱스 테이블 데이터가 포함됨.

	DMS	SMS
제어	<ul style="list-style-type: none"> ▪ 인스턴스가 제어합니다. 	<ul style="list-style-type: none"> ▪ 운영체제가 제어합니다.
컨테이너	<ul style="list-style-type: none"> ▪ 파일 혹은 Row device 	<ul style="list-style-type: none"> ▪ 디렉토리
공간할당	<ul style="list-style-type: none"> ▪ 테이블 스페이스 생성시 크기를 지정할 수 있으며 AUTOMATIC STORAGE 지정시 자동으로 크기가 증가됩니다. ▪ 공간이 미리 할당되며 하나의 Extent안에 있는 공간은 물리적으로 연속적입니다. 	<ul style="list-style-type: none"> ▪ 테이블 스페이스 생성시 크기를 지정할 수 없으며 크기는 운영시스템의 파일 시스템 용량에 좌우됩니다. ▪ 추가 공간이 요구되었을 때마다 한 Page씩 할당됩니다.
테이블 위치	<ul style="list-style-type: none"> ▪ 하나의 테이블을 LOB 데이터와 일반데이터로 테이블 스페이스를 분리하여 저장할 수 있으며 인덱스도 분리하여 별도의 테이블 스페이스에 저장할 수 있습니다. 	<ul style="list-style-type: none"> ▪ 한 테이블은 한 테이블스페이스내에 존재합니다.
장점	<ul style="list-style-type: none"> ▪ 테이블이 저장되는 데이터 타입에 따라 여러 개의 테이블 스페이스로 저장가능하므로 성능이 우수합니다. ▪ DMS 테이블 스페이스는 SMS 방식보다 성능이 우수하므로 TEMPORARY를 제외한 대부분의 경우에 주로 사용됩니다. 	<ul style="list-style-type: none"> ▪ 공간이 필요할 때에만 한 페이지씩 할당되므로 사용하지 않는 공간을 미리 할당해 놓지 않습니다. ▪ 테이블 스페이스 생성시 컨테이너를 사전 정의할 필요가 없으므로 초기 작업이 줄어듭니다. ▪ TEMPORARY 테이블 스페이스 또는 소형 테이블을 저장하는 데에 주로 사용됩니다.

Point DB2의 데이터 입출력의 기본 단위인 페이지와 데이터를 저장할때 할당되는 기본 단위인 Extent 를 알아보도록 합니다.

페이지 정의

- 버퍼풀 및 테이블 스페이스의 입출력 단위로 한 행은 여러 개의 페이지에 나누어 저장될 수 없습니다. 이는 오라클의 체인 또는 마이그레이션으로 인한 성능 저하 현상을 사전에 방지하기 위한 조치입니다.
- 한 행은 여러 개의 페이지로 나누어 저장될 수 없습니다.
- Random 하게 데이터를 읽고 쓰는 OLTP 응용프로그램의 경우 PAGESIZE를 작게 하여 버퍼풀에 원치않는 데이터를 올려놓지 않도록 하는 것이 성능 향상에 도움이 됩니다.
- 한번에 연속적인 많은 행을 접근하는 DSS 응용프로그램의 경우 I/O 요청을 줄이기 위해 보다 큰 PAGESIZE가 유리합니다. 하지만 만약 한 행의 길이가 PAGESIZE/255 보다 작을 경우에는 좀 더 작은 PAGESIZE가 적합합니다.

테이블 제한 사항

테이블 제한 /Page Size	4K	8K	16K	32K
컬럼의 최대수	500개	1012개	1012개	1012개
행의 최대 길이	4005 바이트	8101 바이트	16293 바이트	32677 바이트
테이블의 최대 크기	64GB (Regular 경우) 2048GB (Large 경우)	128GB(Regular 경우) 4096GB (Large 경우)	256GB(Regular 경우) 8192GB (Large 경우)	512GB(Regular 경우) 16384GB (Large 경우)


EXTENT 정의

- 데이터베이스 관리 프로그램이 라운드 로빈 방식으로 데이터를 골고루 분배하기 위해 테이블 스페이스내의 모든 Container들에게 돌아가면서 EXTENT 크기만큼 데이터를 입력합니다.
- 데이터베이스 생성시 DFT_EXTENT_SZ 파라미터 또는 테이블 스페이스 생성시 EXTENTSIZE 에 값을 지정하여 Extent 크기를 지정할 수 있습니다.

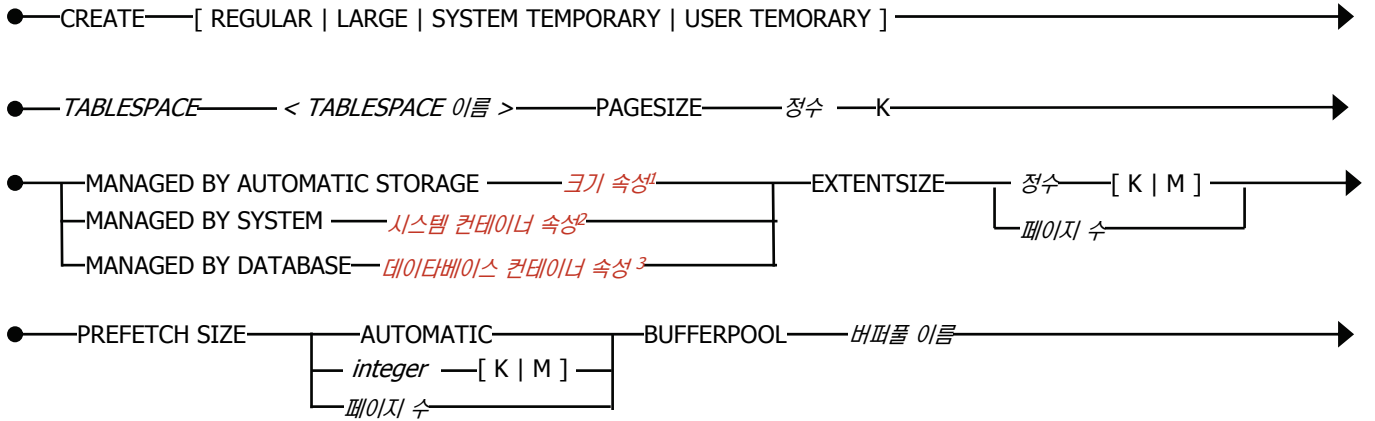
프리페치 정의

- 프리페치는 프리페처 프로세스가 응용프로그램이 데이터를 요구하기 전에 미리 디스크의 데이터를 검색한 후 데이터를 버퍼풀로 옮겨 놓는 방식입니다.
- 데이터 베이스 구성 매개변수인 num_ioservers를 통해 충분한 입출력 서버를 구성할 경우 데이터의 프리페치를 사용할 수 있습니다. 또한 이를 통하여 쿼리의 성능을 크게 향상 시킬 수 있습니다.
- 프리페치 크기는 프리페처 프로세스가 한번에 읽을 페이지 수입니다. 데이터베이스 생성시 dft_prefetch_sz 옵션을 지정하거나 테이블 생성시 prefetchsize 옵션을 지정하면 프리페치 크기를 지정할 수 있습니다.
- 프리페치 크기는 EXTENT 크기의 배수로 지정하는 것이 가장 성능이 좋습니다.

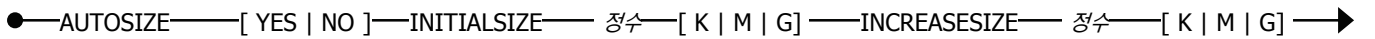
06 명령어를 이용한 테이블 스페이스 생성

Point  명령어를 이용하여 DB2 테이블 스페이스를 생성합니다.

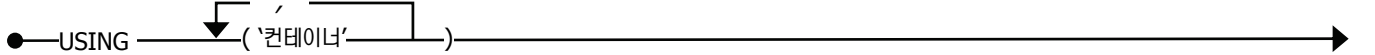
Create Tablespace 구문



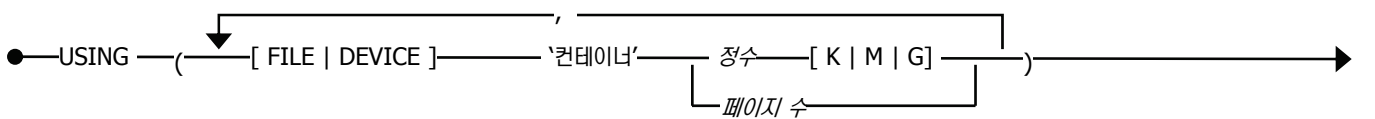
크기 속성¹:



시스템 컨테이너 속성²:



데이터베이스 컨테이너 속성³:



참고 : 이외에도 더 많은 옵션이 있으므로 도움말을 참고하시기 바랍니다.


예제

```
db2=> CREATE TEMPORARY TABLESPACE TEMPSPACE2 MANAGED BY DATABASE
      USING ('d:\wacc_tbsp', 'e:\wacc_tbsp') EXTENTSIZE 64 PREFETCHSIZE 32

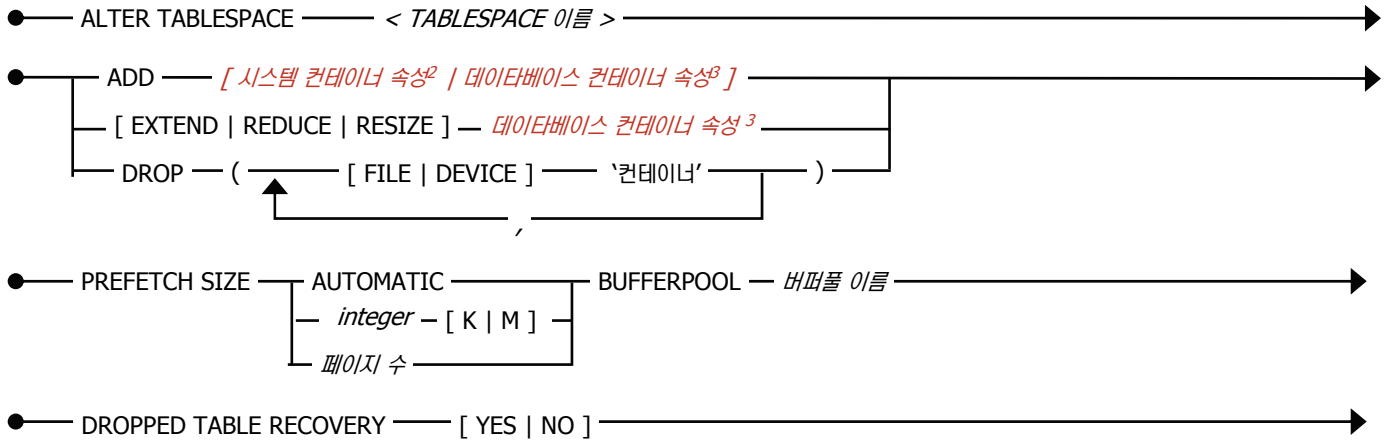
db2=> CREATE TABLESPACE DATATS MANAGED BY AUTOMATIC STORAGE

db2=> CREATE LARGE TABLESPACE LARGEDATA INCREASESIZE 10 PERCENT MAXSIZE 512 M

db2=> CREATE TABLESPACE ACCOUNTING MANAGED BY DATABASE
      USING (FILE '\tbs\acc_1' 50000, FILE '\tbs\acc_2' 50000) EXTENTSIZE 256
```

Point  명령어를 이용하여 DB2 테이블 스페이스를 변경 및 삭제합니다.

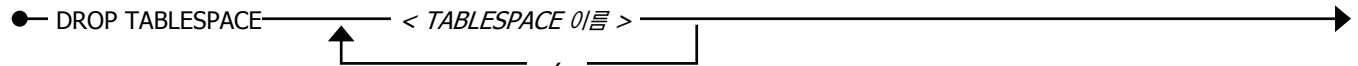
Alter Tablespace 구문



참고) 시스템 컨테이너 속성 및 데이터베이스 컨테이너 속성은 앞장과 동일하므로 생략합니다.

이외에도 더 많은 옵션이 있으므로 도움말을 참고하시기 바랍니다.

Drop Tablespace 구문




예제

```
db2=> ALTER TABLESPACE PAYROLL ADD (DEVICE '/dev/rhdisk9' 10000)
```

```
db2=> ALTER TABLESPACE DATA_TS EXTEND (ALL 1000)
```

```
db2=> ALTER TABLESPACE TS0 ADD (FILE 'cont2' 2000, FILE 'cont3' 2000) ADD (FILE 'cont4' 2000)
      EXTEND (FILE 'cont0' 100) RESIZE (FILE 'cont1' 3000)
```

```
db2=> DROP TABLESPACE TS0
```


Point  명령어를 이용하여 DB2 테이블 스페이스 정보를 조회할 수 있습니다.

list Tablespace 구문

- LIST TABLESPACES [SHOW DETAIL] →
- LIST TABLESPACE CONTAINERS for <TABLESPACE ID> [SHOW DETAIL] →

예제

db2=> list tablespaces show detail
현재 데이터베이스에 대한 테이블 스페이스

테이블 스페이스 ID	= 0
이름	= SYSCATSPACE
유형	= 데이터베이스 관리 스페이스
내용	= 모든 영구 데이터. 일반 테이블 스페이스
상태	= 0x0000

세부사항 설명:

정상	
전체 페이지 수	= 12288
사용 가능한 페이지 수	= 12284
사용된 페이지 수	= 8564
사용 가능한 페이지 수	= 3720
상위 워터 마크(페이지)	= 8564
페이지 크기(바이트)	= 8192
Extent 크기(페이지)	= 4
프리페치 크기(페이지)	= 4
컨테이너 수	= 1



db2=> list tablespace containers for 0 show detail

테이블 스페이스 0에 대한 테이블 스페이스 컨테이너

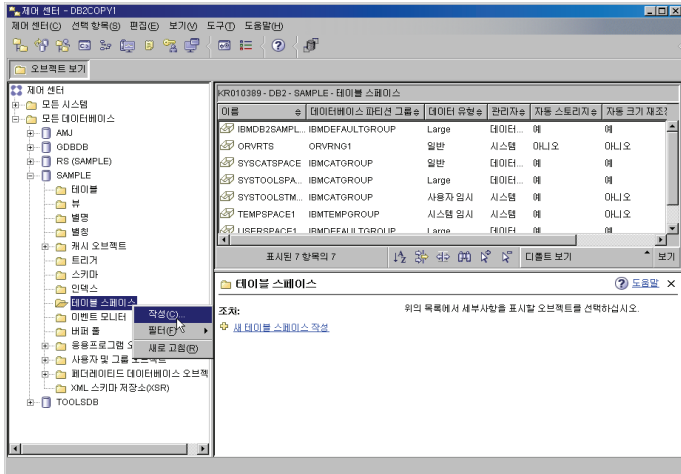
컨테이너 ID	= 0
이름	= C:WDB2\WNODE0000\SAMPLEWT0000000\WC0000000.CAT
유형	= 파일
전체 페이지 수	= 12288
사용 가능한 페이지 수	= 12284
액세스 가능	= 예

09 제어센타를 이용한 테이블 스페이스 생성

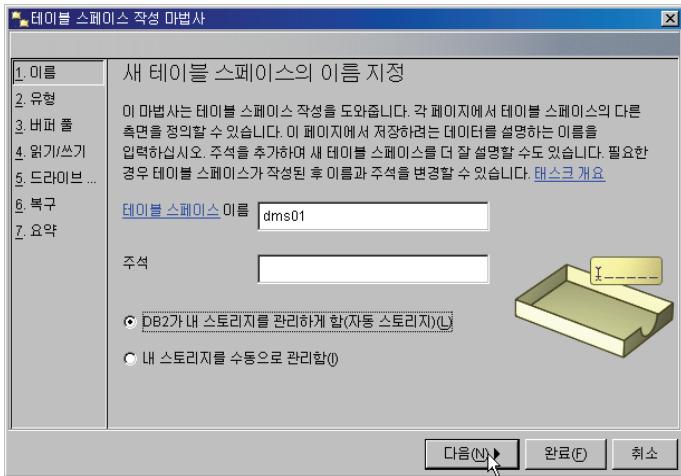
테이블 스페이스 비교

Point 제어센타를 이용하여 테이블 스페이스를 생성합니다.

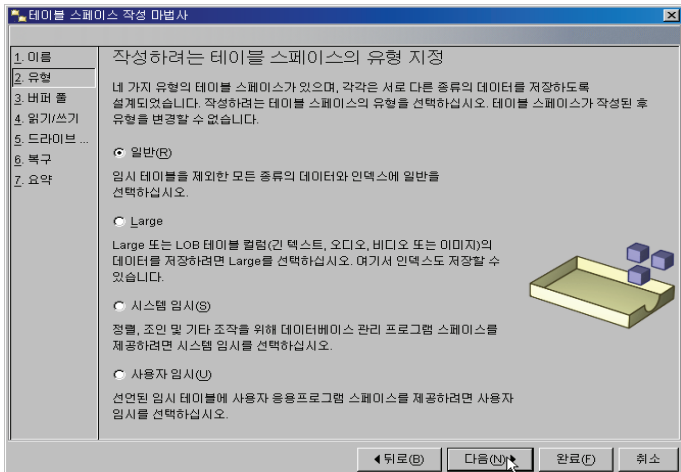
- 제어센타를 실행하는 명령어인 "db2cc" 명령어를 명령창에서 실행한 후 아래와 같은 화면이 나타나면 "모든 데이터베이스 => Sample => 테이블 스페이스"를 선택한 후 오른쪽 버튼을 눌러 작성을 클릭합니다.



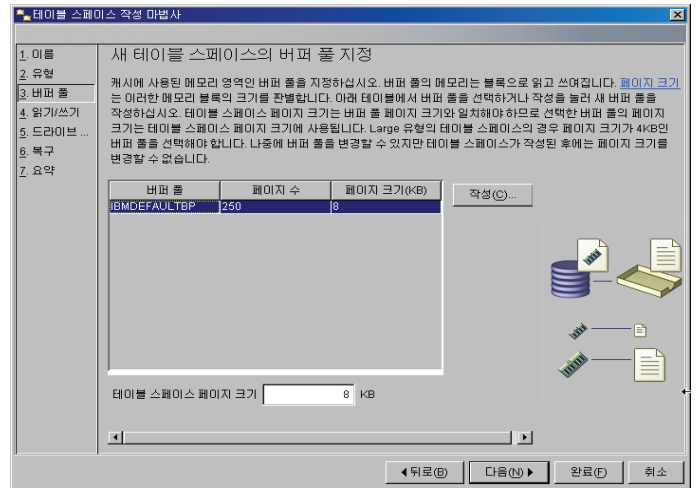
- 테이블 스페이스의 이름을 지정한 후 스토리지를 자동으로 관리할 것인지 수동으로 관리할 것인지 지정합니다.



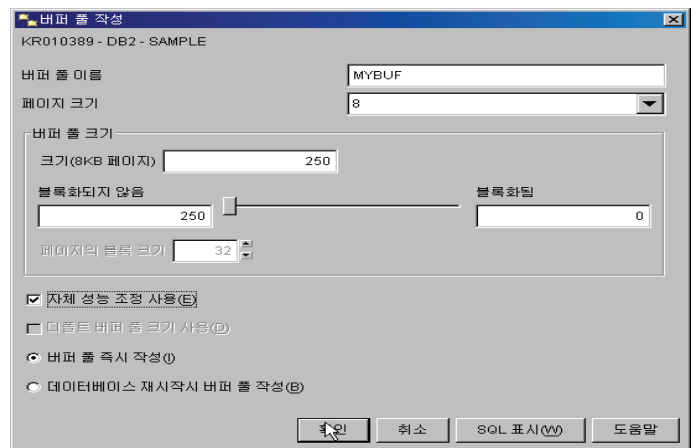
- 테이블 스페이스에 저장할 데이터에 따라 테이블 스페이스의 유형이 달라집니다. 일반 데이터를 저장할 경우 "일반"을 선택합니다.



- 앞에서 설명드린 것처럼 테이블 스페이스별로 버퍼풀을 지정할 수 있습니다. 현재 데이터베이스내에 존재하는 기존의 버퍼풀을 지정할 수도 있으며 새로 작성하는 테이블 스페이스의 전용 버퍼풀을 새로 작성할 수도 있습니다. 이곳에서는 "작성" 버튼을 눌러 새로 작성하는 테이블 스페이스를 위한 전용 버퍼풀을 작성하도록 합니다.



- 새 테이블 스페이스를 위한 버퍼풀을 작성합니다. 버퍼풀 이름을 지정한 후 새로 작성하는 테이블 스페이스의 페이지 크기와 동일한 크기의 페이지 크기를 지정합니다. "자동 성능 조정 사용"을 클릭하여 데이터베이스 Manager가 워크로드에 따라 메모리 크기를 자동으로 조정하도록 합니다. 마지막으로 버퍼풀 생성시점을 바로 즉시로 지정합니다.

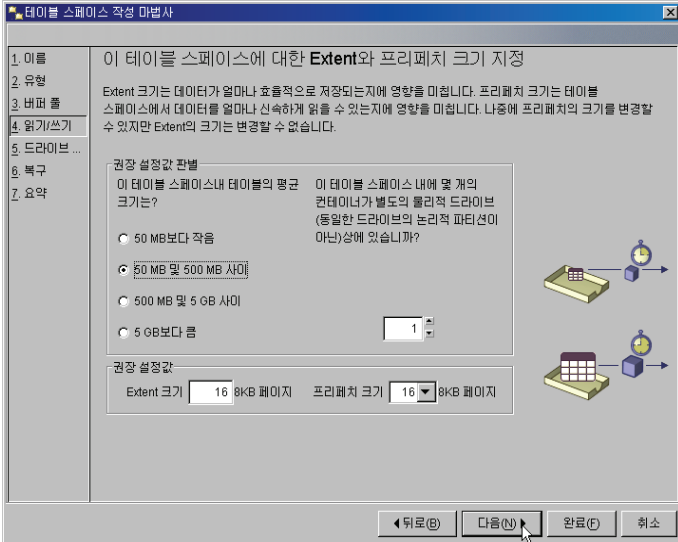


10 제어센타를 이용한 테이블 스페이스 생성

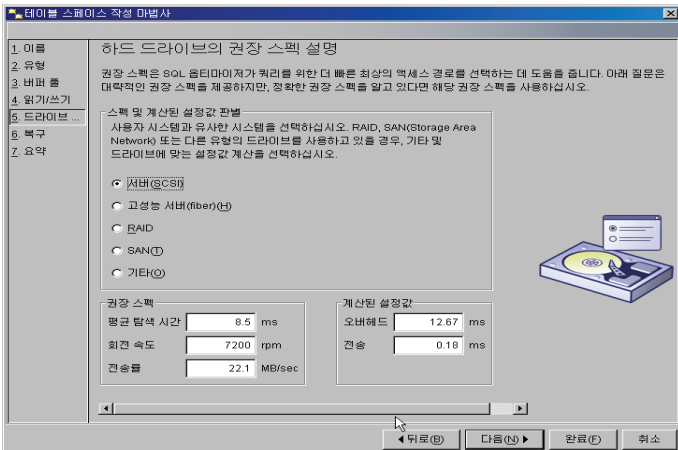
테이블 스페이스 비교

Point 제어센타를 이용하여 테이블 스페이스를 생성합니다.

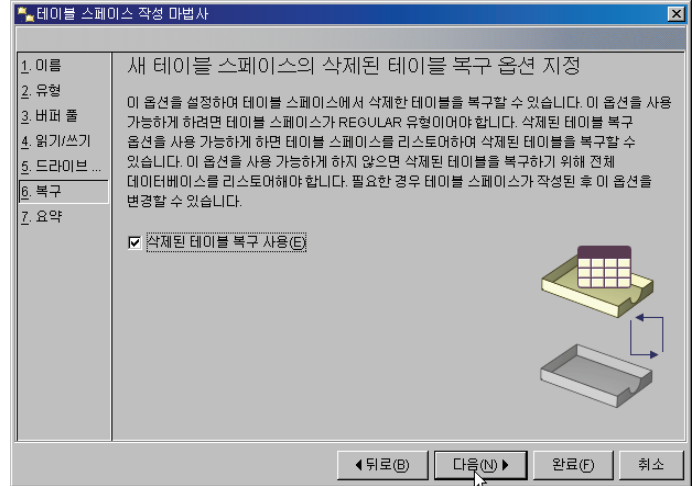
- 데이터를 효율적으로 저장 및 읽기 위해서는 적절한 Extent 크기가 지정되어야 합니다. 새 테이블 스페이스에 저장될 테이블들의 평균 크기를 지정하면 Extent 및 프리패치 크기의 권장값들이 나타납니다.



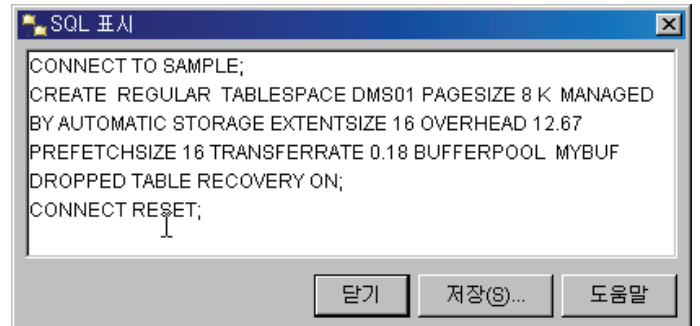
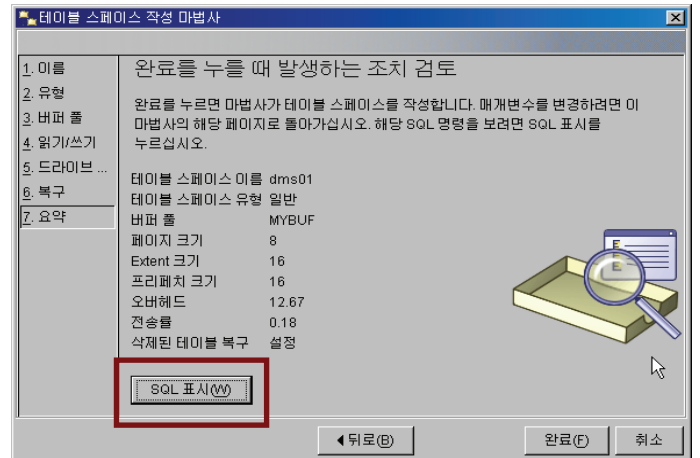
- 스토리지의 정확한 스펙을 지정하면 SQL 옵티마이저가 성능을 향상시키기 위해 액세스 경로를 선택하는데 도움을 줍니다. 정확한 스펙을 알고 있을 경우 입력하고 알지 못할 경우에는 스토리지 유형을 선택하면 권장 스펙이 나타나므로 그 스펙을 사용하도록 합니다.



- “삭제된 테이블 복구 사용” 옵션을 선택하면 테이블 스페이스에서 삭제된 테이블을 복구할 수 있습니다.




- 이제까지 설정한 값을 확인한 후 SQL 표시 버튼을 누르면 SQL 명령문을 확인할 수 있습니다. 완료버튼을 누릅니다.

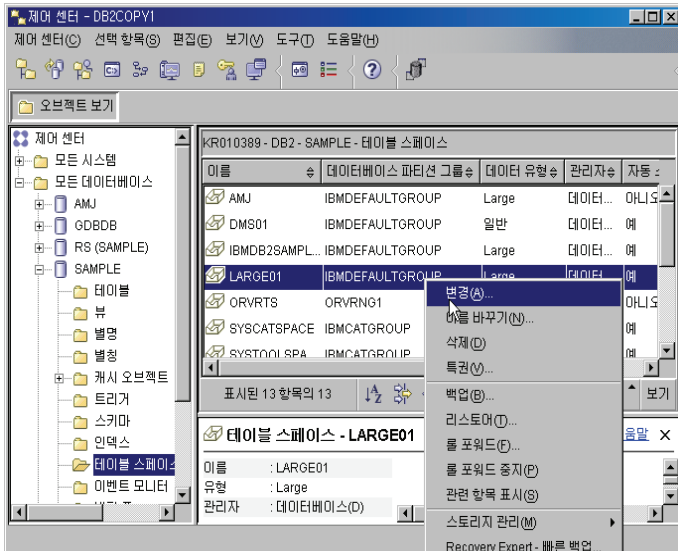


11 제어센타를 이용한 테이블 스페이스 변경

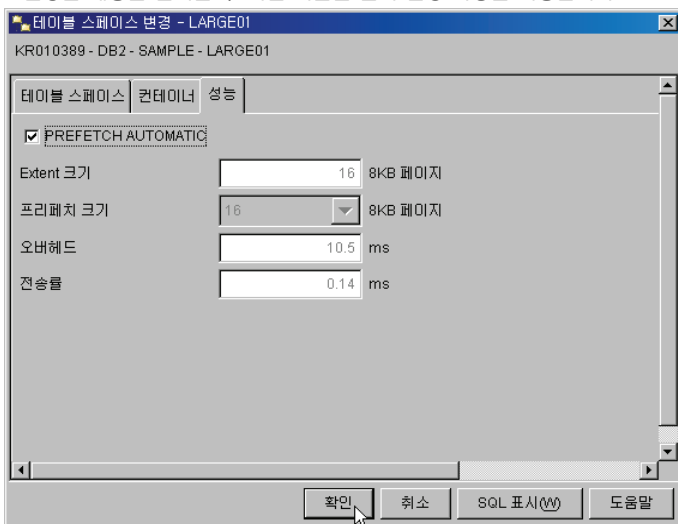
테이블 스페이스 비교

Point  제어센타를 이용하여 테이블 스페이스를 생성합니다.

1. 제어센타에서 "테이블 스페이스"를 선택한 후 오른쪽 패널에서 앞에서 생성한 dms01 테이블 스페이스를 선택합니다. 다시 오른쪽 버튼을 클릭한 후 변경을 선택합니다.



2. 변경할 내용을 선택한 후 확인 버튼을 눌러 변경 사항을 적용합니다.



테이블 스페이스 관련 Q/A

- 아래와 같은 명령어를 이용하여 테이블 스페이스를 생성하려 하는데 에러가 발생합니다. 에러 코드는 SQL0970N이며 "system attempted to write to a read-only file"이라고 나옵니다. 어떻게 조치해야 하는지요?

현재 테이블 스페이스의 컨테이너로 사용될 파일 또는 디렉토리에 테이블 스페이스를 생성하는 사용자에게 권한이 없어 발생하는 에러입니다. OS 시스템에서 사용자에게 파일 또는 디렉토리를 사용할 권한을 부여하면 해결가능 합니다.
- 테이블 스페이스를 생성하는데 버퍼풀 에러가 발생합니다. 어떻게 해결해야 되는건지요 ?

테이블 스페이스의 페이지 SIZE가 얼마인지요? 만약 8K 로 지정하였을 경우 동일한 페이지 크기를 갖는 버퍼풀을 생성해야 합니다. 우선 동일한 페이지 크기를 갖는 버퍼풀을 생성한 후 다시 테이블 스페이스를 생성하면 됩니다.
- 오라클은 사용자마다 테이블 스페이스를 지정할 수 있습니다. DB2는 이럴 경우 어떻게 해야 되는 건지요?

DB2는 오라클처럼 데이터베이스 내부에서 별도의 사용자를 생성하지 않고 OS의 사용자를 이용합니다. 이런 이유로 사용자마다 테이블 스페이스를 지정하지 않습니다. 다만 특정 테이블 스페이스에 대한 권한을 특정 사용자에게만 부여함으로써 그 사용자만 그 테이블 스페이스를 사용하도록 지정할 수 있습니다. 이렇게 할 경우 오라클과 유사하게 사용할 수 있습니다.
- 제가 테이블을 INT, VARCHAR(100), VARCHAR(4000) 컬럼이 포함된 테이블을 생성하려 하는데 다음과 같은 에러가 발생합니다. ERROR [42727] [IBM][DB2/LINUX] SQL0286N A default table space could not be found with a page size of at least "8192" that authorization ID "DB2" is authorized to use. SQLSTATE=42727 어떻게 해야 되는 건지요?

DB2에서는 오라클의 체인 또는 마이그레이션 현상을 사전에 방지하기 위해 한 행의 크기가 페이지 사이즈보다 클 수 없게 되어 있습니다.

위 에러는 테이블 컬럼의 총 크기가 4K 페이지보다 크고 또한 페이지 크기가 8K 인 테이블 스페이스가 없거나 사용 권한이 없어서 발생합니다. 먼저 8K인 테이블 스페이스를 생성하거나 이미 존재한다면 테이블을 생성하는 사용자에게 그 테이블 스페이스를 사용할 권한을 부여하시면 됩니다. 이때 주의하실 점은 8K 버퍼풀도 반드시 존재해야 합니다. 만약 존재하지 않을 경우 비록 8K 페이지 크기를 갖는 테이블 스페이스를 생성하였다 하더라도 에러가 발생합니다. 참고로 테이블 컬럼의 총 크기를 계산할 때 BLOB, CLOB과 같은 LOB형은 제외됩니다. 이 데이터 타입은 따로 저장되기 때문에 계산식에서 제외합니다.



Unit 08 **오브젝트 비교**

오라클 사용자를 위한 DB2 가이드

스키마, 테이블 및 뷰 비교

인덱스 비교

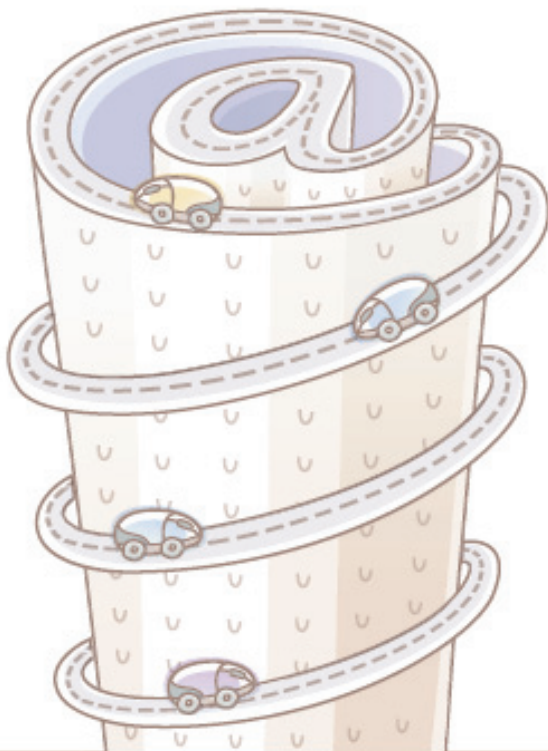
시퀀스 비교

트리거 비교

함수 비교

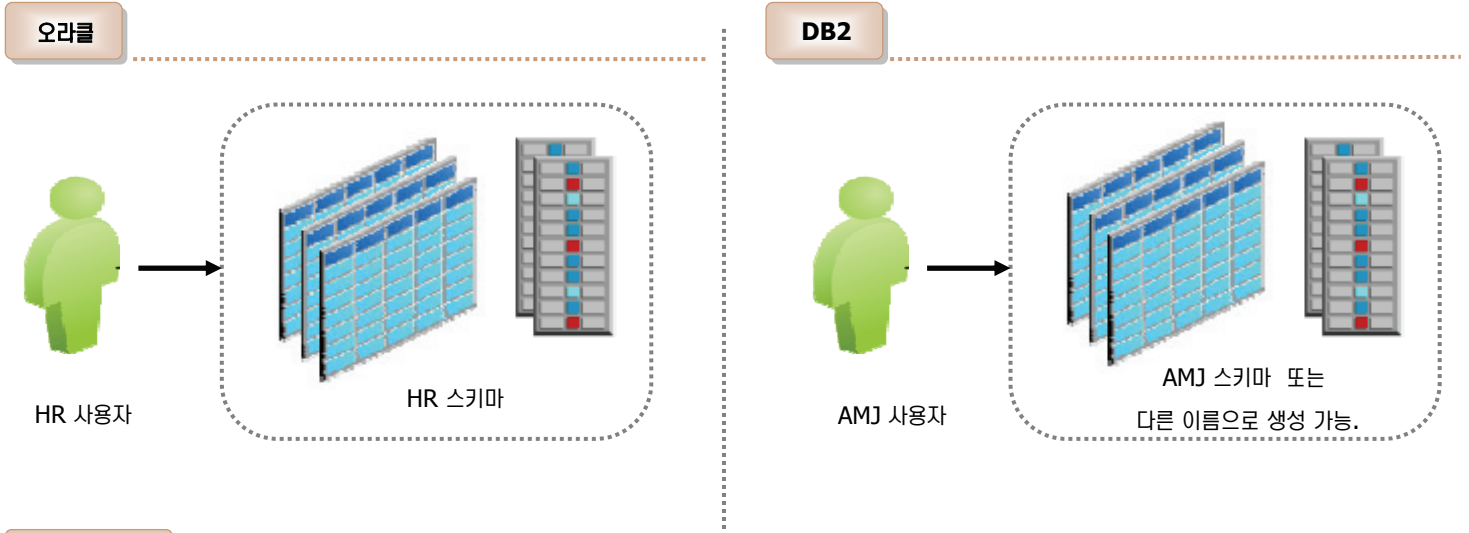
DB2 9.5에서 제공하는 오라클 기능

PL/SQL과 SQLPL 비교



01 스키마 비교


Point DB2와 오라클에서 스키마는 특정 사용자가 소유하는 오브젝트들의 집합입니다. 스키마 오브젝트에는 테이블, 뷰, 인덱스 및 함수, 스토어드 프로시저 등 있습니다. 오라클에서 스키마는 접속한 사용자와 동일한 이름으로 생성해야 합니다. 그러나 DB2에서는 일반적으로 접속한 사용자와 동일한 이름으로 생성되지만 다른 스키마 이름으로도 생성이 가능합니다.



스키마 비교

	오라클	DB2
스키마 설명	<ul style="list-style-type: none"> ▪ 사용자를 생성할 경우 동일한 이름을 가진 스키마가 생성되며 그 스키마의 소유자는 사용자가 됩니다. 	<ul style="list-style-type: none"> ▪ 스키마 이름이 사용자 이름과 동일할 필요는 없습니다. ▪ 스키마를 생성할 경우 그 스키마를 생성한 사용자가 그 스키마의 소유자가 됩니다.
기본 스키마	<ul style="list-style-type: none"> ▪ 데이터베이스 생성시 다음과 같은 기본 스키마가 제공됩니다. <ul style="list-style-type: none"> ▪ SYS: 디렉터리리에 대한 모든 기본 테이블과 뷰의 스키마 ▪ SYSTEM: 관리정보를 보여주는데 필요한 테이블과 뷰의 스키마 ▪ SAMPLE 스키마 (HR, OE, PM, SH)가 존재 	<ul style="list-style-type: none"> ▪ 데이터베이스 생성시 다음과 같은 기본 스키마가 제공됩니다. <ul style="list-style-type: none"> ▪ SYSIBM: 시스템 카탈로그 테이블의 스키마 ▪ SYSCAT: 시스템 카탈로그 뷰의 스키마 ▪ SYSSTAT: 통계자료와 관련된 시스템 카탈로그 뷰의 스키마 ▪ SYSFUN: 기본적으로 제공되는 사용자 정의 함수의 스키마
스키마 생성 구문	<ul style="list-style-type: none"> ▪ CREATE SCHEMA AUTHORIZATION <사용자 명> CREATE TABLE ~ ▪ 옵션 설명 : <ul style="list-style-type: none"> ▪ <사용자 명> : 스키마 명과 사용자 명이 동일하며 스키마에 대한 권한이 사용자에게 부여됩니다. 	<ul style="list-style-type: none"> ▪ CREATE SCHEMA <스키마 명> AUTHORIZATION <authorization_name> ~ ▪ 옵션 설명 : <ul style="list-style-type: none"> ▪ <스키마 명> : 임의의 고유한 이름으로 지정합니다. ▪ <authorization_name> : 스키마의 소유자를 지정합니다.
스키마 제거 구문	<ul style="list-style-type: none"> ▪ 스키마 제거 명령어는 존재하지 않으며 사용자를 제거할 경우 자동적으로 스키마도 제거됩니다. 	<ul style="list-style-type: none"> ▪ DROP SCHEMA <스키마 명> RESTRICT
스키마 정보 조회	<ul style="list-style-type: none"> ▪ 생성된 스키마 정보를 조회하는 dictionary 테이블이 따로 존재하지는 않습니다. 다만 DBA_TABLES 또는 ALL_TABLES 등에 OWNER 칼럼이 스키마가 됩니다. 	<ul style="list-style-type: none"> ▪ 생성된 스키마에 대한 정보는 SYSCAT.SCHEMATA 뷰를 이용하여 확인합니다.
예문	<ul style="list-style-type: none"> ▪ SQL> CREATE USER amj IDENTIFIED BY ***; ▪ SQL> CREATE SCHEMA AUTHORIZATION amj <ul style="list-style-type: none"> ▪ CREATE TABLE AMJ.T1 (C1 INT); ▪ SQL> select * from all_tables; 	<ul style="list-style-type: none"> ▪ db2=> CREATE SCHEMA amj AUTHORIZATION HR ▪ db2=> CREATE TABLE AMJ.T1 (C1 INT) ▪ db2=> list tables for schema <스키마명> ▪ db2=> select * from syscat.schemata

02 테이블 및 뷰 비교


Point  테이블은 데이터를 저장하는 논리적인 저장 장소이며 뷰는 하나 이상의 테이블로부터 논리적으로 데이터를 추출한 부분집합으로 논리적이고 가상적인 테이블입니다. 오라클과 DB2의 테이블 및 뷰는 거의 동일합니다.

테이블 비교

	오라클	DB2
테이블 설명	<ul style="list-style-type: none"> 테이블이 저장되는 테이블 스페이스를 지정하나 물리적인 저장 정보를 테이블 생성시 지정도 가능합니다. 컬럼 삭제 및 컬럼 속성을 변경할 수 있습니다. 	<ul style="list-style-type: none"> 테이블이 저장되는 테이블 스페이스를 지정함으로써 물리적인 저장 장소를 선택합니다. 오라클과는 달리 테이블마다 물리적인 저장 옵션을 지정할 수는 없습니다. 일반 데이터와 LOB 데이터, 인덱스등을 각기 다른 테이블 스페이스에 저장할 수 있습니다. 컬럼 삭제 및 컬럼 속성을 변경할 수 있습니다.
테이블 종류	<ul style="list-style-type: none"> 일반 테이블 전역 임시 테이블 파티션 테이블 인덱스 구조 테이블 	<ul style="list-style-type: none"> 일반 테이블 전역 임시 테이블 파티션 테이블 Multi Dimensional Clustering (MDC) 테이블
테이블 저장장소 지정 방법	<ul style="list-style-type: none"> 지정된 테이블 스페이스에 저장됩니다. PCTFREE 및 스토리지 옵션을 두어 테이블마다 스토리지 값을 변경할 수 있으며 옵션 생략시 지정된 테이블 스페이스의 디폴트 값을 적용합니다. 스토리지 옵션을 두는 이유는 오라클은 DB2와는 달리 체이닝과 마이그레이션이 발생되므로 이 현상을 줄이기 위함입니다. 	<ul style="list-style-type: none"> 지정된 테이블 스페이스에 저장됩니다. 별도의 스토리지 옵션은 존재하지 않으며 테이블 스페이스에서 지정된 스토리지 옵션 값을 적용합니다. 스토리지 옵션 생략시 지정된 테이블 스페이스의 디폴트 값을 적용합니다. DMS 테이블 스페이스의 경우 한 테이블의 데이터를 여러 개의 테이블 스페이스로 나누어 저장할 수 있습니다.
테이블 구문	<ul style="list-style-type: none"> 테이블 생성/변경/제거 구문은 오라클과 DB2 모두 유사합니다. <ul style="list-style-type: none"> CREATE TABLE ~ / ALTER TABLE ~ / DROP TABLE ~ 	
테이블 정보 조회	<ul style="list-style-type: none"> DBA_TABLES, DBA_TAB_COLUMNS 디셔너리 테이블을 조회하면 테이블 및 컬럼에 대한 정보를 조회할 수 있습니다. 	<ul style="list-style-type: none"> 특정 스키마에 속하는 테이블과 뷰를 확인할 경우 "list tables for schema <스키마 명>" 명령어를 실행합니다. SYSCAT.TABLES, SYSCAT.COLUMNS 카탈로그 테이블에 테이블 및 컬럼 정보가 저장되어 있습니다.
테이블 구조 정보	<ul style="list-style-type: none"> DESC [테이블 명]를 이용하여 테이블 구조를 확인합니다. 	<ul style="list-style-type: none"> DESCRIBE TABLE [테이블 명]을 이용하여 테이블 구조를 확인합니다.
Dummy 테이블	<ul style="list-style-type: none"> DUAL 테이블입니다. <ul style="list-style-type: none"> sqlplus > SELECT TO_CHAR(SYSDATE,'YYYYMMDD') FROM DUAL; 	<ul style="list-style-type: none"> SYSIBM.SYSDUMMY1 입니다. <ul style="list-style-type: none"> db2=> SELECT TO_CHAR(CURRENT TIMESTAMP , 'YYYYMMDDHH24MISS') FROM SYSIBM.SYSDUMMY1;

뷰 비교

	오라클	DB2
뷰 생성 구문	<ul style="list-style-type: none"> CREATE VIEW view_name AS SELECT 구문 	<ul style="list-style-type: none"> CREATE VIEW view_name AS SELECT 구문
뷰 변경 구문	<ul style="list-style-type: none"> ALTER VIEW view_name ~ 	<ul style="list-style-type: none"> ALTER VIEW view_name ~
뷰 제거 구문	<ul style="list-style-type: none"> DROP View view_name 	<ul style="list-style-type: none"> DROP VIEW view_name
뷰 정보 조회	<ul style="list-style-type: none"> DBA_VIEW 디셔너리 테이블을 조회합니다. 	<ul style="list-style-type: none"> SYSCAT.VIEW 카탈로그 테이블을 조회합니다.
예문	<pre>SQL> CREATE VIEW view01 AS SELECT ID,NAME FROM EMP;</pre>	<pre>db2=> CREATE VIEW view01 AS SELECT ID,NAME FROM EMP;</pre>

Point  오라클과 DB2는 모두 데이터베이스의 메타 데이터 정보를 저장하는 공간이 존재합니다. 오라클에서는 이를 데이터 디렉터리라 하고 DB2에서는 카탈로그 테이블이라 합니다. 다음은 이러한 메타 데이터를 저장하고 관리하는 방식에 대해 비교합니다.

오라클 디렉터리와 DB2 카탈로그 테이블 비교

오라클 데이터 디렉터리	DB2 카탈로그 테이블
<ul style="list-style-type: none"> • DBA_XXX: 모든 사용자의 객체에 대한 정보 제공 • USR_XXX: 자신이 소유한 오브젝트에 대한 정보 제공 • ALL_XXX: 접근 권한이 가능한 오브젝트에 대한 정보 제공 • V\$_XXX: 오라클의 성능 분석, 통계 정보 제공 	<ul style="list-style-type: none"> • SYSIBM.XXX: 데이터베이스내의 모든 객체에 대한 정보 제공. • SYSCAT.XXX: SYSIBM에 속하는 테이블 중 중요 정보만 조합하여 제공. • SYSSTAT.XXX: SYSIBM에 속하는 테이블중 통계 정보만 조합하여 제공. • SYSFUN.XXXX: 사용자 정의함수에 대한 정보 제공. • SYSIBMADM.XXXX: DB2의 성능 분석 및 관리 정보 제공
<ul style="list-style-type: none"> • 디렉터리 테이블의 종류를 확인할 경우 다음과 같은 SQL 문을 이용하여 확인이 가능합니다. SQL> SELECT * FROM DICTIONARY; 	<ul style="list-style-type: none"> • 카탈로그 테이블의 종류를 확인할 경우 다음과 같은 명령어를 이용하여 확인이 가능합니다. db2=> LIST TABLES FOR SYSTEM


테이블 비교 :

오라클 데이터 디렉터리	DB2 카탈로그 테이블
DBA_TABLES	SYSCAT.TABLES
DBA_TAB_COLUMNS	SYSCAT.COLUMNS
DBA_TABLESPACES	SYSCAT.TABLESPACES
DBA_INDEXES	SYSCAT.INDEXES
DBA_CONSTRAINTS	SYSCAT.TABCONST
DBA_CONS_COLUMNS	SYSCAT.COLCHECKS,SYSCAT.REFERENCES,SYSCAT.INDEXES
DBA_VIEWS	SYSCAT.VIEWS
DBA_SEQUENCES	SYSCAT.SEQUENCES
DBA_TRIGGERS	SYSCAT.TRIGGERS
DBA_USERS	OS의 사용자를 사용함으로 해당 정보에 대한 관리가 필요치 않습니다.
DBA_ROLES	SYSCAT.ROLES
TABLE_PRIVILEGES	SYSCAT.TABAUTH,SYSCAT.COLAUTH,SYSCAT.INDEXAUTH 등 각 OBJECT 마다 정보를 담고 있는 테이블이 다름.




오라클의 Data Dictionary 테이블과 DB2의 Catalog Table은 데이터베이스의 메타 정보를 저장합니다. 그러나 유사한 정보를 각기 다른 테이블에 저장하고 있기 때문에 1대 1로 Mapping 하기는 어렵습니다. 위의 내용은 그 중 1대 1로 Mapping 이 가능한 테이블을 비교한 표이므로 참고하시기 바라며 이외의 테이블에 대해서는 위의 "list tables for system" 명령어를 이용하여 카탈로그 테이블의 종류를 확인하시기 바랍니다.

04 테이블 데이터 타입 비교

Point  테이블 데이터 타입은 저장되는 데이터에 따라 숫자, 문자, 시간, 날짜 등 다양한 형태의 데이터가 저장됩니다. 오라클과 DB2는 데이터의 형태에 따라 타입을 분류합니다. 다음은 오라클과 DB2 데이터 타입을 비교합니다.

데이터 타입 비교

Oracle	DB2	범위
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)	<ul style="list-style-type: none"> SMALLINT INTEGER BIGINT 	<ul style="list-style-type: none"> 1 <= p <= 4일 경우 SMALLINT 5 <= p <= 9일 경우 INTEGER 10 <= p <= 18일 경우 BIGINT
NUMBER (p, s)	DECIMAL (p, s)	s > 0일 경우
NUMBER	<ul style="list-style-type: none"> FLOAT REAL DOUBLE 	
BLOB	BLOB(n)	n <= 2GB일 경우
CLOB	CLOB(n)	n <= 2GB일 경우
NCLOB	DBCLOB(n)	n <= 2GB일 경우 DBCLOB(n/2)를 사용합니다.
DATE	<ul style="list-style-type: none"> TIMESTAMP DATE(MM/DD/YYYY) TIME(HH24:MI:SS) 	<ul style="list-style-type: none"> DATE 와 TIME이 모두 있을 경우 TIMESTAMP 날짜만 표현할 경우 DATE(MM/DD/YYYY)가 됩니다. 시간만 표현할 경우 TIME(HH24:MI:SS)가 됩니다.
TIMESTAMP	TIMESTAMP	
XMLType	XML	
RAW (n)	<ul style="list-style-type: none"> CHAR(n) FOR BIT DATA VARCHAR(n) FOR BIT DATA BLOB(n) 	<ul style="list-style-type: none"> n <= 254일 경우 CHAR 254 < n <= 32672일 경우 VARCHAR 32672 < n <= 2GB일 경우 BLOB
LONG RAW	<ul style="list-style-type: none"> LONG VARCHAR(n) FOR BIT DATA BLOB(n) 	<ul style="list-style-type: none"> n <= 32700일 경우 LONG 32700 < n <= 2GB일 경우 BLOB

Point  테이블 데이터 타입은 저장되는 데이터에 따라 숫자, 문자, 시간, 날짜 등 다양한 형태의 데이터가 저장됩니다. 오라클과 DB2는 데이터의 형태에 따라 타입을 분류합니다. 다음은 오라클과 DB2 데이터 타입을 비교합니다.

제약 조건 비교

- 제약조건이란 테이블에 부적절한 자료가 입력되는 것을 방지하기 위한 여러 가지 규칙입니다. 간단하게 테이블 안에서 데이터의 성격을 정의하는 것으로 제약조건입니다.
 - 데이터의 무결성 유지를 위하여 사용자가 지정할 수 있는 성질입니다.
 - 모든 CONSTRAINT는 데이터 사전(DICTIONARY)에 저장 됩니다
 - 의미있는 이름을 부여했다면 CONSTRAINT를 쉽게 참조할 수 있습니다.
 - 제약조건은 테이블을 생성할 당시에 지정할 수도 있고, 테이블 생성 후 구조변경(ALTER)명령어를 통해서도 추가가 가능합니다.
 - NOT NULL 제약조건은 반드시 컬럼 레벨에서만 정의가 가능합니다.
- 오라클과 DB2에서 제공하는 제약조건의 종류로는 다음과 같습니다.
 - NOT NULL 조건 : 컬럼을 필수 필드와 시킬 때 사용합니다.
 - UNIQUE 조건 : 데이터의 유일성을 보장(중복되는 데이터가 존재할수 없습니다.) 자동으로 index가 생성됩니다
 - CHECK 조건 : 컬럼의 값을 어떤 특정 범위로 제한할 수 있습니다.
 - DEFAULT(컬럼 기본값) 지정 : 데이터 입력시에 입력을 하지 않아도 지정된 값이 입력될수 있습니다.
 - PRIMARY KEY 지정 : 기본키는 UNIQUE 와 NOT NULL의 결합과 같습니다.
 - FOREIGN KEY(외래 키)지정 : 기본키를 참조하는 컬럼 또는 컬럼들의 집합입니다.

예문

오라클

```
CREATE TABLE EMP (
  EMPNO NUMBER CONSTRAINT emp_pk PRIMARY,
  ENAME VARCHAR2(20),
  JOB VARCHAR2(40),
  COMM NUMBER )
PCTFREE 20 PCT USED 50;

ALTER TABLE EMP ADD CONSTRAINT emp_ck
CHECK ( comm >= 10 AND comm <= 100000);

ALTER TABLE EMP DROP CONSTRAINT emp_ck;
```




DB2

```
CREATE TABLE EMP (
  EMPNO INTEGER NOT NULL
  CONSTRAINT emp_pk PRIMARY KEY,
  ENAME VARCHAR(20),
  JOB VARCHAR(40),
  COMM INTEGER )
IN DMS01;

ALTER TABLE EMP ADD CONSTRAINT emp_ck
CHECK ( comm >= 10 AND comm <= 100000);

ALTER TABLE EMP DROP CONSTRAINT emp_ck;
```

Point  Partitioned 테이블의 특정 컬럼의 데이터를 키 값을 기반으로 파티션이라고 불러주는 스토리지 오브젝트에 나뉘어 지도록 하는 기능입니다. 오라클과 DB2는 모두 테이블 파티션을 지원합니다.

파티션 테이블 비교

	오라클	DB2
파티션 설명	<ul style="list-style-type: none"> Range, Hash, List, Composite 등의 파티션 테이블이 존재합니다. 기존 단일 테이블을 파티션 테이블로 붙일 수 있는 방법은 없습니다. 파티션 테이블의 파티션을 단일 테이블로 분리할 수 있습니다. 	<ul style="list-style-type: none"> Range 파티션 테이블이 존재합니다. ALTER TABLE의 "ATTACH" 옵션을 이용하여 기존 테이블을 파티션 테이블의 파티션으로 붙일 수 있습니다. ALTER TABLE의 "DETACH" 옵션을 이용하여 파티션 테이블의 파티션을 단일 테이블로 분리할 수 있습니다.
파티션 정보 조회	<ul style="list-style-type: none"> DBA_TAB_PARTITIONS 덱서너리 테이블에 파티션 정보가 포함됩니다. 	<ul style="list-style-type: none"> SYSCAT.DATAPARTITIONS 카탈로그 테이블을 조회하면 파티션 정보를 확인할 수 있습니다.

예문

오라클

파티션 테이블 생성 :

```
CREATE TABLE ORDER_TRANS
(
  ORD_NUMBER NUMBER(12),
  ORD_DATE DATE,
  PROD_ID VARCHAR2(15),
  QUANTITY NUMBER(15,3) )
PARTITION BY RANGE ( ORD_DATE )
(
  PARTITION FY2007Q4 VALUES LESS THAN
    TO_DATE('01012008','MMDDYYYY') )
  TABLESPACE FY2007Q4,
  PARTITION FY2008Q1 VALUES LESS THAN
    TO_DATE('04012008','MMDDYYYY') )
  TABLESPACE FY2008Q1,
  STORAGE( INITIAL 500M NEXT 500M )
)
STORAGE ( INITIAL 200M NEXT 200M ) NOLOGGING;
```

파티션 추가 :

```
ALTER TABLE order_trans
  ADD PARTITION 2008Q3
  VALUES LESS THAN
  TO_DATE('10012008','MMDDYYYY') )
  TABLESPACE FY2008Q3 ;
```

단일 테이블을 파티션 테이블의 파티션으로 병합: 존재하지 않음.

기존 파티션을 단일 테이블로 분리:

```
ALTER TABLE ORDER_TRANS
  EXCHANGE PARTITION FY2007Q4 WITH TABLE
  ORDER_2007Q4 INCLUDING INDEXES
  WITHOUT VALIDATION
  EXCEPTIONS INTO uwclass.problems;
```

DB2

파티션 테이블 생성 :

```
CREATE TABLE ORDER_TRANS
(
  ORD_NUMBER BIGINT,
  ORD_DATE DATE,
  PROD_ID VARCHAR(15),
  QUANTITY DEC(15,3)
)
PARTITION BY RANGE ( ORD_DATE )
(
  PARTITION FY2007Q4
  STARTING '10/1/2007' INCLUSIVE
  ENDING '1/1/2008' EXCLUSIVE
  IN FY2007Q4
  , PARTITION FY2008Q1
  STARTING '1/1/2008' INCLUSIVE
  ENDING '4/1/2008' EXCLUSIVE
  IN FY2008Q1
) NOT LOGGED INITIALLY;
```

파티션 추가 :

```
ALTER TABLE order_trans
  ADD PARTITION FY2008Q3
  STARTING '7/1/2008' INCLUSIVE
  ENDING '10/1/2008' EXCLUSIVE
  IN FY2008Q3;
```

단일 테이블을 파티션 테이블의 파티션으로 병합:


```
ALTER TABLE ORDER_TRANS
  ATTACH PARTITION FY2008Q2
  STARTING '4/1/2008' INCLUSIVE
  ENDING '7/1/2008' EXCLUSIVE
  FROM TABLE NEW_TRANS;
```

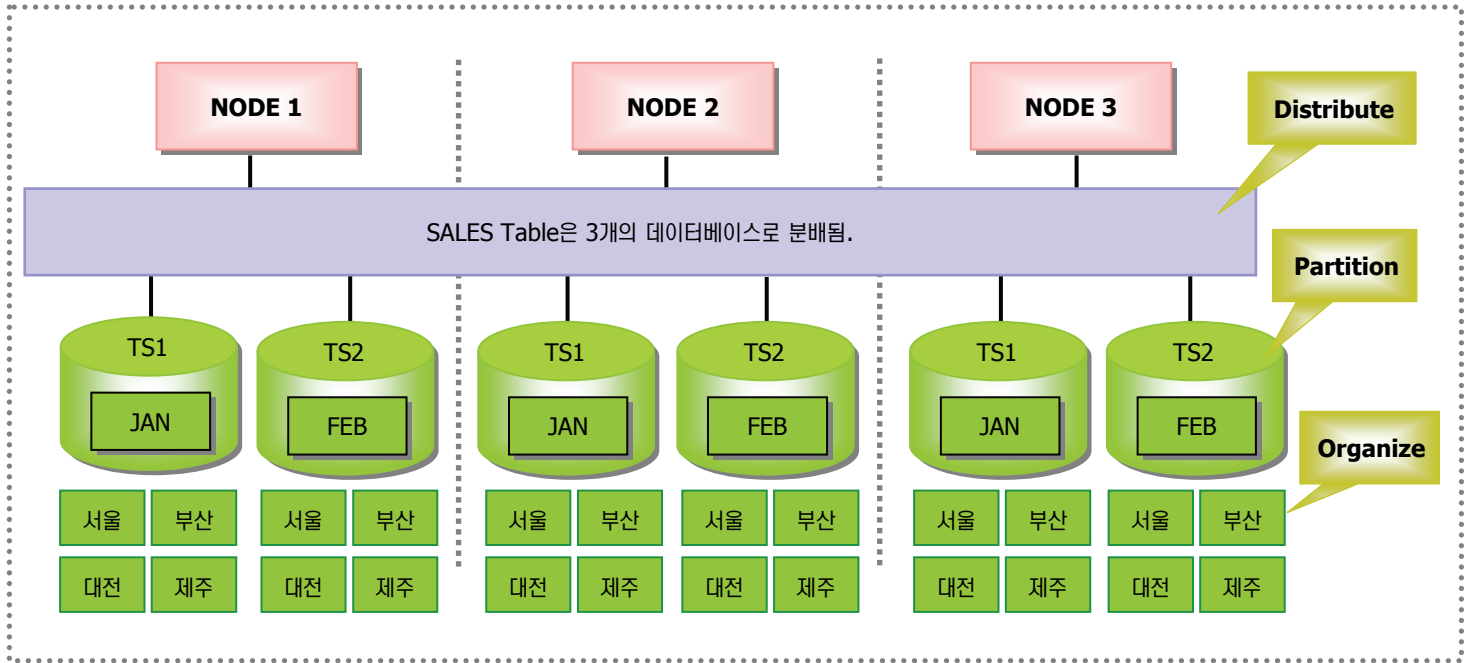
기존 파티션을 단일 테이블로 분리:

```
ALTER TABLE ORDER_TRANS
  DETACH PARTITION FY2007Q4 INTO FY2007Q4;
```



07 DB2 파티셔닝 기능

Point  대용량 데이터베이스와 테이블은 성능 및 확장에 있어서 많은 제약사항이 따릅니다. DB2는 데이터베이스 파티셔닝, 테이블 파티셔닝 및 다차원 클러스터링을 통해 데이터를 분배 및 클러스터함으로써 성능을 향상시킬 수 있습니다. 다음은 DB2에서 지원하는 파티셔닝 기능을 살펴봅니다.



파티셔닝 기능

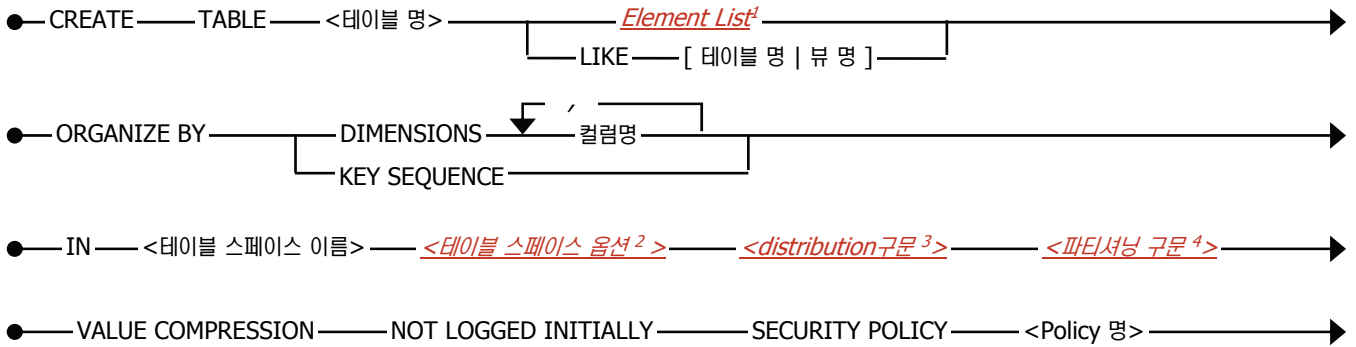
DB2는 다음과 같이 3가지 방식을 이용하여 데이터를 분산할 수 있습니다.

	Database Partitioning Feature	다차원 클러스터링 (MDC)	테이블 파티셔닝 (TP)
설명	<ul style="list-style-type: none"> 데이터베이스 파티셔닝 전반에 골고루 레코드를 분배합니다. 	<ul style="list-style-type: none"> 테이블내의 유사한 값을 갖는 레코드를 Block 이라고 하는 같은 물리적인 위치에 다차원적으로 그룹핑하여 저장합니다. 	<ul style="list-style-type: none"> 지정된 범위내에 있는 데이터를 그룹핑하여 같은 데이터 파티션에 넣습니다.
혜택	<ul style="list-style-type: none"> 확장성이 좋아집니다. 	<ul style="list-style-type: none"> 쿼리 성능이 좋아집니다. 	<ul style="list-style-type: none"> 쿼리 성능이 좋아지며 데이터 이동이 용이합니다.
테이블 디자인	<ul style="list-style-type: none"> 대용량 테이블로 서버의 리소스를 초과하는 테이블일 경우 적합합니다. 	<ul style="list-style-type: none"> 쿼리의 결과값이 동일한 값을 갖는 레코드를 반환하는 경우에 적합합니다. 	<ul style="list-style-type: none"> 대량의 레코드가 주기적으로 추가되거나 주기적으로 삭제될 경우 적합합니다.
CREATE TABLE 구문 절	<ul style="list-style-type: none"> DISTRIBUTE BY HASH 	<ul style="list-style-type: none"> ORGANIZE BY DIMENSION 	<ul style="list-style-type: none"> PARTITION BY RANGE
파티션 키	<ul style="list-style-type: none"> Distribution Key 	<ul style="list-style-type: none"> Dimension 	<ul style="list-style-type: none"> table partition key
파티션 키 추천컬럼	<ul style="list-style-type: none"> 서로 다른 값이 많은 컬럼을 선택합니다. 	<ul style="list-style-type: none"> date 컬럼을 포함하여 region 과 product_type과 같이 서로 다른 컬럼을 선택합니다. 	<ul style="list-style-type: none"> 시간에 관련된 데일을 갖는 컬럼을 선택합니다.
인덱스	<ul style="list-style-type: none"> 인덱스는 테이블이 위치한 데이터베이스 파티션에 존재합니다. 	<ul style="list-style-type: none"> 레코드 기반의 인덱스가 아닌 Block 기반의 인덱스가 생성되어 Block 을 가르키는 포인터 정보가 들어 있습니다. 	<ul style="list-style-type: none"> 인덱스는 하나의 테이블 스페이스에 저장됩니다.

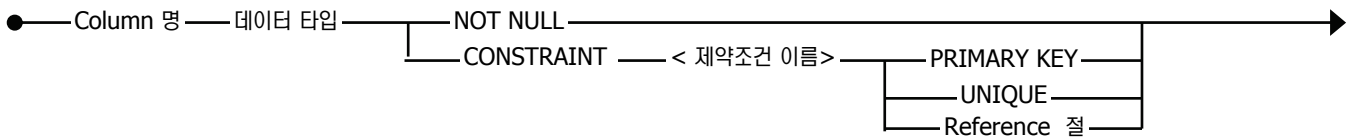
08 명령어를 이용한 테이블 생성

Point 명령어를 이용하여 DB2 테이블을 생성합니다.

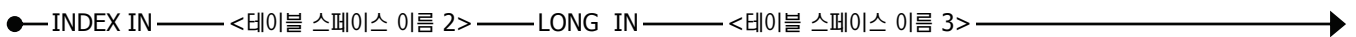
CREATE TABLE 구문



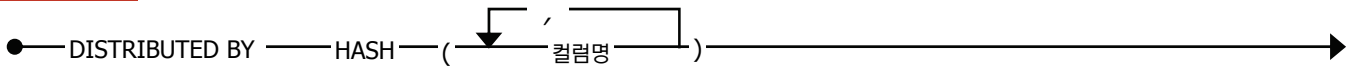
Element List¹:



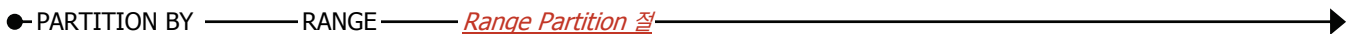
테이블 스페이스 옵션²



Distribution 구문³:




Distribution 구문⁴:

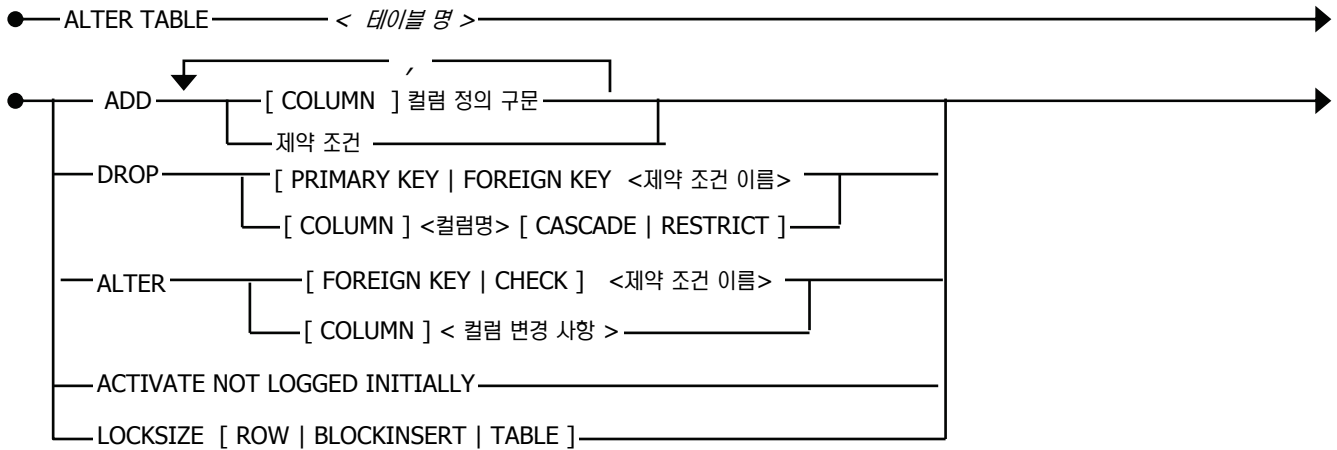


참고 : 이외에도 더 많은 옵션이 있으므로 도움말을 참고하시기 바랍니다.

예문

```
CREATE TABLE EMPLOYEE
(
  ID          SMALLINT          NOT NULL,
  NAME       VARCHAR(9),
  DEPT       SMALLINT CHECK ( DEPT BETWEEN 10 AND 100 ),
  JOB        CHAR(5) CHECK ( JOB IN ( 'Sales', 'Mtr', 'Clerk' ) ),
  HIREDATE   DATE,
  SALARY     DECIMAL ( 7,2 ),
  COMM       DECIMAL ( 7 2 ),
  PRIMARY KEY ( ID ),
  CONSTRAINT YEARSAL CHECK ( YEAR( HIREDATE ) > 1986  OR SALARY > 40500 )
)
IN EMPTBS
INDEX IN IDXTBS
```

Point  명령어를 이용하여 DB2 테이블을 변경 및 삭제합니다.

ALTER TABLE 구문

참고) 이외에도 더 많은 옵션이 있으므로 도움말을 참고하시기 바랍니다.

DROP TABLE 구문

● DROP TABLE < TABLE명 >

예문Column 추가 예제 :

```
ALTER TABLE DEPARTMENT ADD COLUMN RATING CHAR(1);
ALTER TABLE PROJECT ADD STTE_NOTES VARCHAR(1000) FOR BIT DATA;
```

외래 키 추가 예제 :

```
ALTER TABLE EQUIPMENT
  ADD CONSTRAINT DEPTQUIP
  FOREIGN KEY ( EQUIP_OWNER) REFERENCES DEPARTMENT ON DELETE SET NULL;
```

CHECK 제약 조건 추가 예제 :

```
ALTER TABLE EMPLOYEE ADD CONSTRAINT REVENUE CHECK ( SALARY + COMM > 30000 );
```

기존 컬럼 제거 예제 :

```
ALTER TABLE EMPLOYEE DROP CONSTRAINT REVENUE;
```

디폴트 값 변경 예제 :

```
ALTER TABLE EMPLOYEE ALTER COLUMN WORKDEPT SET DEFAULT '123';
```


SECURITY Policy 추가 예제 :

```
ALTER TABLE EMPLOYEE ADD SECURITY POLICY DATA_ACCESS;
```

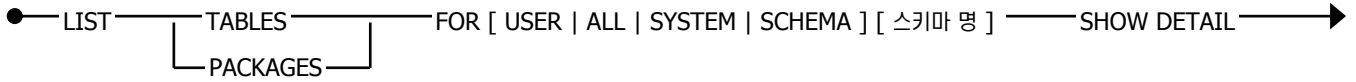
SALARY 컬럼을 보호하기 위해 EMPLOYEE 테이블 변경 :

```
ALTER TABLE EMPLOYEE ALTER COLUMN SALARY SECURED WITH EMPLOYEESECLABEL;
```


10 명령어를 이용한 테이블 조회

Point  명령어를 이용하여 DB2 테이블 스페이스 정보를 조회할 수 있습니다.

LIST TABLE 구문



파라미터 설명

파라미터	설명
USER	현재 사용자의 테이블 또는 패키지 목록을 조회합니다.
ALL	모든 사용자의 테이블 또는 패키지 목록을 조회합니다.
SYSTEM	시스템 테이블 또는 패키지 목록을 조회합니다 .
SCHEMA [스키마 명]	지정된 스키마의 테이블 또는 패키지 목록을 조회합니다.

예문


db2=> list tables for schema amj

테이블/뷰	스키마	유형	작성 시간
SALES	AMJ	T	2008-01-08-15.00.06.671000

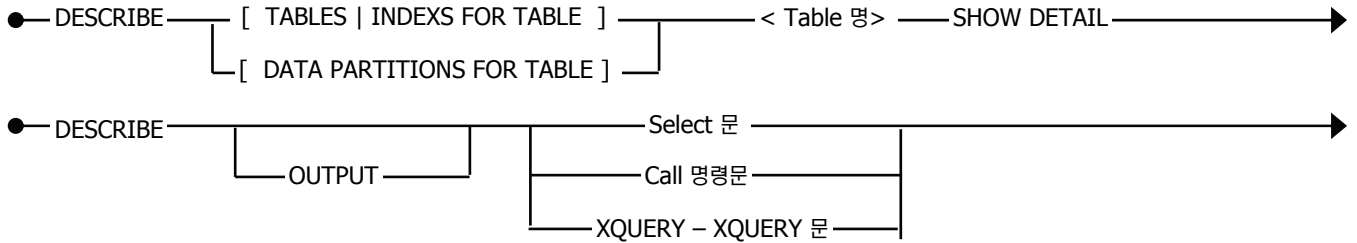
db2=> list tables for schema amj

테이블/뷰	스키마	유형	작성 시간
ATTRIBUTES	SYSCAT	V	2007-11-27-14.25.58.609007
AUDITPOLICIES	SYSCAT	V	2007-11-27-14.25.58.843001
AUDITUSE	SYSCAT	V	2007-11-27-14.25.58.937003
BUFFERPOOLDBPARTITIONS	SYSCAT	V	2007-11-27-14.25.58.984003
BUFFERPOOLNODES	SYSCAT	V	2007-11-27-14.25.59.015003
BUFFERPOOLS	SYSCAT	V	2007-11-27-14.25.59.015007
CASTFUNCTIONS	SYSCAT	V	2007-11-27-14.25.59.031003
CHECKS	SYSCAT	V	2007-11-27-14.25.59.031007
COLAUTH	SYSCAT	V	2007-11-27-14.25.59.046003
-- More --			

11 명령어를 이용한 테이블 구조 조회

Point  명령어를 이용하여 테이블 구조를 조회할 수 있습니다. 이는 오라클의 DESC 명령어와 유사합니다.

DESCRIBE 구문



파라미터 설명

파라미터	설명
TABLE	지정된 테이블 구조에 대한 정보를 조회합니다.
INDEXS FOR TABLE	지정된 테이블의 인덱스 정보를 조회합니다.
DATA PARTITIONS FOR TABLE	지정된 테이블의 데이터 파티션 정보를 조회합니다.

예문

db2=> describe table user1.departmentzzzz

Table: USER1.DEPARTMENT

Column name	Type schema	Type name	Length	Scale	Nulls
AREA	SYSIBM	SMALLINT	2	0	No
DEPT	SYSIBM	CHARACTER	3	0	No
DEPTNAME	SYSIBM	CHARACTER	20	0	Yes

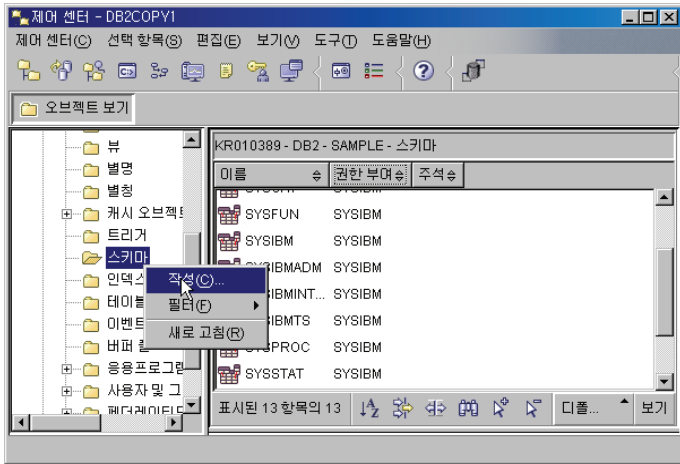
db2=> describe data partitions for table amj.sales show detail

PartitionId	Inclusive (y/n)	Low Value	Inclusive (y/n)	High Value
0	Y	'2008-01-01'	N	'2008-02-01'
1	Y	'2008-02-01'	N	'2008-03-01'
2	Y	'2008-03-01'	N	'2008-04-01'
3	Y	'2008-04-01'	N	'2008-05-01'
4	Y	'2008-05-01'	N	'2008-06-01'
5	Y	'2008-06-01'	N	'2008-07-01'
6	Y	'2008-07-01'	N	'2008-08-01'

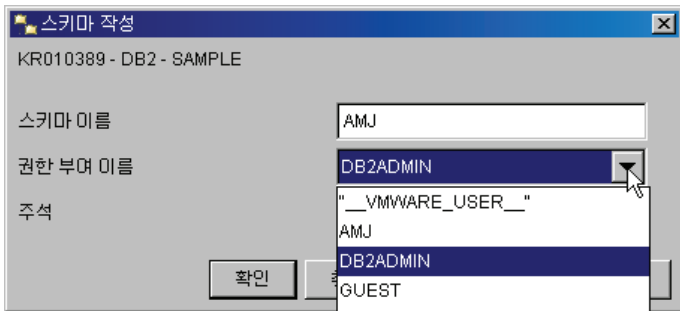
12 제어센타를 이용한 스키마 및 테이블 생성

Point 제어센타를 이용하여 스키마와 테이블을 생성합니다.

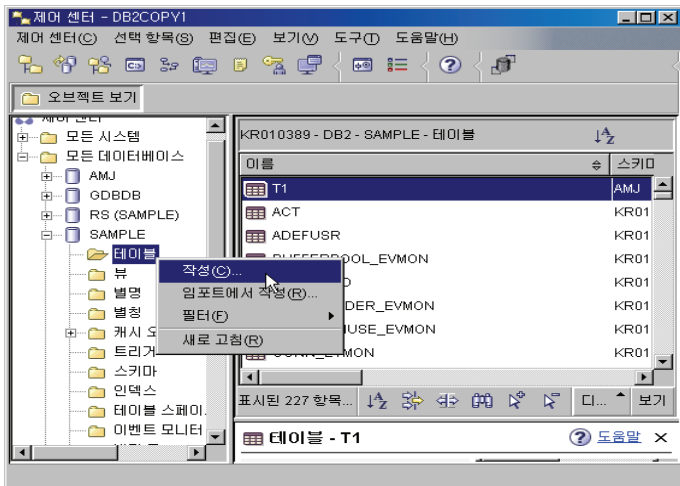
- 제어센타에서 스키마를 선택한 후 오른쪽 버튼을 클릭한 후 서브메뉴가 나타나면 작성 버튼을 눌러 스키마를 생성합니다.



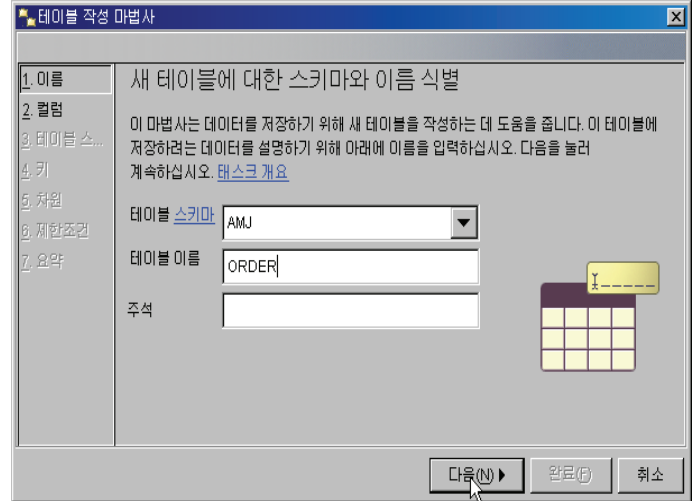
- 생성할 스키마를 입력하고 스키마에 대한 권한을 부여할 사용자 "DB2ADMIN"을 선택하고 확인 버튼을 클릭합니다.



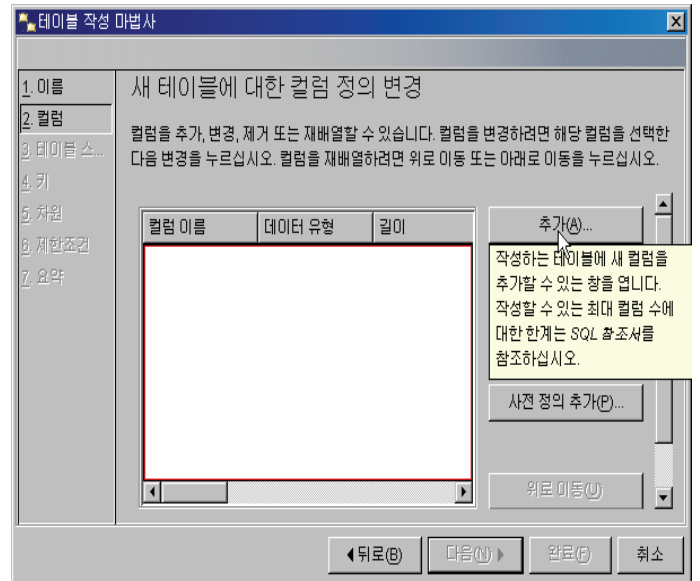
- 제어센타에서 테이블을 선택한 후 오른쪽 버튼을 클릭합니다. 서브메뉴중 작성 버튼을 눌러 테이블을 생성합니다.



- 테이블을 생성할 스키마와 테이블 이름을 지정한 후 "다음" 버튼을 클릭합니다.



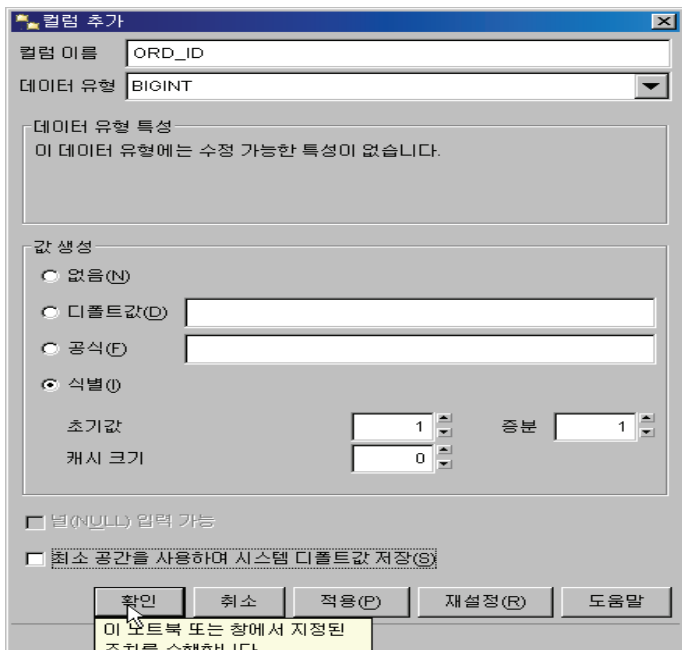
- 추가 버튼을 눌러 테이블의 컬럼을 정의합니다.



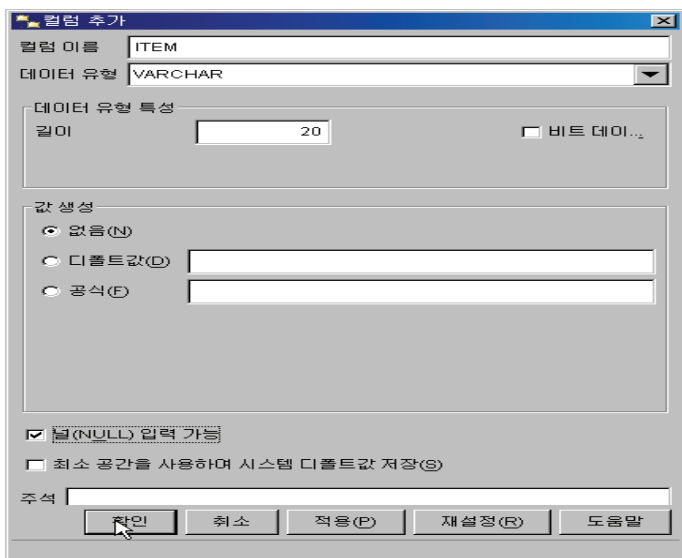
13 제어센타를 이용한 스키마 및 테이블 생성

Point 제어센타를 이용하여 스키마와 테이블을 생성합니다.

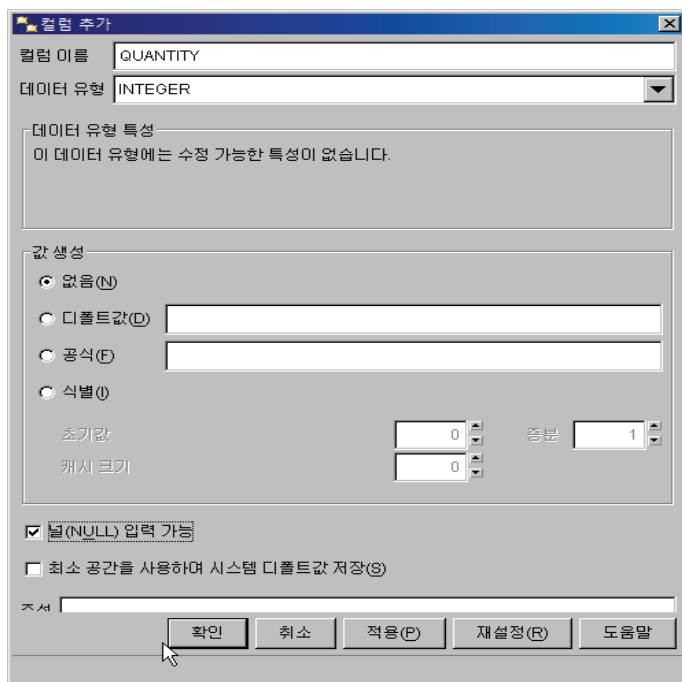
- 컬럼이름을 입력한 후 데이터 유형을 선택합니다. 값 생성에 대해 선택합니다. 값 생성은 컬럼에 대한 값이 지정되지 않았을 경우에 대해 어떠한 action 을 취할 것인지 결정합니다. 없음은 NULL 값을 입력합니다. 지정되지 않았을 경우 디폴트값이 입력됩니다. 공식은 지정된 계산값으로 컬럼의 값이 입력됨을 의미합니다. 식별은 시퀀스와 유사하게 초기값을 기반으로 증가값씩 늘리면서 값이 입력됨을 의미합니다. 식별은 값이 반드시 존재하므로 NULL 입력이 가능하지 않습니다. 초기값을 1로 변경한 후 확인 버튼을 클릭합니다.



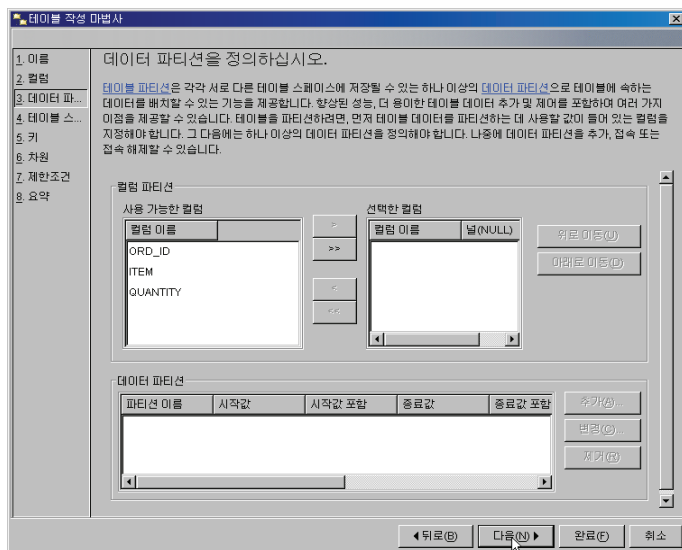
- 다시 추가 버튼을 눌러 ITEM 컬럼을 추가합니다. 데이터 유형은 VARCHAR, 길이는 20으로 변경한 후 NULL 입력 기능을 체크한 후 확인 버튼을 클릭합니다.



- 다시 추가 버튼을 눌러 QUANTITY 컬럼을 추가합니다. QUANTITY 컬럼의 데이터 유형은 INTEGER 로 지정하고 Null 입력이 가능하도록 체크박스에 체크한 후 확인 버튼을 누른 후 원 화면으로 돌아가서는 다음 버튼을 클릭합니다.



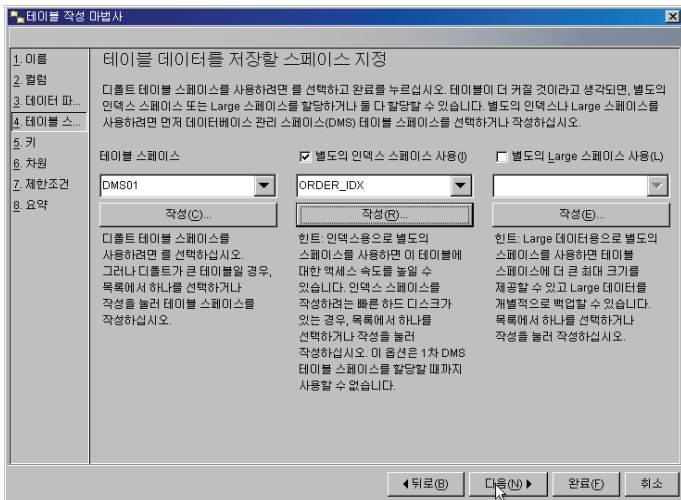
- 생성할 테이블을 파티션 테이블로 생성할 것인지 지정합니다. 이 곳에서는 파티션 테이블로 생성하지 않을 것이기 때문에 아무것도 지정하지 않은 채 다음 버튼을 클릭합니다.



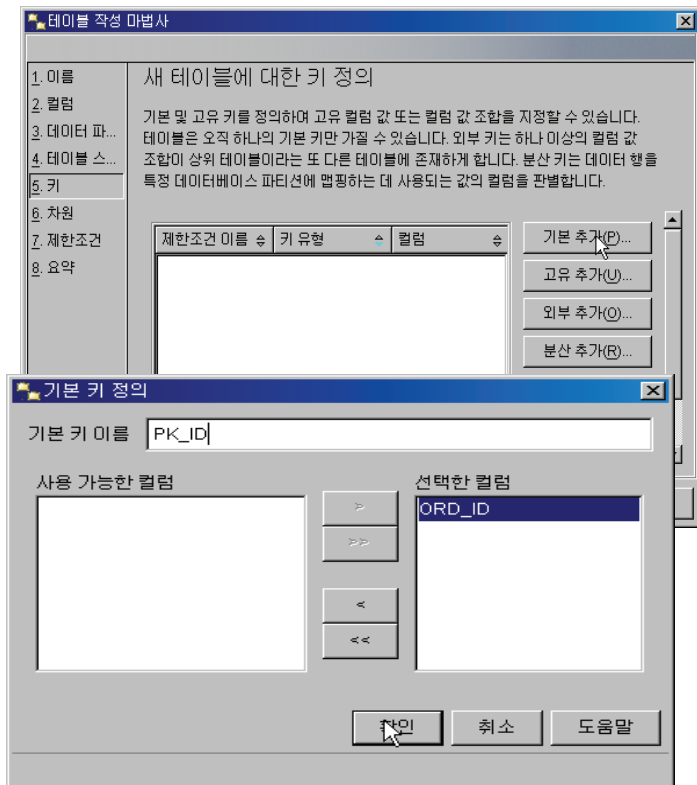
14 제어센타를 이용한 스키마 및 테이블 생성

Point 제어센타를 이용하여 스키마와 테이블을 생성합니다.

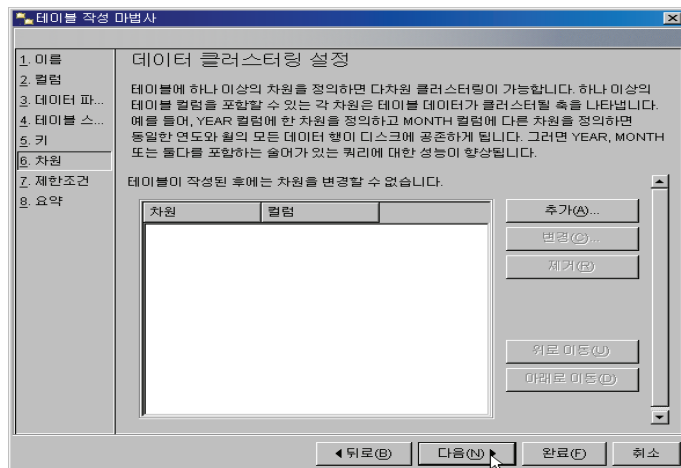
- 테이블이 저장될 테이블 스페이스를 선택합니다. 만약 인덱스를 별도의 공간에 저장하고 싶을 경우 별도의 인덱스 스페이스 사용을 체크하고 테이블 스페이스를 선택합니다. 또한 테이블내에 LOB 데이터를 별도의 공간에 저장하고 싶을 경우에도 별도의 LARGE 스페이스 사용을 체크하고 테이블 스페이스를 선택하면 됩니다. 작성 버튼을 누를 경우 이 곳에서 테이블 스페이스를 바로 생성할 수 있습니다.



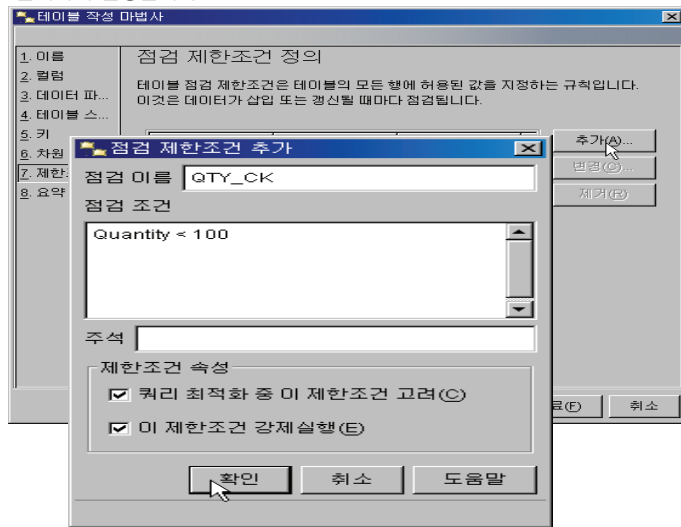
- 제약조건인 키를 정의합니다. 이 곳에서는 기본 키를 추가합니다. 기본 추가 버튼을 클릭한 후 기본 키 이름을 입력하고 기본 키를 정의할 컬럼을 선택합니다. 다음버튼을 눌러 진행합니다.



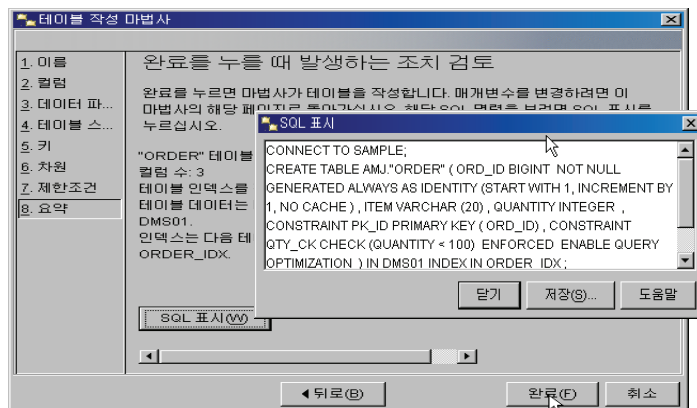
- 데이터 클러스터링을 설정할 수 있습니다. 그러나 현재 테이블은 다차원으로 데이터를 클러스터링할 필요가 없으므로 다음 버튼을 클릭합니다.




- 점검 제한 조건을 정의합니다. 추가 버튼을 눌러 점검 이름과 점검 조건 (Quantity < 100) 에 대해 입력한 후 확인버튼을 누른 후 다음 버튼을 클릭하여 진행합니다.



- 완료화면이 나타나면 SQL 표시 버튼을 눌러 SQL문을 확인한 후 완료버튼을 클릭합니다.



Point  다음은 명령어를 이용하여 대용량 테이블을 파티셔닝과 MDC를 이용한 방법입니다.

파티션 테이블 + MDC


```
CREATE TABLE AMJ.SALES
(
  DATE           DATE           NOT NULL ,
  REGION         VARCHAR(15)    NOT NULL ,
  PRODUCT_ID    BIGINT         NOT NULL ,
  REVENUE        DECIMAL(15,0) NOT NULL
)
PARTITION BY RANGE("DATE")
(
  PART FY20081Q STARTING('2008-01-01') IN TBS1Q LONG IN LTBS1Q,
  PART FY20082Q STARTING('2008-04-01') IN TBS2Q LONG IN LTBS2Q,
  PART FY20083Q STARTING('2008-07-01') IN TBS3Q LONG IN LTBS3Q,
  PART FY20084Q STARTING('2008-10-01') ENDING('2009-01-01') EXCLUSIVE IN TBS4Q LONG IN LTBS4Q
)
ORGANIZE BY
(
  ("REGION" , "PRODUCT_ID" )
);
```

```
CREATE UNIQUE INDEX AMJ.PARTITION_IDX ON AMJ.SALES ( DATE ASC ) INCLUDE ("REGION" )
NOT PARTITIONED IN DMS01
PCTFREE 10 MINPCTUSED 10 ALLOW REVERSE SCANS;
```

```
SELECT DATAPARTITIONNAME, TABSCHEMA, TABNAME,
FROM SYSCAT.DATAPARTITIONS WHERE TABSCHEMA = 'AMJ' AND TABNAME = 'SALES';
```

결과 값 >

DATAPARTITIONNAME	TABSCHEMA	TABNAME	LOWINCLUSIVE	LOWVALUE	HIGHINCLUSIVE	HIGHVALUE
FY20081Q	AMJ	SALES	Y	'2008-01-01'	N	'2008-04-01'
FY20082Q	AMJ	SALES	Y	'2008-04-01'	N	'2008-07-01'
FY20083Q	AMJ	SALES	Y	'2008-07-01'	N	'2008-10-01'
FY20084Q	AMJ	SALES	Y	'2008-10-01'	N	'2009-01-01'

Point  인덱스는 테이블과 결합된 선택적인 구조로 쿼리 속도를 향상시키기 위해 키와 물리적 저장 위치의 정보인 RID 정보가 저장됩니다. 오라클과 DB2는 쿼리 속도를 향상시키기 위해 모두 인덱스를 사용합니다.

인덱스 비교

	오라클	DB2
인덱스	<ul style="list-style-type: none"> ▪ B- Tree 구조를 사용합니다. ▪ 최대 32컬럼까지 하나의 인덱스에서 지정할 수 있습니다. ▪ UNIQUE 인덱스는 고유값만 저장되나 NULL 값일 경우 하나 이상의 NULL 값은 입력이 가능합니다. ▪ Clustered index는 clustered Table 만을 위해서만 사용됩니다. ▪ IOT는 Index에 모든 데이터가 들어 있는 테이블입니다. 	<ul style="list-style-type: none"> ▪ B+ Tree 구조를 사용합니다. ▪ 최대 64 컬럼까지 하나의 인덱스에서 지정할 수 있습니다. ▪ UNIQUE 인덱스는 고유값만 저장되나 NULL 값일 경우 하나 이상의 NULL 값은 입력이 가능합니다. ▪ ALTER 명령문은 존재하지 않습니다. 변경할 경우 삭제한 후 재생성합니다. ▪ Cluster Index는 테이블당 하나만 사용 가능합니다. ▪ Include 옵션을 두어 인덱스 컬럼은 아니지만 자주 액세스하는 컬럼을 인덱스에 포함시켜 인덱스만 조회해도 데이터를 읽을 수 있도록 할 수 있습니다.
인덱스 정보 조회	<ul style="list-style-type: none"> ▪ DBA_INDEXES 디렉터리 테이블을 조회합니다. 	<ul style="list-style-type: none"> ▪ SYSCAT.INDEXES 카탈로그 테이블을 조회합니다. ▪ describe indexes for table <테이블 명> show detail 명령어를 이용하여 조회합니다.

예문

오라클

```
CREATE INDEX IDX_SALES ON SALES ( DATE ASC);
DROP INDEX IDX_SALES;
ALTER INDEX REBUILD;
```



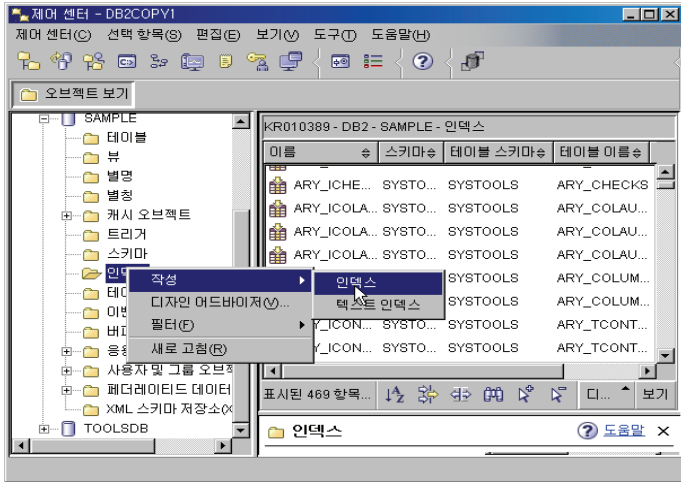
DB2

```
CREATE INDEX IDX_SALES ON SALES ( DATE ASC);
DROP INDEX IDX_SALES;
ALTER INDEX는 존재하지 않습니다.
```

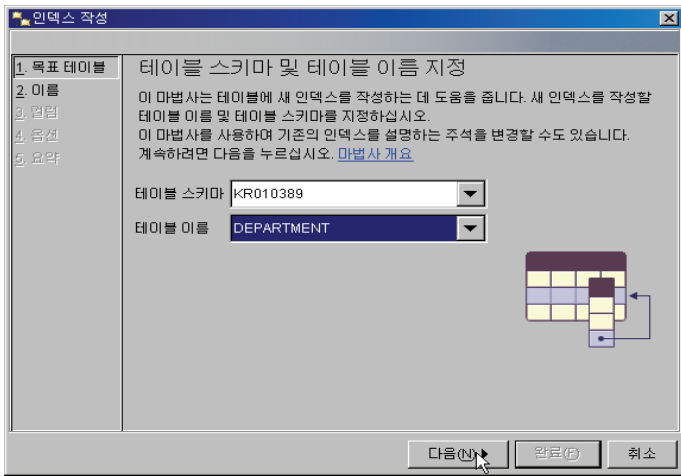
17 제어센타를 이용한 인덱스 생성

Point 제어센타를 이용하여 인덱스를 생성합니다.

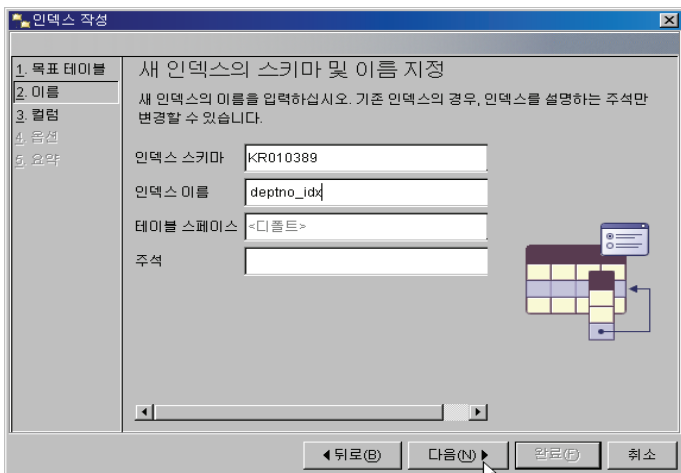
- 제어센타에서 인덱스를 선택한 후 오른쪽 버튼을 클릭합니다. 서브메뉴중 작성 버튼을 눌러 인덱스를 생성합니다.



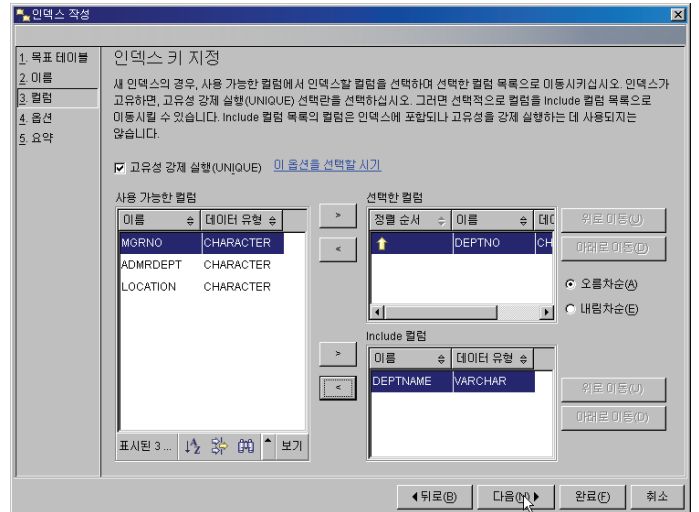
- 인덱스를 생성할 스키마와 테이블 이름을 선택하고 다음 버튼을 클릭합니다.



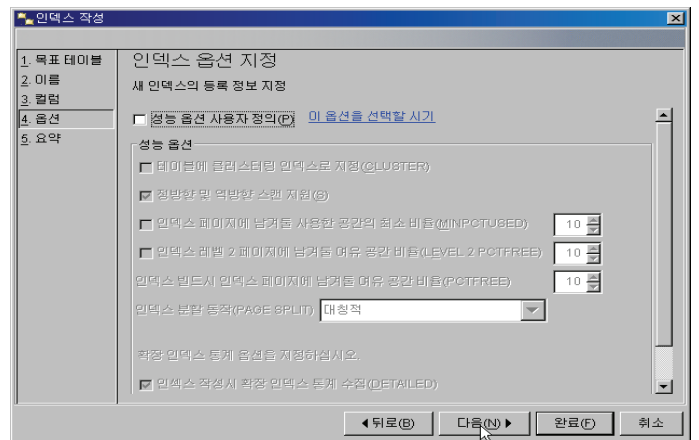
- 인덱스 이름을 입력하고 다음 버튼을 클릭합니다.



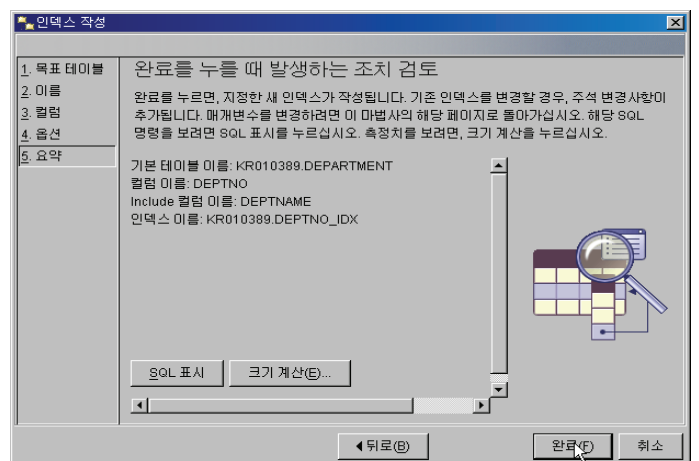
- 인덱스를 생성할 컬럼을 선택합니다. "고유성 강제 실행"을 체크한 후 인덱스 컬럼은 아니지만 인덱스 테이블에 같이 포함시킬 DEPTNAME 컬럼을 선택합니다.




- 성능 옵션을 직접 지정하려면 "성능 옵션 사용자 정의"를 체크합니다. 디폴트 지정시 다음버튼을 클릭하여 진행합니다.



- SQL 표시 버튼을 눌러 SQL문을 확인할 수 있습니다. 확인한 후 완료버튼을 클릭하여 인덱스를 생성합니다.



Point  시퀀스는 자동 생성 일련 번호로써 오라클과 DB2의 시퀀스의 기능은 거의 동일하며 구문만 상이합니다.

시퀀스 비교

	오라클	DB2
시퀀스 사용 방법	<ul style="list-style-type: none"> • <시퀀스 이름>.CURRVAL과 <시퀀스 이름>.NEXTVAL를 이용하여 현재값과 다음 값을 조회합니다. 	<ul style="list-style-type: none"> • PREVAL FOR <시퀀스 이름>과 NEXTVAL FOR <시퀀스 이름>를 이용하여 현재값과 다음 값을 조회합니다.
시퀀스 정보 조회	<ul style="list-style-type: none"> • DBA_SEQUENCES 덱서너리 테이블을 조회하면 시퀀스 정보를 조회할 수 있습니다. 	<ul style="list-style-type: none"> • SYSCAT.SEQUENCES 카탈로그 테이블을 조회하면 시퀀스 정보를 조회할 수 있습니다.

예문

오라클

```
CREATE SEQUENCE ORDER_SEQ
  INCREMENT BY 1
  START WITH 100
  NO MAXVALUE
  NO CYCLE
  NOCACHE ;

INSERT INTO ORDER_HIS VALUES
(  ORDER_SEQ.NEXTVAL,
  'BOOK',
  100
);


SELECT ORDER_SEQ.CURRVAL FROM DUAL;
DROP SEQUENCE ORDER_SEQ;
```

DB2

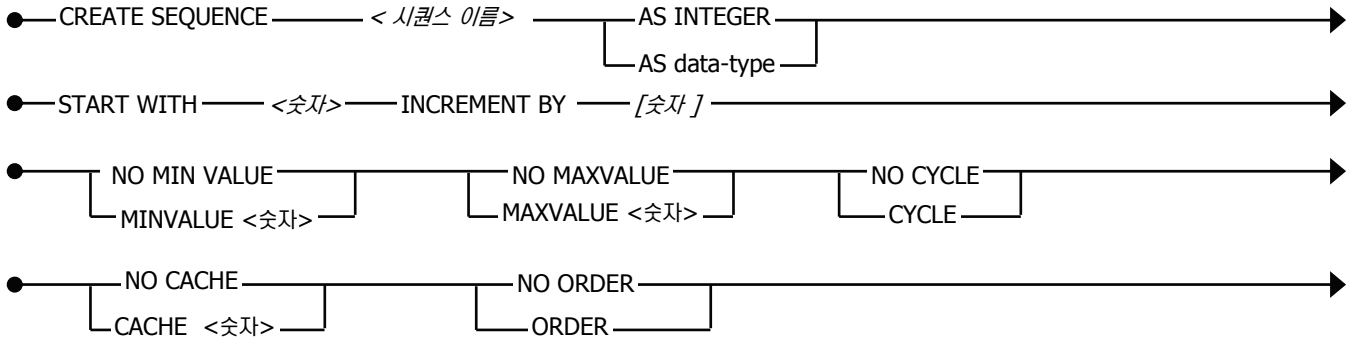
```
CREATE SEQUENCE ORDER_SEQ
  START WITH 100
  INCREMENT BY 1
  NO MAXVALUE
  NO CYCLE
  NOCACHE ;

INSERT INTO ORDER_HIS VALUES
(  NEXTVAL FOR ORDER_SEQ ,
  'BOOK',
  100
);

SELECT PREVAL FOR ORDER_SEQ
FROM SYSIBM.SYSDUMMY1;
DROP SEQUENCE ORDER_SEQ;
```

Point  다음은 DB2 시퀀스 생성 방법입니다.

시퀀스 생성 구문



참고 : 이외에도 다양한 옵션이 지정되어 있으므로 도움말을 참고하시기 바랍니다.

옵션 설명

옵션	설명
AS < data-type >	SMALLINT, INT, BIGINT, DECIMAL 중에서 원하는 데이터 유형을 선택합니다.
START WITH	음수 또는 양수의 시작값을 지정합니다.
INCREMENT BY	음수 또는 양수의 증가값을 지정합니다.
MINVALUE	음수 또는 양수의 최소값을 지정합니다.
MAXVALUE	음수 또는 양수의 최대값을 지정합니다.
CYCLE	최대값 또는 최소값에 도달하면 최소값 또는 최대값을 생성합니다.
CACHE	지정된 개수의 생성값을 미리 캐시에 보관하는 옵션으로 성능 향상에 도움이 됩니다.

참고 : 이외에도 다양한 옵션이 지정되어 있으므로 도움말을 참고하시기 바랍니다.

예문

시퀀스 생성

```

CREATE SEQUENCE order_seq
START WITH 100
INCREMENT BY 1
NO MAXVALUE
NO CYCLE
CACHE 24
    
```

시퀀스 활용

```

INSERT INTO ORDER VALUES
(
    nextval for order_seq
    , ' PAPER BAG'
);
INSERT INTO ORDER VALUES
(
    nextval for order_seq
    , "NOTE"
);
    
```


시퀀스 조회

```

Select * from ORDER;
ORDER_NUM  ORDER_NAME
-----
100        PAPER BAG
101        NOTE
    
```

```

Select prevval for order_seq
FROM SYSIBM.SYSDUMMY1;
1
101
    
```


Point  트리거는 특정 테이블에 대해 INSERT, UPDATE, DELETE 문이 실행될 때 자동으로 실행되는 프로그램입니다.

트리거 비교

	오라클	DB2
트리거	<ul style="list-style-type: none"> 스토리지 옵션을 두어 테이블마다 스토리지를 제어할 수 있으며 스토리지 옵션 생략시 지정된 테이블 스페이스의 디폴트 값을 적용합니다. 	<ul style="list-style-type: none"> REPLACE 를 지원하지 않으므로 변경시에는 트리거를 제거한 후 다시 생성합니다.
트리거 발생 시점	<ul style="list-style-type: none"> 트리거 이벤트의 BEFORE, AFTER로 구분합니다. 	<ul style="list-style-type: none"> 트리거 이벤트의 BEFORE, AFTER로 구분합니다.
트리거 이벤트	<ul style="list-style-type: none"> INSERT, UPDATE, DELETE 입니다. 	<ul style="list-style-type: none"> INSERT, UPDATE, DELETE 입니다.

예문

오라클

```
CREATE OR REPLACE TRIGGER con_audit_trg
  BEFORE INSERT ON CON_AUDIT
  FOR EACH ROW
BEGIN
  :new.timestamp := SYSDATE;
END;

CREATE OR REPLACE TRIGGER con_audit_trg
  BEFORE INSERT ON CON_AUDIT
  FOR EACH ROW
BEGIN
  :new.timestamp := SYSDATE;
END;

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 con_audit_trg
  NO CASCADE BEFORE INSERT ON CON_AUDIT
  REFERENCING NEW as n
  FOR EACH ROW
  MODE DB2SQL
BEGIN ATOMIC
  SET n.timestamp = CURRENT TIMESTAMP;
END!

CREATE TRIGGER con_audit_trg
  NO CASCADE BEFORE INSERT ON CON_AUDIT
  REFERENCING NEW as n
  FOR EACH ROW
  BEGIN ATOMIC
  SET n.timestamp = CURRENT TIMESTAMP;
END!

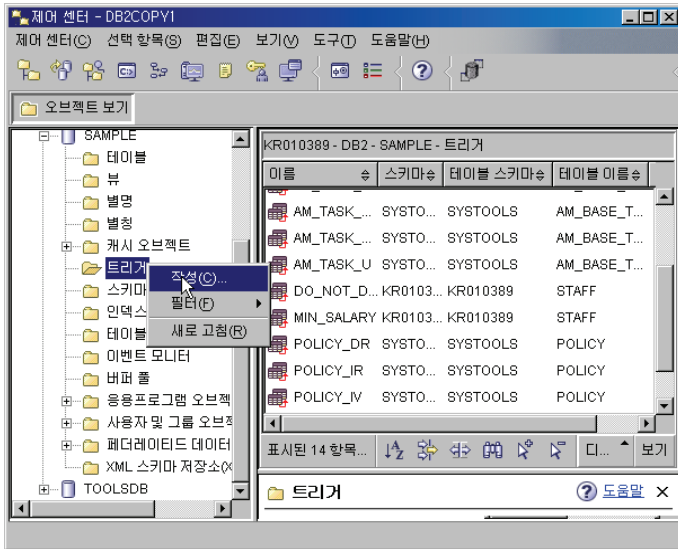
CREATE TRIGGER emp_history_trg
  AFTER DELETE ON EMPLOYEES
  REFERENCING OLD as d
  FOR EACH ROW
  BEGIN ATOMIC
  INSERT INTO emp_history ( emp_id
    , first_name
    , last_name )
  VALUES ( d.emp_id ,d.first_name,d.last_name );
END!
```



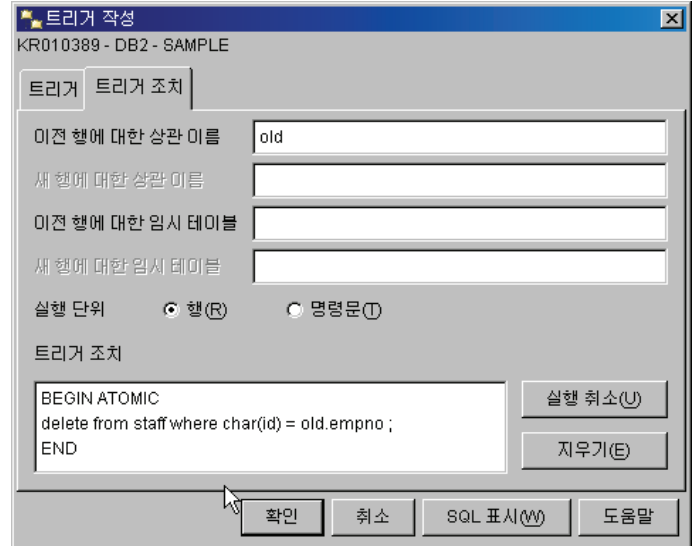
21 제어센타를 이용한 트리거 생성

Point 제어센타를 이용하여 트리거를 생성합니다.

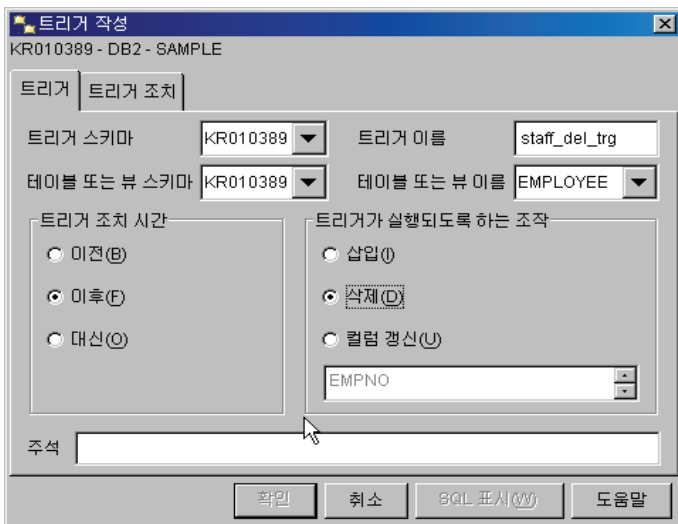
- 제어센타에서 트리거를 선택한 후 오른쪽 버튼을 클릭합니다. 서브메뉴중 작성 버튼을 눌러 트리거를 생성합니다.



- 트리거 조치 탭으로 이동하여 이전 행에 대한 상관 이름을 "old"로 지정합니다. 트리거 조화에 수행할 SQL문을 아래와 같이 입력하고 확인버튼을 클릭합니다.



- 트리거 이름을 "staff_del_trg"로 입력하고 이벤트를 발생시킬 EMPLOYEE 테이블과 그 테이블이 속하는 KR010389 스키마를 선택합니다. 트리거가 실행될 시점을 "이후"로 선택하고 트리거가 실행되도록 하는 조작을 "삭제"로 선택한 후 확인 버튼을 클릭합니다.



Point 함수에는 내장함수와 사용자가 작성하는 함수가 있습니다. 오라클에서는 사용자가 작성하는 함수를 Stored Function 이라 하고 DB2 에서는 User Defined Function(UDF)라 합니다. 함수는 프로시저와 달리 리턴값이 존재합니다. 이번 장에서는 함수 및 내장 함수를 비교합니다.


함수 비교

	오라클	DB2
제약 사항	<ul style="list-style-type: none"> 테이블 생성시 CHECK, DEFAULT 제약 조건내에 사용자 함수를 사용할 수 없습니다. 	<ul style="list-style-type: none"> 테이블 생성시 CHECK 제약 조건내에서 사용자 함수 사용 가능합니다.
함수 리턴 타입	<ul style="list-style-type: none"> 테이블 및 Boolean 타입으로 리턴하지 못합니다. 	<ul style="list-style-type: none"> 테이블 타입으로 리턴이 가능합니다.
외부 루틴 사용 가능 여부	<ul style="list-style-type: none"> JAVA 또는 C 프로그램으로 작성된 외부 루틴을 호출하여 함수를 작성할 수 있습니다. 	<ul style="list-style-type: none"> JAVA 또는 C 프로그램으로 작성된 외부 루틴을 호출하여 함수를 작성할 수 있으며 OLE DB Provider에 등록하여 함수를 작성할 수 있습니다.
함수 생성 구문	<pre>CREATE [OR REPLACE] FUNCTION func_name [(변수명 [IN OUT] datatype [DEFAULT value] [, ..])] RETURN datatype [PIPELINED] { IS AS } SQL 블록 /</pre>	<pre>CREATE FUNCTION func_name [(변수명 datatype (size) [, ..])] RETURN { Scalar_return_data_type TABLE (column_list) } LANGUAGE SQL [READ SQL DATA CONTAINS SQL] SQL 블록 @</pre>
예문	<pre>sql> CREATE OR REPLACE FUNCTION test01 (x number DEFAULT 1000) RETURN number IS BEGIN RETURN (0.5 * x * x * 3.14); END; /</pre>	<pre>db2=> CREATE FUNCTION test01 (x numeric(10,2)) RETURNS numeric(10,2) LANGUAGE SQL BEGIN ATOMIC RETURN (0.5 * x * x * 3.14); END@</pre>

* 부연설명

오라클은 함수 또는 스토어드 구문내에서 SQL 문의 결과 함수 끝을 표시하는 문자로 세미콜론을 사용합니다. 이렇게 작성된 스크립트를 @a.sql 문을 이용하여 수행합니다. 그러나 DB2에서는 함수(또는 프로시저) 결과 함수내에 포함된 SQL문의 끝을 표시하는 문자가 반드시 달라야 합니다. SQL문의 끝을 세미콜론으로 표시하였다면 함수의 끝은 @ 또는 ! 문자로 표시해야 합니다. 그리고 이 스크립트는 "db2 -td! -vf"와 같은 명령어 옵션을 사용하여 수행합니다.

23 내장 함수 비교 - 숫자

Point  숫자 함수를 비교합니다.

숫자 함수 비교

오라클	DB2	설명
ABS	ABS	• 절대값을 리턴합니다.
FLOOR	FLOOR	• Argument 값과 같거나 그보다 작은 값을 리턴합니다.
MOD	MOD	• 첫번째 argument 값을 두번째 argument로 나누었을때 나머지 값을 리턴합니다.
POWER	POWER	• 첫번째 Argument에 대한 두번째 argument의 승수값을 리턴합니다.
N/A	RAND	• 무작위 숫자를 리턴합니다.
ROUND(arg1,arg2)	ROUND(arg1,arg2)	• 반올림값을 리턴합니다.
TRUNC(n[,m])	TRUNC (n[,m])	• 버림값을 리턴합니다.
LEAST	LEAST	• Argument 집합 중 가장 작은 값을 리턴합니다.
GREATEST	GREATEST	• Argument 집합 중 가장 큰 값을 리턴합니다.
CEIL	CEIL	• 올림값을 리턴합니다.

예문

오라클

```
sql> SELECT ABS(-2) FROM DUAL;
      ABS(-2)
-----
           2
sql> SELECT ROUND(45.926,2) FROM DUAL;
      ROUND ( 45.926,2)
-----
          45.93
sql> SELECT TRUNC(45.926,2) FROM DUAL;
      ROUND ( 45.926,2)
-----
          45.92
sql> SELECT MOD(1600,300) FROM DUAL;
      MOD(1600,300)
-----
           100
sql> SELECT CEIL(13.11) FROM DUAL;
      CEIL(13.11)
-----
           14
```

DB2

```
db2=> SELECT ABS(-2) FROM SYSIBM.SYSDUMMY1
      1
-----
      2
db2=> VALUES ROUND(45.926,2)
      1
-----
      45.930
db2=> VALUES TRUNC(45.926)
      1
-----
      45.920
db2=> VALUES MOD(1600,300)
      1
-----
      100
db2=> SELECT CEIL(13.11) FROM SYSIBM.SYSDUMMY1
      1
-----
      14.
db2=> VALUES RAND()
      1
-----
+1.25125888851588E-003
```

24 내장 함수 비교 - 문자

Point 문자 함수를 비교합니다.

문자 함수 비교

오라클	DB2	설명
ASCII	ASCII	• 문자의 숫자 값을 리턴합니다.
CHR(n)	CHR(n)	• n을 표시하는 ASCII 코드값을 리턴합니다.
CONCAT	CONCAT	• 두개의 문자열을 붙여 리턴합니다.
N/A	GENERATE_UNIQUE	• 동일한 함수의 실행내에 유일한 문자열을 리턴합니다.
INITCAP	N/A (UDF)	• 문자열의 첫문자를 대문자로 나머지 문자는 소문자로 변환합니다.
INSTR	POSSTR / POSITION LOCATE	• 지정된 문자열이 문자열에서 몇번째에 위치하는지 그 위치값을 리턴합니다.
LENGTH	LENGTH	• 문자열 길이를 리턴합니다.
LENGTHB /LENGTHC	N/A (UDF)	• 다른 문자세트에 있는 문자열 길이를 리턴합니다.
LOWER	LOWER / LCASE	• 문자열의 소문자값을 리턴합니다.
LPAD(arg1,arg2,arg3)	N/A (UDF)	• arg1에 arg2의 문자길이가 될때까지 arg3를 왼쪽부터 arg1에 채웁니다 .
RPAD(arg1,arg2,arg3)	N/A (UDF)	• arg1에 arg2의 문자길이가 될때까지 arg3 오른쪽부터 arg1에 채웁니다.
LTRIM / RTRIM	LTRIM/RTRIM	• 문자열의 좌측, 우측에 공백이 있을 경우 공백을 제거합니다.
TRIM	TRIM/STRIP	• 문자열의 좌측, 우측에 공백이 있을 경우 공백을 제거합니다.
REPLACE(arg1,arg2,arg3)	REPLACE(arg1,arg2,arg3)	• arg1 문자열에서 arg2와 매치되는 문자열을 arg3로 변경합니다.
SUBSTR	SUBSTR/SUBSTRING	• 문자열의 서브 문자열을 리턴합니다.
TRANSLATE	TRANSLATE	• 문자열에서 하나이상의 문자를 다른 문자로 변환하여 리턴합니다.
TREAT	CAST	• 표현식의 선언된 타입을 변경합니다.
UPPER	UPPER /UCASE	• 모든 문자열을 대문자로 변경하여 리턴합니다.

예문


오라클

```
sql> SELECT CONCAT('ABC','CDE') FROM DUAL;
      CONCAT('ABC'
-----
      ABCCDE
sql> SELECT SUBSTR('string',1,3) FROM DUAL;
      SUBSTR
-----
      str
```

DB2

```
db2=> VALUES CONCAT('ABC','CDE')
      1
-----
      ABCCDE
db2=> VALUES SUBSTR('string',1,3)
      1
-----
      str
```


25 내장 함수 비교 - 날짜/시간 함수

Point  날짜 및 시간 함수를 비교합니다.

날짜/시간 함수 비교

오라클	DB2	
ADD_MONTHS	Date 컬럼 + 1 MONTH	• Date Argument에 정수로 된 "달"을 더하여 리턴합니다. 즉 2008년 1월 1일 + 1 일 경우 2008년 2월 1일을 리턴합니다.
CURRENT_DATE	CURRENT DATE	• 현재 날짜를 리턴합니다.
CURRENT_TIMESTAMP	CURRENT TIMESTAMP	• 현재 날짜와 시간을 리턴합니다.
SYSDATE	CURRENT DATE	• 현재 날짜를 리턴합니다.
SYSTIMESTAMP	CURRENT TIMESTAMP + CURRENT TIMEZONE	• 현재 Timestamp를 리턴합니다.
EXTRACT(datetime)	YEAR() / MONTH() / DAY() / HOUR() / MINUTE() / SECOND()	• datetime 시간으로부터 지정된 datetime 필드 값을 추출하여 리턴합니다.
LAST_DAY	N/A (UDF)	• 주어진 달의 마지막 날을 반환합니다.
NEXT_DAY	N/A (UDF)	• 시작 날짜 다음에 지정된 요일이 처음으로 나오는 날짜를 반환합니다.
ROUND	N/A (UDF)	• 월, 년도, 세기 등과 같이 선택한 날짜 파라미터를 반올림하여 리턴합니다.
TRUNC	N/A (UDF)	• 일, 월 등과 같이 지정된 날짜 파라미터를 잘라냅니다.
MONTHS_BETWEEN	N/A (UDF)	• 날짜와 날짜사이의 개월 수를 리턴합니다.

예문

오라클


```
sql> SELECT CURRENT_DATE TODAY,
        ADD_MONTHS(CURRENT_DATE,1) "NEXT"
FROM DUAL;
TODAY      NEXT
-----
28-JAN-08  28-FEB-08
```

DB2

```
db2=> VALUES ( CURRENT DATE, CURRENT DATE + 1
              MONTH)
  1          2
-----
2008-01-28  2008-02-28
```



DB2에서는 MONTH, DAY, YEAR, HOUR, MINUTE, SECOND 등 지정된 연산자가 존재하여 DATE, TIMESTAMP의 Data Type을 좀 더 용이하게 연산을 수행할 수 있습니다. 예를 들면, VALUES CURRENT DATE + 1 DAY 일 경우 현재 날짜값 + 1 day 에 대한 결과 값을 리턴합니다. 마찬가지로 VALUES CURRENT TIME + 1 HOUR 일 경우 현재 시간에 1시간을 더한 값을 리턴합니다.

Point  변환 함수를 비교합니다.

변환 함수 비교

오라클	DB2	
TO_CHAR(datetime)	TO_CHAR	• Datetime값을 지정된 형태의 문자열로 변환합니다.
TO_CHAR(datetime,'DAY')	DAYNAME	• 요일을 리턴합니다.
TO_CHAR(datetime,'MONTHD')	MONTHNAME	• 달의 이름을 리턴합니다.
TO_CHAR(number)	N/A (UDF)	• 숫자값을 지정된 형태의 문자열로 변환합니다.
TO_DATE	DATE() / FORMAT_TIMESTAMP() / TO_DATE ()	• 문자열을 Date 값으로 변환합니다.
TO_NUMBER	BIGINT() / INT() / SMALLINT() / FLOAT() / DOUBLE() / REAL() / DECIMAL () 등	• 문자열을 Oracle의 NUMBER 형태로 변환합니다. DB2에서는 숫자를 표현하는데 다양한 Data Type 이 존재하므로 해당하는 함수를 선택하여 변환합니다.
TO_TIMESTAMP	TO_DATE / TIME / TIMESTAMP / TIMESTAMP_FORMAT / TIMESTAMP_ISO	• TIMESTAMP Data Type의 값을 CHAR Data Type으로 변환합니다.
NVL	COALESCE NVL	• NULL 값을 지정된 값으로 변환합니다.

예문

오라클


```
sql> SELECT
TO_CHAR( CURRENT_TIMESTAMP,'YYYY/MM/DD')
FROM DUAL;
TO_CHAR(CURRENT_TIME
-----
2008/01/29
sql> SELECT NVL(MANAGER_NAME,'NO VALUE')
FROM EMPLOYEE;
NVL(MANAGER_ID
-----
MJAHN
NO VALUE
sql> SELECT TO_CHAR(current_timestamp,'DAY')
FROM DUAL;
TO_CHAR(current_timestamp,'DAY')
-----
TUESDAY
```

DB2

```
db2=> VALUES TO_CHAR ( CURRENT_TIMESTAMP,
'YYYY/MM/DD')
1
-----
2008/01/29
sql> SELECT COALESCE(MANAGER_NAME,'NO VALUE')
FROM EMPLOYEE
1
-----
MJAHN
NO VALUE
sql> VALUES DAYNAME(current_timestamp)
1
-----
화요일
```



27 내장 함수 비교 - 그룹 함수

Point  그룹 함수를 비교합니다.

변환 함수 비교

오라클	DB2	설명
COUNT	COUNT	• 행의 총 수를 리턴합니다.
AVG	AVG	• 평균값을 리턴합니다.
SUM	SUM	• 총 합계를 리턴합니다.
MIN	MIN	• 최소값을 리턴합니다.
MAX	MAX	• 최대값을 리턴합니다.
RANK() OVER (분석절)	RANK	• 행의 rank를 계산하여 리턴합니다.
ROW_NUMBER() OVER (ORDER BY ...)	ROW_NUMBER() OVER (ORDER BY ...)	• 1부터 시작하여 쿼리에 리턴된 각 행에 유니크한 번호를 순서적으로 리턴합니다.
STDDEV	STDDEV	• 표준편차를 리턴합니다.
VARIANCE	VARIANCE	• 분산값을 리턴합니다.

예문

오라클

```
sql> SELECT COUNT(*) FROM CUSTOMER
COUNT(*)-----
100002

sql> SELECT EMPNO,SALARY , ROW_NUMBER() OVER
( ORDER BY SALARY) FROM EMPLOYEE;
EMPNO  SALARY  ROW_NUMBER()OVER
-----
132    2100    1
128    2200    2
136    2200    3

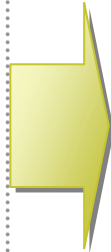
sql> SELECT DEPT_ID,
AVG(SALARY + BONUS ) AS AVG_SALARY ,
RANK() OVER ( ORDER BY AVG(SALARY +
BONUS )) AS RANK_AVG_SAL
FROM HR.EMPLOYEES
GROUP BY DEPARTMENT_ID
ORDER BY RANK_AVG_SAL;
DEPTNO  AVG_SALARY  RANK_AVG_SAL
-----
B01     95050      1
E01     80975      2
C01     77847.5    3
```

DB2

```
db2=> SELECT COUNT(*) FROM CUSTOMER
1
-----
10002

db2=> SELECT EMP_ID,SALARY, ROW_NUMBER() OVER
( ORDER BY SALARY) FROM EMPLOYEES;
EMPNO  SALARY  3
-----
132    2100    1
128    2200    2
136    2200    3

SELECT WORKDEPT, AVG(SALARY+BONUS) AS
AVG_TOTAL_SALARY,
RANK() OVER (ORDER BY AVG(SALARY+BONUS)
DESC) AS RANK_AVG_SAL
FROM EMPLOYEE
GROUP BY WORKDEPT
ORDER BY RANK_AVG_SAL
DEPTNO  AVG_SALARY  RANK_AVG_SAL
-----
B01     95050      1
E01     80975      2
C01     77847.5    3
```



Point 모든 RDBMS는 SELECT 구문을 이용하여 데이터베이스내의 데이터를 조회합니다. SELECT 구문은 SQL 언어의 가장 기본이 되면서 다양한 동작과 옵션을 가지고 있습니다. 오라클과 DB2 는 기본적으로 표준 SQL을 준수합니다. 그러나 오라클이 자체적으로 개발한 연산자가 존재하므로 이 차이점에 대해 설명하도록 합니다.

SELECT 구문의 공통점


키워드	설명
SELECT	• 조회할 컬럼명과 별명 또는 상수, DISTINCT, 함수 등이 나열되며 때에 따라서는 서브 쿼리가 위치할 수도 있습니다.
INTO	• 변수에 값을 대입합니다.
FROM	• 조회할 대상이 되는 테이블, 뷰 또는 서브 쿼리가 별명과 함께 나열되며 JOIN 문장의 조건이 기술될 수도 있습니다.
WHERE	• 조회를 위한 조건문이 기술됩니다. 이 절에 사용되는 컬럼은 FROM 절에서 선언한 테이블 혹은 뷰, 서브 쿼리에 존재해야 합니다.
GROUP BY	• 데이터를 묶어서 조회하기 위해 묶을 기준 컬럼들과 HAVING 절과 같은 그룹의 제한 조건이 기술됩니다.
ORDER BY	• 조회하는 레코드셋을 정렬하기 위한 옵션이 기술됩니다. SELECT 절에서 사용된 컬럼명, 별명 또는 순서와 정렬 옵션이 나열됩니다.

비교 연산자 비교

오라클	DB2	설명
$A = B$	$A = B$	A 와 B 값이 동일할 경우
$A < B, A > B$	$A < B, A > B$	A가 B 값보다 작을 경우, A가 B 값보다 클 경우
$A \leq B, A \geq B$	$A \leq B, A \geq B$	A가 B 값보다 작거나 같을 경우, A가 B 값보다 크거나 같을 경우
$A \lt;> B$ 또는 $A \neq B$	$A \lt;> B$	A와 B가 같지 않을 경우
AND, OR	AND, OR	AND, OR 의 Boolean 연산자
[NOT] IN (SET A)	[NOT] IN (SET A)	SET A 에 포함되거나 포함되지 않을 경우
[NOT] EXISTS (SET A)	[NOT] EXISTS (SET A)	SET A 에 포함되거나 포함되지 않을 경우
[NOT] BETWEEN A and B	[NOT] BETWEEN A and B	
[NOT] LIKE '%STRING_'	[NOT] LIKE '%STRING_'	
IS [NOT] NULL	IS [NOT] NULL	

그룹 연산자 비교

오라클	DB2	설명
UNION	UNION	중복값이 제거된 합집합 연산자
UNION ALL	UNION ALL	중복값이 제거되지 않은 합집합 연산자
MINUS	EXCEPT	차집합 연산자
INTERSECT	INTERSECT	교집합 연산자


Point  다음은 조회 쿼리문에서 DECODE 구문을 비교합니다. DB2 9.5에서는 오라클의 DECODE 구문과 동일한 DECODE 구문을 제공하므로 변환하지 않아도 됩니다.

DECODE 구문 변환

Oracle	DB2
DECODE(condition,case1,assign1,case2,assign2.. default)	CASE condition WHEN case1 THEN assign 1 WHEN case2 THEN assign 2 ELSE default END DB2 9.5에서 제공하는 DECODE 구문 : DECODE(condition,case1,assign1,case2,assign2.. default)
<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> DB2 9.5에서 제공하는 DECODE 구문 : <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>

ROWNUM 변환

구문	오라클	DB2
SELECT	<pre>SELECT * FROM TAB1 WHERE ROWNUM < 10</pre>	<pre>SELECT * FROM TAB1 FETCH FIRST 9 ROWS ONLY</pre>
DELETE	<pre>DELETE FROM TAB1 WHERE ROWNUM <= 100</pre>	9.1의 Fixpak 4일 경우 : <pre>DELETE FROM (SELECT 1 FROM tab1 FETCH FIRST 100 ROWS ONLY)</pre>
UPDATE	<pre>UPDATE TAB1 SET c1 = v1 WHERE c2 = v2 and ROWNUM <= 10</pre>	9.1의 Fixpak 4일 경우 : <pre>UPDATE (SELECT c1 FROM TAB1 WHERE c2 = v2 FETCH FIRST 10 ROWS ONLY) SET c1 = V1</pre>

Point  JOIN을 통해 여러 테이블에 분산되어 있는 데이터를 한번에 조회할 수 있습니다. 오라클과 DB2는 모두 INNER JOIN, OUTER JOIN을 모두 지원합니다. Inner join은 동일하나 Outer join은 아래와 같이 차이점이 있습니다.

OUTER JOIN

	Oracle	DB2
RIGHT OUTER JOIN	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;
LEFT OUTER JOIN	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;
FULL OUTER JOIN	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;

예문

오라클

```
sql> SELECT t1.surname
FROM EXAMPLE_TABLE1 t1,
EXAMPLE_TABLE2 t2,
EXAMPLE_TABLE3 t3,
EXAMPLE_TABLE4 t4
WHERE
((t1.emptytype = 1) OR (t1.position = 'Manager'))
AND (t1.empid = t2.empid(+))
AND (t2.empid = t3.empid(+))
AND (t2.sin = t3.sin(+))
AND (t3.jobtype(+) = 'Full-Time')
AND (t2.empid = t4.empid(+))
AND (t2.sin = t4.sin(+))
ORDER BY t1.emptytype, t2.other
```



DB2

```
db2=> SELECT t1.surname,
FROM
EXAMPLE_TABLE1 t1 LEFT OUTER JOIN
EXAMPLE_TABLE2 t2 ON (t2.empid =
t1.empid) LEFT OUTER JOIN
EXAMPLE_TABLE3 t3 ON (t3.sin = t2.sin)
AND (t3.empid = t2.empid)
AND (t3.jobtype = 'Full-Time')
LEFT OUTER JOIN
EXAMPLE_TABLE4 t4 ON (t4.sin = t2.sin)
AND (t4.empid = t2.empid)
WHERE ((t1.emptytype = 1) OR (t1.position
= 'Manager'))
ORDER BY t1.emptytype, t2.other
```

31 DB2 9.5에서 제공하는 오라클 Feature

Point DB2 9.5에서는 오라클에서 가장 많이 사용되는 함수 및 연산자들을 DB2에서 그대로 사용할 수 있는 기능을 제공합니다.

DB2 Registry 를 사용한 오라클 기능

아래 오라클 함수 및 연산자를 DB2 V9.5에서 사용하기 위해서는 DB2 Registry 변수인 DB2_COMPATIBILITY_VECTOR를 설정해야 합니다. 사용하고자 하는 기능의 HEX 값을 설정한 후 인스턴스를 재시작합니다.

오라클 / DB2	HEX 값
ROWNUM	0x01
DUAL	0x02
+ (outer 조인 연산자)	0x04
CONNECT BY	0x08

설정 방법 :

- 1 위의 오라클 기능을 그대로 사용하기 위해서는 DB2 Registry 변수인 DB2_COMPATIBILITY_VECTOR에 활성화하는 Hex 값을 설정합니다. (Hex 값은 위의 표를 참고하시기 바랍니다.)
- 2 DB2 Instance를 재시작합니다.

```
db2stop
db2set DB2_COMPATIBILITY_VECTOR=0F
db2start
```

DB2 Registry 설정없이 사용 가능한 오라클 기능


아래 오라클 함수 및 연산자는 DB2 V9.5 에서 어떤 설정없이도 사용이 가능합니다.

오라클 / DB2	설명
TO_CHAR	TO_CHAR와 TO_DATE는 DB2에서 원래 제공하던 함수이나 DB2 V9.5부터는 오라클처럼 다양한 포맷을 지원합니다.
TO_DATE	
NVL	NULL 값을 지정된 값으로 리턴합니다. DB2의 기존 함수인 COALESCE도 동일한 기능을 제공합니다.
GREATEST / LEAST	아규먼트 집합중 가장 큰 값, 가장 작은 값을 리턴합니다.
DECODE	조건에 맞는 값을 리턴합니다. WHEN ~ CASE 구문으로도 변환이 가능합니다.
BITAND / BITANDNOT / BITOR / BITXOR / BITNOT	비트 연산자 함수로 오라클과 동일한 기능입니다.
MINUS	차집합을 구하는 연산자입니다.
UNIQUE	DISTINCT와 동일한 기능을 제공합니다.
인라인 뷰의 이름 지정	오라클과 DB2는 모두 inline 뷰를 지원하나 DB2에서는 V9.5 이전에는 인라인 뷰의 이름을 지정해야 했으나 이제는 오라클처럼 지정할 필요가 없습니다.
시퀀스의 CURRVAL, NEXTVAL	시퀀스의 현재값과 다음값을 리턴합니다.



DB2 V9.5에서 사용하고자 하는 오라클 함수 또는 연산자를 각각 따로 Hex 값을 이용하여 설정이 가능합니다. 그러나 만약 혼합되어 사용할 경우, 예를 들면 CONNECT BY와 DUAL, ROWNUM 을 사용하고자 할 경우에는 0x08 + 0x02 + 0x01 = 0x0B를 설정하면 됩니다.

32 DB2 9.5에서 제공하는 오라클 Feature

Point  DB2 9.5에서는 오라클에서 가장 많이 사용되는 함수 및 연산자들을 DB2에서 그대로 사용할 수 있는 기능을 제공합니다. 다음 함수와 연산자는 DB2 Registry 변수인 DB2_COMPATIBILITY_VECTOR를 설정해야 사용 가능합니다.

DB2 9.5에서 ROWNUM 사용

오라클	DB2
SELECT * FROM TAB1 WHERE ROWNUM < 10	db2set DB2_COMPATIBILITY_VECTOR=01 SELECT * FROM TAB1 WHERE ROWNUM < 10

DB2 9.5에서 DUAL 사용

오라클	DB2
SELECT * FROM DUAL;	db2set DB2_COMPATIBILITY_VECTOR=02 SELECT * FROM DUAL;

DB2 9.5에서 CONNECT BY 사용

오라클	DB2
SELECT name, LEVEL, salary, CONNECT_BY_ROOT name AS root, SUBSTR(SYS_CONNECT_BY_PATH(name, ':'), 1, 25) AS chain FROM emp1 START WITH name = 'Goyal' CONNECT BY PRIOR empid = mgrid ORDER SIBLINGS BY salary;	db2set DB2_COMPATIBILITY_VECTOR=08 SELECT name, LEVEL, salary, CONNECT_BY_ROOT name AS root, SUBSTR(SYS_CONNECT_BY_PATH(name, ':'), 1, 25) AS chain FROM emp1 START WITH name = 'Goyal' CONNECT BY PRIOR empid = mgrid ORDER SIBLINGS BY salary;

DB2 9.5에서 (+) Join 연산자 사용

오라클	DB2
SELECT empname, deptname FROM emp, dept WHERE emp.deptid = dept.deptid (+); SELECT emp.empname, mgr.empname AS mgrname FROM emp, dept, emp as mgr WHERE emp.deptid = dept.deptid (+) AND dept.mgrid = mgr.empid (+);	db2set DB2_COMPATIBILITY_VECTOR=03 SELECT empname, deptname FROM emp, dept WHERE emp.deptid = dept.deptid (+); SELECT emp.empname, mgr.empname AS mgrname FROM emp, dept, emp as mgr WHERE emp.deptid = dept.deptid (+) AND dept.mgrid = mgr.empid (+);

33 DB2 9.5에서 제공하는 오라클 Feature

Point  다음은 DB2 Registry 변수인 DB2_COMPATIBILITY_VECTOR 설정없이도 그대로 사용가능한 오라클 함수 또는 연산자입니다.

TO_CHAR 함수

오라클	DB2
<pre>SELECT TO_CHAR(sysdate,'MM/DD/YYYY') FROM DUAL;</pre>	<pre>db2set DB2_COMPATIBILITY_VECTOR=02 SELECT TO_CHAR(CURRENT_TIMESTAMP,'MM/DD/YYYY') FROM DUAL;</pre>

TO_DATE 함수

오라클	DB2
<pre>SELECT TO_DATE('2007/07/12','YYYY/MM/DD') FROM DUAL;</pre>	<pre>db2set DB2_COMPATIBILITY_VECTOR=02 SELECT TO_DATE('2007/07/12','YYYY/MM/DD') FROM DUAL;</pre>


MINUS 연산자

오라클	DB2
<pre>SELECT * FROM T1 MINUS SELECT * FROM T2;</pre>	<pre>SELECT * FROM T1 MINUS SELECT * FROM T2;</pre>

시퀀스 연산자


오라클	DB2
<pre>SELECT SEQ.CURRVAL FROM DUAL; SELECT SEQ.NEXTVAL FROM DUAL;</pre>	<pre>SELECT SEQ.CURRVAL FROM DUAL; SELECT SEQ.NEXTVAL FROM DUAL;</pre>

34 PL/SQL과 SQL PL 비교

Point  오라클과 DB2는 모두 구조화 데이터베이스 프로그래밍 언어를 사용하여 서버 프로그램을 작성합니다. 오라클에서 사용되는 이러한 언어를 PL/SQL이라 하며 DB2에서는 SQLPL이라 합니다. 이 곳에서는 두 데이터베이스의 서버 프로그래밍 언어에 대해 비교합니다.

PL/SQL 과 SQLPL 비교

	Oracle PL/SQL	DB2 SQLPL
정의	<ul style="list-style-type: none"> PL/SQL(Procedural Language/SQL)의 약어로 오라클 DB 환경에서 실행되는 절차적인 데이터베이스 프로그래밍 언어입니다. 표준 SQL과 3세대 언어의 일부 기능을 포함한 SQL의 확장 언어로 프로시저 및 함수 등 애플리케이션 로직을 작성합니다. 	<ul style="list-style-type: none"> SQL PL(SQL Procedural Language)은 프로시저 및 함수를 작성하기 위해 SQL 과 함께 사용하는 구조화 프로그래밍 언어로 복합 SQL문 및 프로시저를 작성하는 기능을 개발자에게 제공합니다.
표준 여부	<ul style="list-style-type: none"> 표준 SQL과 표준 3세대 언어의 일부 기능을 포함한 SQL의 확장 언어. 	<ul style="list-style-type: none"> 표준 SQL/PSM(SQL Persistent Stored Modules) 언어의 서브셋
명령문	<ul style="list-style-type: none"> 모든 SQL문 변수 및 상수 등의 선언문 대입문 조건 판단문 제어 흐름문 반복 처리문 	<ul style="list-style-type: none"> 변수 관련 명령문 조건문 반복문 제어 전달문 오류 관리문 결과 세트 조작문
설명	<p>Oracle PL/SQL과 DB2 SQLPL은 모두 서버 프로그램 작성을 위한 언어로 Oracle PL/SQL 작성한 서버 프로그램을 DB2 SQLPL로 변환시킬 수 있습니다.</p> <p>그러나 DB2 SQLPL이 표준 SQL/PSM 을 따르는 반면 Oracle PL/SQL은 3세대 언어의 일부기능을 포함해 독자적으로 SQL 언어를 확장하였기 때문에 DB2 SQL PL 와 Oracle PL/SQL의 구문상 차이가 있습니다.</p>	

Point  PL/SQL과 SQLPL의 변수 선언 및 조건 구문과 흐름 제어 구문에 대해 비교합니다.

변수 선언

	오라클 PL/SQL	DB2 SQLPL
변수 선언 영역	<ul style="list-style-type: none"> Stored Procedure / Function 의 parameter list에서 선언. Stored procedure / Function / Trigger 의 Body에서 선언. Package 선언부 또는 BODY 선언부에서 선언. 	<ul style="list-style-type: none"> Stored Procedure / Function 의 Parameter list에서 선언. Stored Procedure / Function / Trigger의 Body에서 선언.
변수 값 지정 방법	<pre>l_balance NUMBER(10,2) := 0.0; l_balance := 19.99;</pre>	<pre>DECLARE l_balance NUMERIC(10,2) DEFAULT 0.0; SET l_balance = 19.99; SET 구문은 또한 다음과 같이 local 변수에 Table Column 값을 할당할 수 있습니다. SET l_balance = (SELECT balance FROM account_info WHERE account_no = actNo);</pre>

조건 구문 및 흐름 제어

오라클 PL/SQL	DB2 SQLPL
<pre>IF - THEN - END IF ; IF - THEN - ELSE - END IF ; IF - THEN - ELSIF - END IF;</pre>	<pre>IF - THEN - END IF ; IF - THEN - ELSE - END IF ; IF - THEN - ELSEIF - END IF ;</pre>
<pre>LOOP statements ; END LOOP;</pre>	<pre>[L1:] LOOP statements; LEAVE L1; END LOOP [L1];</pre>
<pre>LOOP statements; EXIT WHEN condition; END LOOP;</pre>	<pre>REPEAT statements; UNTIL condition; END REPEAT;</pre>
<pre>WHILE condition LOOP statements; END LOOP;</pre>	<pre>WHILE condition DO statements; END WHILE ;</pre>
<pre>FOR l_count IN lower_bound ..upper_bound LOOP statements; END LOOP;</pre>	<pre>SET l_count = lower_bound; WHILE l_count <= upper_bound DO statements; SET l_count = l_count + 1; END WHILE ;</pre>
<pre>OPEN cursor_variable FOR select_statements;</pre>	<pre>FOR variable AS cursor_name CORSOR FOR select_statement DO statements; END FOR ;</pre>

Point  PL/SQL과 SQLPL의 Dynamic SQL를 비교합니다.

Stored Procedure 비교

오라클


```
CREATE OR REPLACE PROCEDURE
get_emp_name_v2(emp_id IN NUMBER) AS
sql_stmt VARCHAR2(1000);
v_result VARCHAR2(20);
BEGIN
    sql_stmt := 'SELECT last_name FROM employees
WHERE emp_id = :x';
    EXECUTE IMMEDIATE sql_stmt
    INTO v_result
    USING emp_id;
    dbms_output.put_line(v_result.last_name);
END;
```



DB2

```
CREATE PROCEDURE
get_emp_name_v2 ( IN emp_id FLOAT)
LANGUAGE SQL
BEGIN
    DECLARE v_dyn_sql VARCHAR(1000);
    DECLARE v_sql_stmt STATEMENT;
    DECLARE c_employees CURSOR FOR v_sql_stmt;
    SET v_dyn_sql = 'SELECT last_name FROM employees
WHERE emp_id = '
    || CHAR(emp_id);
    PREPARE v_sql_stmt FROM v_dyn_sql;
    OPEN c_employees;
    -- FETCH ...
    CLOSE c_employees;
END!
```

37 커서 정의 비교

Point  PL/SQL과 SQLPL의 커서 정의 및 커서 사용법을 비교합니다.

커서 비교

Operation	Oracle	DB2 UDB
커서 선언	CURSOR cursor_name [(cursor_parameter(s))] IS select_statement	DECLARE cursor_name CURSOR [WITH HOLD] [WITH RETURN] [T OCALLER TO CLIENT] FOR Select_statement
커서 열기	OPEN cursor_name [(cursor_parameter(s))]	OPEN cursor_name [USING host-variable]
커서로부터 Fetch	FETCH cursor_name INTO variable(s)	FETCH [from] cursor_name INTO variable(s)
Fetch된 행 수정	UPDATE table_name SET statement(s)... WHERE CURRENT OF Cursor_name	UPDATE table_name SET statements... WHERE CURRENT OF Cursor_name
Fetch된 행 삭제	DELETE FROM table_name WHERE CURRENT OF cursor_name;	DELETE FROM table_name WHERE CURRENT OF cursor_name
커서 닫기	CLOSE cursor_name;	CLOSE cursor_name;

예문

오라클

```
PROCEDURE get_sum_projects
( v_office_id IN NUMBER ,sum_projects OUT NUMBER)
AS v_prj NUMBER(3);
CURSOR c1 IS
  SELECT current_projects FROM employees
  WHERE office_id = v_office_id;
BEGIN
  sum_projects := 0;
  OPEN c1;
  LOOP
    FETCH c1 INTO v_prj;
    EXIT WHEN c1%NOTFOUND;
    sum_projects := sum_projects + v_prj;
  END LOOP;
END;
```



DB2

```
CREATE PROCEDURE get_sum_projects( IN v_office_id
INTEGER ,OUT sum_projects INTEGER)
BEGIN
  DECLARE SQLCODE INT DEFAULT 0;
  DECLARE v_prj SMALLINT default 0;
  DECLARE v_no_data SMALLINT DEFAULT 0;
  DECLARE c1 CURSOR FOR
  SELECT current_projects FROM employees
  WHERE office_id = v_office_id;
  DECLARE CONTINUE HANDLER FOR NOT FOUND
  SET v_no_data = 1;
  SET sum_projects = 0;
  OPEN c1;
  FETCH c1 INTO v_prj;
  WHILE (v_no_data =0) DO
    SET sum_projects = sum_projects + v_prj;
    FETCH c1 INTO v_prj;
  END while;
  CLOSE c1;
END!
```

Point  PL/SQL과 SQL PL의 커서 속성을 비교합니다.


커서 비교

오라클 커서	오라클 Code	DB2 Code
%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 int var1; END; ... FETCH c1 into var1;</pre>
%NOTFOUND	<pre>OPEN cur1; LOOP FETCH cur1 INTO v_var1; EXIT WHEN cur1%NOTFOUND; END LOOP;</pre>	<pre>DECLARE SQLCODE int DEFAULT 0; OPEN c1; L1: LOOP FETCH c1 INTO v_var1; IF SQLCODE = 100 THEN LEAVE L1; END IF; ... END LOOP L1;</pre>
%FOUND	<pre>DELETE FROM emp WHERE empno = my_empno; IF SQL%FOUND THEN -- delete succeeded INSERT INTO emp_table VALUES (my_empno, my_ename);</pre>	<pre>DELETE FROM emp WHERE empno = my_empno; IF SQLCODE = 0 THEN -- delete succeeded INSERT INTO emp_table VALUES (my_empno, my_ename);</pre>

Point  PL/SQL과 SQLPL의 커서 속성을 비교합니다.

커서 비교

오라클	오라클 Code	DB2 Code
%ROWCOUNT (10행만 처리하는 경우)	<pre> LOOP FETCH c1 INTO my_ename, my_deptno; IF c1%ROWCOUNT > 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>
%ROWCOUNT (Cursor로부터 처리되 어진 Rows 수를 계산할때)	<pre> LOOP FETCH c1 INTO my_ename, my_deptno; IF c1%ROWCOUNT > 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 > 10 THEN END IF ; END LOOP L1; </pre>
%ROWCOUNT (10개의 row를 삭제한 후 수행할 작업이 있을 경우)	<pre> DELETE FROM emp_table WHERE IF SQL%ROWCOUNT > 10 THEN END IF; </pre>	<pre> DECLARE rc INT DEFAULT 0; DELETE FROM emp_table WHERE ... GET DIAGNOSTICS rc = ROW_COUNT; IF rc > 10 THEN ... END IF; </pre>

Point  Collection을 사용한 PL/SQL을 DB2 SQLPL로 변환합니다.

Nested Table 및 varray 변환

오라클 Code	DB2 Code
<pre> DECLARE TYPE EmpList IS TABLE OF emp_table.ename%TYPE ; CURSOR c1 IS SELECT emp_name FROM emp_table WHERE dept = v_dept; EmpName emp_table.ename%TYPE; empNum NUMBER; BEGIN LOOP FETCH c1 INTO EmpName; WHEN c1%NOTFOUND EXIT; empNum := empNum + 1; EmpList(empNum):= EmpName; END LOOP; CLOSE c1; END; </pre>	<pre> DECLARE GLOBAL TEMPORARY TABLE SESSION.temp_emp_list (num integer,EmpName varchar(30)) WITH REPLACE ON COMMIT PRESERVE ROWS NOT LOGGED; INSERT INTO session.temp_emp_list SELECT row_number() over(), emp_name FROM emp_table WHERE dept = v_dept; </pre>

Bulk Collect 변환

오라클 Code	DB2 Code
<pre> DECLARE TYPE EmpList IS TABLE OF emp_table.ename%TYPE ; CURSOR c1 IS SELECT emp_name FROM emp_table WHERE dept = v_dept; BEGIN OPEN c1; FETCH c1 BULK COLLECT INTO EmpList; CLOSE c1; END; </pre>	<pre> DECLARE v_empname varchar(30); DECLARE v_num INT DEFAULT 0; DECLARE GLOBAL TEMPORARY TABLE SESSION.temp_emp_list (num INTEGER, EmpName VARCHAR(30)) WITH REPLACE ON COMMIT PRESERVE ROWS NOT LOGGED; INSERT INTO SESSION.temp_emp_list (SELECT emp_name FROM emp_table WHERE detp = v_dept); </pre>

오브젝트 관련 Q/A

- 다음과 같이 테이블을 생성하는데 에러가 발생합니다. ERROR [42727] [IBM][DB2/LINUX] SQL0286N A default table space could not be found with a page size of at least "8192" that authorization ID "DB2" is authorized to use. 어떻게 해결해야 되는지요?

DB2는 오라클과는 달리 한 행의 크기가 Page (오라클의 Block에 해당) 사이즈에 꼭 들어가야 합니다. 이는 Chaining이 발생되어 성능이 떨어지는 것을 사전에 방지하기 위함입니다. 그러므로 한 행의 크기를 계산하여 이보다 큰 page size를 갖는 테이블 공간을 생성한 후 테이블 생성시 그 테이블 공간을 지정하면 해결됩니다.
- 오라클은 여러 함수와 스토어드 프로시저를 작성하여 하나의 패키지로 묶을 수 있습니다. DB2도 패키지라는 것이 존재하는 것 같은데 오라클과 동일한 용도인가요?

DB2도 패키지가 존재하나 오라클의 패키지와는 다른 개념입니다. 오라클의 패키지는 함수 및 스토어드 프로시저를 하나의 그룹으로 묶어 수행되는 오브젝트이나 DB2의 패키지는 서버 프로그램 또는 Embedded Client 프로그램 작성시 프로그램 내부의 SQL 구문의 Access Path를 서버에 저장하여 프로그램 실행시 Optimizer가 이러한 실행 계획을 읽어 실행할 수 있도록 하는 오브젝트입니다.
- 오라클은 사용자 생성시 사용자가 사용할 수 있는 테이블 스페이스를 지정할 수 있으며 또한 특정 테이블 스페이스에서 사용자가 사용가능한 크기를 제한할 수 있습니다. DB2는 어떻게 할 수 있는지요?

DB2는 특정 Tablespace 에 대한 사용 권한을 특정 사용자에게만 부여함으로써 오라클과 유사하게 제한할 수 있습니다. 다만 한 테이블 스페이스를 여러 사용자가 사용할 경우 사용자마다 사용가능한 크기를 제한할 수는 없습니다.
- 오라클에서 DB2로 Migration 하려는데 오브젝트 이름의 길이는 어떻게 되는지요?

오브젝트의 유형 및 버전에 따라 오브젝트 이름의 길이가 달라집니다. 테이블 및 인덱스 이름의 길이는 128 바이트입니다. 다른 오브젝트의 이름에 대한 길이를 알고 싶으실 경우에는 DB2 Information Center 에서 "SQL and XML limits"를 참고하시기 바랍니다.
- 오라클에서는 FBI (functional Based Index)를 지원하는데 DB2는 지원하지 않는지요? 만약 FBI 인덱스와 같은 것을 생성하려면 어떻게 해야 되는지요?

DB2에서는 FBI를 지원하지 않습니다. 만약 오라클의 FBI 와 같은 인덱스를 생성하려면 Column 을 하나 더 생성하여 Function 을 수행한 결과값을 그 Column에 삽입한 후 그 칼럼에 index를 생성하도록 합니다.



Unit 09 : Security 비교

오라클 사용자를 위한 DB2 가이드

권한 및 특권 비교

Role 비교

DB2 보안 모델



Point 데이터베이스에서 보안은 데이터베이스에 접속하려는 사용자의 인증(authentication), 특정 SQL 문 실행 및 특정 오브젝트에 대한 접근을 제어할 수 있는 권한(authorities) 및 특권(privilege), 데이터 암호화, 감사(audit) 등을 말합니다.

보안 관련 용어

- 인증 (authentication)
 - 데이터베이스에 접속하려는 사용자를 패스워드를 이용하여 확인하는 과정입니다.
- 권한 (authorities) 및 특권 (privilege)
 - 데이터베이스내의 명령문을 실행할 수 있는 능력 및 특정 오브젝트 및 데이터에 대한 접근 제어, 리소스 사용을 제어할 수 있는 권리를 권한이라 합니다.
- DCL (Data control Language)
 - 특권을 특정 사용자 또는 그룹, Role 에 부여하거나 제거하는 명령문입니다.
- Role
 - 하나 이상의 특권을 그룹화한 데이터베이스 오브젝트로 "grant" 명령문으로 사용자 및 그룹, 또는 다른 Role에 부여할 수 있습니다.
- Explicit 및 Implicit 부여 차이점
 - Explicit는 직접 사용자에게 권한/특권을 부여하는 것을 의미합니다.
 - Implicit 는 사용자가 그룹에 속함으로써, 또는 role 을 부여받음으로써 그룹 또는 Role 에 권한/특권이 부여될때 자동으로 동일한 권한/특권을 부여받는 것을 의미합니다.

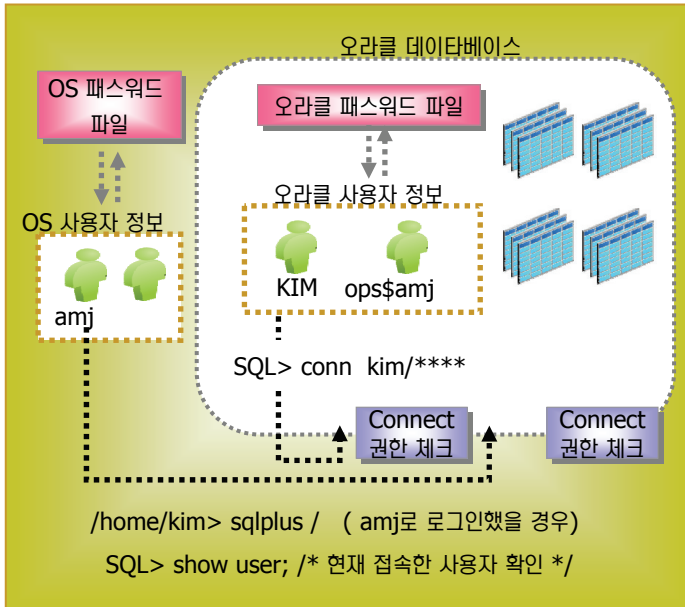
보안 비교

	오라클	DB2
인증 (Authentication)	<ul style="list-style-type: none"> • 데이터베이스 내의 사용자 정보 또는 OS 사용자를 이용하여 인증합니다. 	<ul style="list-style-type: none"> • OS 사용자 또는 third 보안 제품을 통해 인증합니다.
인스턴스 관리 권한	<ul style="list-style-type: none"> • Administrative 특권은 인스턴스 및 데이터베이스 관리작업을 수행할 수 있는 권한입니다. 	<ul style="list-style-type: none"> • 인스턴스 권한은 인스턴스 및 인스턴스 내의 데이터베이스를 관리할 수 있는 권한입니다.
특권(Privilege)	<ul style="list-style-type: none"> • 오라클의 특권은 스키마에 대한 특정 작업을 수행할 수 있는 시스템 특권과 오브젝트에 액세스 및 DDL 문을 수행할 수 있는 오브젝트 특권으로 나뉩니다. 	<ul style="list-style-type: none"> • 데이터베이스 내에서 특정 명령문을 수행할 수 있는 데이터베이스 권한과 오브젝트에 액세스 및 DDL문을 수행할 수 있는 특권으로 나뉘어 집니다.
DCL	<ul style="list-style-type: none"> • grant/revoke 명령문을 이용하여 특정 사용자, 그룹, Role에 특권을 부여/해지 합니다. 	<ul style="list-style-type: none"> • grant/revoke 명령문을 이용하여 특정 사용자, 그룹, Role에 특권을 부여/해지 합니다.
Role	<ul style="list-style-type: none"> • 특권을 그룹화하여 특정 사용자,그룹,다른 Role에 부여할 수 있습니다. • create role < Role 이름> ~을 이용하여 생성합니다. 	<ul style="list-style-type: none"> • 특권을 그룹화하여 특정 사용자,그룹,다른 Role에 부여할 수 있습니다. • create role < Role 이름> 을 이용하여 생성합니다.
암호화	<ul style="list-style-type: none"> • Oracle Advanced Security 옵션을 사용하여 데이터 암호화, 백업 이미지 암호화등 보안관련 기능을 이용할 수 있습니다. 	<ul style="list-style-type: none"> • ENCRYPT 빌트인 함수를 이용하여 데이터를 암호화합니다. • DB2 Encryption Expert 를 이용하여 데이터 암호화 및 백업 이미지 암호화, 암호화된 데이터에 대한 인덱스 기능등 다양한 기능을 이용할 수 있습니다.
LBAC (Label Based Access)	<ul style="list-style-type: none"> • Label 기반의 액세스 컨트롤을 지원함으로써 각 행 및 컬럼에 대한 접근을 제어할 수 있습니다. 	<ul style="list-style-type: none"> • Label 기반의 액세스 컨트롤을 지원함으로써 각 행 및 컬럼에 대한 접근을 제어할 수 있습니다.
Audit	<ul style="list-style-type: none"> • AUDIT_TRAIL 파라미터를 설정함으로써 데이터베이스 활동들에 대해 모니터링하고 감사 레코드를 생성합니다. 	<ul style="list-style-type: none"> • db2audit 명령어(무료)를 이용하여 데이터베이스 이벤트에 대해 모니터링하고 감사 레코드를 생성합니다.

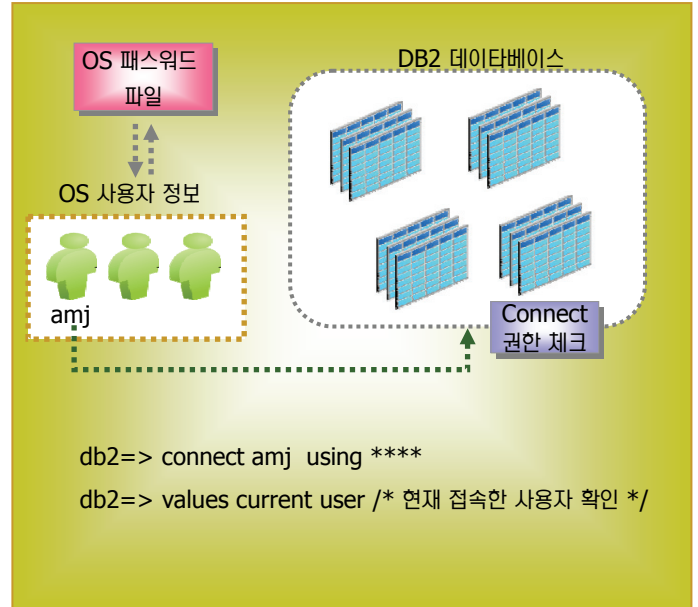
02 사용자 인증 비교

Point 데이터베이스를 접속하기 위해서는 접속하는 사용자가 누구인지를 확인합니다. 이를 사용자 인증 (Authentication) 과정이라 합니다. 오라클에서는 외부에 있는 사용자 정보 혹은 내부에서 생성한 사용자 정보를 이용하여 인증합니다. 그러나 DB2에서는 데이터베이스 외부에 있는 사용자 정보를 이용합니다.

오라클 권한



DB2 권한



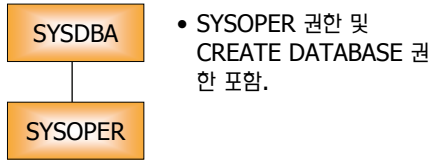
- 데이터베이스 내부 사용자를 이용한 인증.
 - 데이터베이스내에 사용자를 생성하며 이를 이용하여 인증합니다. 이때 사용자 패스워드는 오라클 내부에 저장 및 관리합니다.
- 운영체제에 의한 사용자 인증.
 - 운영체제의 사용자를 이용하여 인증합니다. 그러나 운영체제의 사용자와 동일한 사용자를 오라클내에서 생성해야 하며 이때 패스워드는 OS 패스워드 파일에서 저장 및 관리합니다.
- Third-party 제품을 이용한 인증
 - Kerberos와 같은 제품을 이용하여 원격 사용자를 인증합니다.

- 운영체제 및 플러그인 보안체계를 이용한 사용자 인증.
 - 운영체제 또는 운영체제에 플러그인의 보안체계를 이용하여 사용자를 인증합니다. 오라클과는 달리 데이터베이스내에 별도의 사용자를 생성할 필요가 없습니다.
- Third-party 제품을 이용한 인증
 - Kerberos와 같은 제품을 이용하여 사용자를 인증합니다.

Point 데이터베이스내의 명령문을 실행할 수 있는 능력 및 특정 오브젝트 및 데이터에 대한 액세스 제어, 리소스 사용에 대한 제어할 수 있는 권리를 권한 및 특권이라 합니다. 오라클과 DB2는 오브젝트 유형에 따른 특권이 유사합니다.

오라클 권한

관리 특권



시스템 특권

테이블공간	테이블	인덱스	시퀀스
<ul style="list-style-type: none"> CREATE ALTER DROP MANAGE UNLIMITED 	<ul style="list-style-type: none"> CREATE DROP ALTER INSERT DELETE BACKUP LOCK 	<ul style="list-style-type: none"> CREATE ALTER DROP 	<ul style="list-style-type: none"> CREATE ALTER DROP SELECT
프로시저	사용자	PROFILE	
<ul style="list-style-type: none"> CREATE ALTER DROP EXECUTE 	<ul style="list-style-type: none"> CREATE ALTER DROP 	<ul style="list-style-type: none"> CREATE ALTER DROP 	

오브젝트 특권

테이블	인덱스	시퀀스	프로시저
<ul style="list-style-type: none"> DELETE INSERT SELECT UPDATE ALTER INDEX REFERENCES 	<ul style="list-style-type: none"> CREATE ALTER DROP 	<ul style="list-style-type: none"> ALTER SELECT 	<ul style="list-style-type: none"> EXECUTE

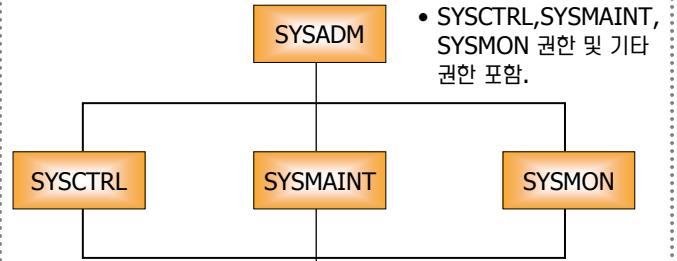
예문

```

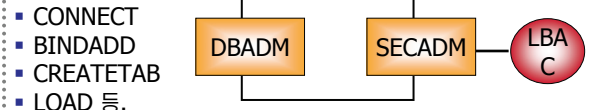
SQL> GRANT SELECT,INSERT,UPDATE,DELETE ON
customers TO AMJ WITH ADMIN OPTION;
SQL> GRANT SELECT ON customers TO public;
SQL> GRANT SELECT ON AMJ.CUST_SEQ TO hr;
  
```

DB2 권한

인스턴스 권한



데이터베이스 권한



- CONNECT
- BINDADD
- CREATETAB
- LOAD 등.

오브젝트 권한

테이블공간	스키마	테이블	인덱스
<ul style="list-style-type: none"> USE 	<ul style="list-style-type: none"> ALTERIN CREATEIN DROPIN 	<ul style="list-style-type: none"> DELETE INSERT SELECT UPDATE ALTER CONTROL INDEX REFERENCES 	<ul style="list-style-type: none"> CONTROL
Alias	시퀀스	루틴	패키지
<ul style="list-style-type: none"> USE 	<ul style="list-style-type: none"> USAGE ALTER 	<ul style="list-style-type: none"> EXECUTE 	<ul style="list-style-type: none"> BIND CONTROL EXECUTE

예문

```

db2=> GRANT SELECT,INSERT,UPDATE,DELETE ON
customers TO AMJ WITH GRANT OPTION
db2=> GRANT SELECT ON customers TO public
db2=> GRANT USAGE ON AMJ.CUST_SEQ TO hr
  
```




오라클과 DB2 권한 및 특권의 종류와 제어 방법은 대상에 따라 달라집니다. 다음은 인스턴스 권한과 데이터베이스 권한, 오브젝트 권한을 비교한 표입니다.

인스턴스 권한 비교

	오라클	DB2
인스턴스 관리 권한	<ul style="list-style-type: none"> 인스턴스와 데이터베이스를 관리하는 SYSDBA와 SYSOPER 특권이 존재합니다. 	<ul style="list-style-type: none"> 인스턴스 및 인스턴스 내에 존재하는 데이터베이스를 제어하는 SYSADM, SYSCTRL, SYSMANT, SYSMON 권한이 존재합니다.
제어 방법	<ul style="list-style-type: none"> 다음과 같이 OS 인증과 자체 패스워드 인증에 따라 달라집니다. <ul style="list-style-type: none"> OS 인증일 경우 오라클 설치시 지정한 OSDBA 그룹에 속하는 사용자는 SYSDBA 특권으로 데이터베이스에 접속이 가능하며 OSOPER 그룹에 속하는 사용자는 SYSOPER 특권으로 데이터베이스에 접속이 가능합니다. 패스워드 인증일 경우 사용자와 패스워드를 이용하여 SYSDBA, SYSOPER 특권으로 데이터베이스에 접속할 수 있습니다. 데이터베이스 생성시 자동으로 생성되는 SYS 사용자에게 SYSOPER 특권이 부여되며 SYSTEM 사용자에게 SYSOPER 특권이 부여됩니다. 	<ul style="list-style-type: none"> 인스턴스 생성시 인스턴스 소유자인 사용자 및 동일 그룹에 속하는 사용자에게 SYSADM이 부여되며 database manager configuration 파일을 수정하여 SYSADM, SYSCTRL, SYSMANT 권한을 부여할 수 있습니다.
권한 설명	<ul style="list-style-type: none"> SYSDBA 및 SYSOPER 은 인스턴스 시작과 종지를 수행할 수 있습니다. SYSDBA는 SYSOPER 특권을 포함하여 데이터베이스를 생성 및 삭제할 수 있는 등 보다 많은 특권을 가지고 있습니다. 	<ul style="list-style-type: none"> SYSADM, SYSCTRL, SYSMANT 는 모두 인스턴스 시작과 종지를 수행할 수 있습니다. SYSADM > SYSCTRL > SYSMANT 순으로 권한이 높습니다. 자세한 사항은 뒤에서 다루도록 합니다.

데이터베이스 권한 및 특권 비교

	오라클	DB2
데이터베이스 권한	<ul style="list-style-type: none"> 데이터베이스 생성시 자동으로 생성되는 DBA라는 Role을 통하여 데이터베이스 시스템 특권들이 부여됩니다. 	<ul style="list-style-type: none"> DBADM 이라는 특권을 통해 데이터베이스에 속하는 여러가지 특권을 부여받을 수 있습니다.
제어 방법	<ul style="list-style-type: none"> grant/revoke 명령어를 통해 특권 부여 및 해지합니다. 	
권한 설명	<ul style="list-style-type: none"> 대부분의 시스템 특권이 포함되어 있는 Role이며 Administrator가 아닌 사용자에게는 부여할 수 없습니다. 	<ul style="list-style-type: none"> SECADM 을 제외한 데이터베이스의 특권들, 즉 LOAD, CONNECT, BINDADD 등이 모두 부여되며 데이터베이스 내에 존재하는 모든 오브젝트들의 모든 특권들이 부여됩니다.

오브젝트 특권 비교

	오라클	DB2
오브젝트 특권	<ul style="list-style-type: none"> 각 오브젝트마다 특권 종류가 달라지나 동일한 오브젝트일 경우 특권의 종류도 유사합니다. 	
제어 방법	<ul style="list-style-type: none"> grant/revoke 명령어를 통해 특권 부여 및 해지합니다. 	



오라클과 DB2 권한 및 특권의 종류와 제어 방법은 대상에 따라 달라집니다. 다음은 인스턴스 권한과 데이터베이스 권한, 오브젝트 권한을 비교한 표입니다.

권한 제어 방법 비교

	오라클	DB2
인스턴스 권한 제어 방법	<ul style="list-style-type: none"> OS 인증 사용시 SYSDBA 특권을 부여할 사용자를 OSDBA 그룹에 포함시킵니다. SYSOPER 특권을 부여할 사용자를 OSOPER 그룹에 포함시킵니다. 	<ul style="list-style-type: none"> database manager configuration 의 파라미터인 SYSADM, SYSCTRL, SYMAINT, SYSMON 을 해당 OS 그룹을 지정하면 각 그룹에 속하는 사용자가 권한을 부여받게 됩니다.
부여 대상자	<ul style="list-style-type: none"> 특정 사용자, Role, PUBLIC에게 권한 또는 특권을 부여할 수 있습니다. 	<ul style="list-style-type: none"> 특정 사용자, 그룹, Role, PUBLIC에게 권한 또는 특권을 부여할 수 있습니다. 오라클과는 달리 OS 그룹에게 권한 및 특권을 부여할 수 있습니다.
부여 방법	<ul style="list-style-type: none"> GRANT [특권 role] TO [user role PUBLIC] WITH ADMIN OPTION * 자세한 grant 구문은 매뉴얼을 참고하시기 바랍니다. 	<ul style="list-style-type: none"> GRANT [특권 role] TO [user group role PUBLIC] WITH GRANT OPTION * 자세한 grant 구문은 매뉴얼을 참고하시기 바랍니다.
해지 방법	<ul style="list-style-type: none"> REVOKE <특권> FROM [user role PUBLIC] * 자세한 revoke 구문은 매뉴얼을 참고하시기 바랍니다. 	<ul style="list-style-type: none"> REVOKE 특권 FROM TO [user group role PUBLIC] * 자세한 revoke 구문은 매뉴얼을 참고하시기 바랍니다.
기타 설명	<ul style="list-style-type: none"> WITH ADMIN OPTION은 부여받은 사용자가 또 다른 사용자에게 동일한 특권을 부여할 수 있는 권한을 주는 옵션입니다. 	<ul style="list-style-type: none"> WITH GRANT OPTION은 오라클의 WITH ADMIN OPTION과 동일한 옵션입니다.

권한/특권 조회 방법 비교

	오라클	DB2
테이블 권한	<ul style="list-style-type: none"> ALL_TAB_PRIVS DBA_TAB_PRIVS USER_TAB_PRIVS ALL_COL_PRIVS DBA_COL_PRIVS USER_COL_PRIVS DBA_SYS_PRIVS USER_SYS_PRIVS SESSION_PRIVS 	<ul style="list-style-type: none"> SYSIBMADM.PRIVILEGES SYSIBMADM.AUTHORIZATIONIDS SYSIBMADM.OBJECTOWNERS SYSCAT.DBAUTH SYSCAT.TABUATH SYSCAT.COLAUTH SYSCAT.INDEXAUTH SYSCAT.SCHEMAAUTH SYSCAT.SEQUENCEAUTH
예제	<pre>SQL> SELECT * FROM DBA_TAB_PRIVS WHERE TABLE_NAME = 'CUSTOMER'; SQL> SELECT * FROM DBA_SYS_PRIVS WHERE GRANTEE = 'JOHN';</pre>	<pre>db2=> SELECT * FROM SYSCAT.TABAUTH WJERE TABNAME = 'CUSTOMER' db2=> SELECT * FROM SYSIBMADM.PRIVILEGES WHERE AUTHID = 'JOHN'</pre>



Role 은 특권들의 집합으로 특권의 관리가 편리해집니다. 오라클과 DB2 9.5는 Role을 지원함으로써 특권들을 편리하게 관리할 수 있습니다.

Role 비교

	오라클	DB2
Role	<ul style="list-style-type: none"> 어떠한 사용자도 Role을 소유하지 않습니다. Role이 처음 생성될 때 Role과 결합된 특권은 존재하지 않습니다. Role의 CREATE 특권 및 DROP ANY 특권을 부여받은 사용자가 Role을 생성 및 제거할 수 있습니다. 사용자와 유사하게 패스워드를 두어 Role을 활성화/비활성화할 수 있습니다. CONNECT, RESOURCE, DBA 등과 같이 미리 정의된 Role 이 존재합니다. 	<ul style="list-style-type: none"> 어떠한 사용자도 Role을 소유하지 않습니다. Role이 처음 생성될 때 Role과 결합된 특권은 존재하지 않습니다. SECADM 권한을 가진 사용자만이 Role을 생성 및 제거할 수 있습니다. 미리 정의된 Role은 없습니다. 그러나 오라클의 CONNECT, DBA 는 DB2에서는 Role이 아닌 데이터베이스 권한이므로 권한을 부여하면 됩니다.
Role 생성 방법	<ul style="list-style-type: none"> 사용자와 동일한 이름을 가진 Role을 생성할 수 없습니다. 구문 : SQL> CREATE ROLE <Role Name> [IDENTIFIED BY <패스워드>] 	<ul style="list-style-type: none"> 사용자와 동일한 이름을 가진 Role을 생성할 수 있습니다. 구문 : db2=> CREATE ROLE <Role Name>
Role 부여 방법	<ul style="list-style-type: none"> GRANT [특권 role] TO [user role PUBLIC] [WITH ADMIN OPTION] * 자세한 grant 구문은 매뉴얼을 참고하시기 바랍니다. 	<ul style="list-style-type: none"> GRANT [특권 role] TO [user group role PUBLIC] [WITH GRANT OPTION] * 자세한 grant 구문은 매뉴얼을 참고하시기 바랍니다.
Role 해지 방법	<ul style="list-style-type: none"> REVOKE <특권> FROM [user role PUBLIC] * 자세한 revoke 구문은 매뉴얼을 참고하시기 바랍니다. 	<ul style="list-style-type: none"> REVOKE 특권 FROM TO [user group role PUBLIC] * 자세한 revoke 구문은 매뉴얼을 참고하시기 바랍니다.
Role 정보 테이블	<ul style="list-style-type: none"> DBA_ROLES SESSION_ROLES DBA_ROLE_PRIVS ROLE_ROLE_PRIVS ROLE_SYS_PRIVS 	<ul style="list-style-type: none"> SYSCAT.ROLES SYSCAT.ROLEAUTH SYSIBM.SYSROLES SYSIBM.SYSROLEAUTH SYSIBMADM.PRIVILEGES
SET ROLE 명령어	<ul style="list-style-type: none"> "SET ROLE" 명령어를 이용하여 ROLE 을 활성화 또는 비활성화합니다. 예) SQL> SET ROLE EDITOR SET ROLE < Role Name> IDENTIFIED BY <패스워드> 	<ul style="list-style-type: none"> "SET ROLE" 명령어를 이용하여 현재 접속한 사용자가 특정 Role에 대한 권한이 있는지 확인합니다. 예) db2=> SET ROLE EDITOR <ul style="list-style-type: none"> 현재 접속한 사용자가 EDITOR라는 Role에 권한이 없을 경우 예러가 발생하며 권한이 있을 경우에는 어떠한 메시지도 발생하지 않습니다.
<ul style="list-style-type: none"> 오라클과 DB2의 SET ROLE 명령어는 기능이 다르므로 이 점 유의하시기 바랍니다. 		

예문

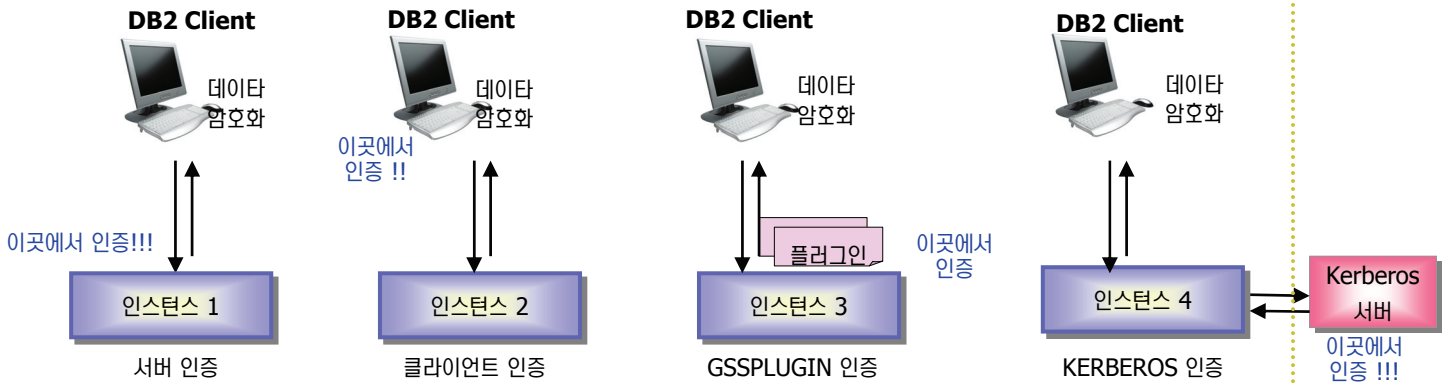
오라클	DB2
<pre>SQL> CREATE ROLE EDITOR IDENTIFIED BY EDITOR; SQL> GRANT SELECT ON TABLE TEST.T1 TO EDITOR; SQL> GRANT EDITOR TO AMJ; SQL> SELECT ROLE, PRIVILEGE FROM ROLE_SYS_PRIVS WHERE ROLE = 'EDITOR'; SQL> SELECT * FROM DBA_ROLE_PRIVS; SQL> REVOKE EDITOR FROM AMJ;</pre>	<pre>db2=> CREATE ROLE EDITOR db2=> GRANT SELECT ON TABLE TEST.T1 TO ROLE EDITOR db2=> GRANT EDITOR TO AMJ; db2=> SELECT * FROM SYSIBMADM.PRIVILEGES WHERE AUTHID = 'EDITOR'; db2=> SELECT * FROM SYSCAT.TABAUTH WHERE GRANTEE = 'EDITOR' db2=> REVOKE ROLE EDITOR FROM AMJ</pre>

Point 다음은 DB2 보안 모델입니다. 인증을 어디에서 할 것인지 지정하는 방법과 권한 부여 방법에 대해서도 설명합니다.

인증 (AUTHENTICATION)

인증 변경 방법 :

```
db2=> update dbm cfg using AUTHENTICATION <SERVER | CLIENT | ...>
```



권한 (AUTHORIZATION)

인스턴스 권한 부여 방법 :

```
db2=> update dbm cfg using SYSADM <OS Group Name>
db2=> update dbm cfg using SYSCTRL <OS Group Name>
```

데이터베이스 권한 부여 방법 :

```
db2=> grant dbadm on database to AMJ
```

인스턴스 권한

- SYSADM
- SYSCTRL
- SYSMANT
- SYSMON

데이터베이스 권한


- DBADM
- SECADM
- CONNECT
- BINDADD
- LOAD
- CREATETAB
- IMPLICIT_SCHEMA
- QUIESCE_CONNECT
- CREATE_EXTERNAL_ROUTINE
- CREATE_NOT_FENCED_ROUTINE

특권 (Object Privileges)

특권 부여 방법 :

```
db2=> GRANT SELECT,INSERT,UPDATE,DELETE ON customers TO AMJ WITH GRANT OPTION
```



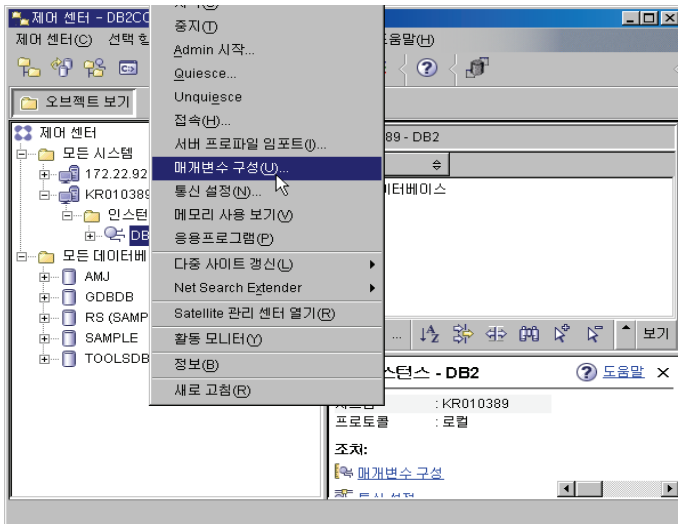
Point  다음은 인스턴스 권한과 데이터베이스 권한이 수행할 수 있는 기능에 대해 요약한 표입니다.

기능	SYSADM	SYSCTRL	SYSMAINT	SYSMON	DBADM	SECADM
Migrate Database	가능					
Update DBM CFG	가능					
Establish/Change SYSCTRL and SYSMAINT	가능					
Grant / Revoke DBADM	가능					
Quiesce Instance	가능					
Update db / node DPF / dcs Directories	가능	가능				
Force users off the Database	가능	가능				
Create / Drop Database	가능	가능				
Create / Drop / Alter Table Space	가능	가능				
Restore to new Database	가능	가능				
Update DB CFG	가능	가능	가능			
Backup Database / Table Space	가능	가능	가능			
Perform roll forward recovery	가능	가능	가능			
Start / Stop Instance	가능	가능	가능			
Restore Table Space	가능	가능	가능			
Run Trace	가능	가능	가능			
Take Monitor Snapshots	가능	가능	가능	가능		
Query Table Space states	가능	가능	가능		가능	
Update / Prune Log History Files	가능	가능	가능		가능	
Quiesce Table Space	가능	가능	가능		가능	
Reorgs & Runstats	가능	가능	가능		가능	
Quiesce database	가능				가능	
Load Tables	가능				가능	
READ LOG FILES	가능				가능	
Create / Drop Activity or Event Monitors	가능				가능	
Create / Drop Security Label components, policies 및 labels						가능
LBAC rules 및 set USERSESSION						가능

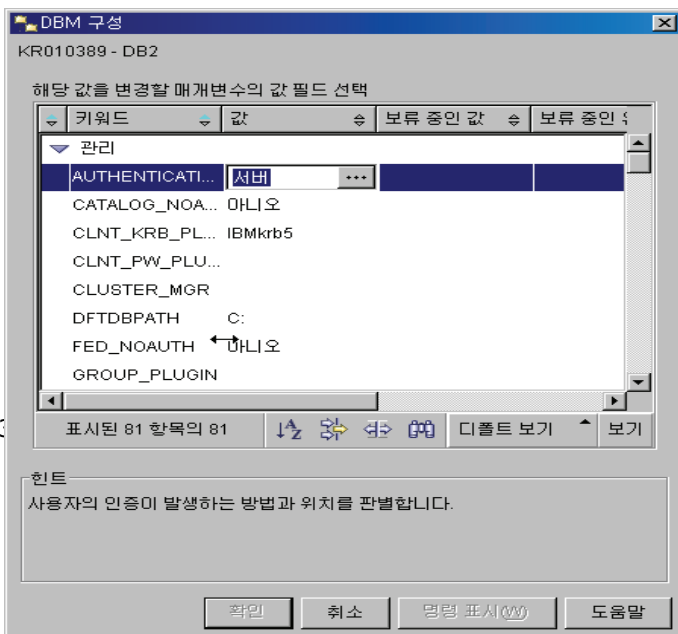
09 제어센타를 이용한 인증 방법 설정

Point 제어센타를 이용하여 인증 방법을 설정합니다.

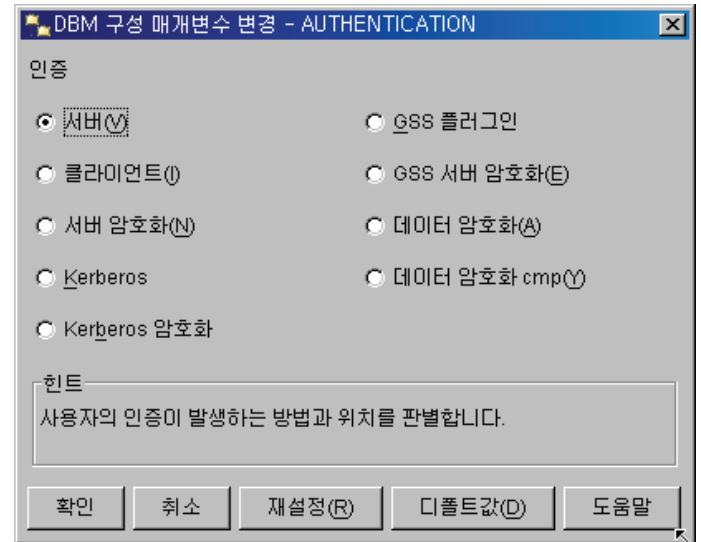
1. 시작 => 모든 프로그램 => IBM DB2 => DB2COPY1(디폴트) => 일반 관리 도구 => 제어센타를 실행합니다. 아래와 같이 제어센타가 나타나면 모든 시스템 => <시스템 이름 >=> 인스턴스 => DB2를 선택한 후 오른쪽 버튼을 클릭하여 "매개변수"를 선택합니다.



2. 관리 밑에 있는 AUTHENTICATION 매개변수에서 값을 선택합니다.



4. 아래와 같은 창이 나타나면 인증방법을 선택합니다. 디폴트는 서버이며 이는 데이터베이스를 원격에서 접속할 경우 원격 서버의 사용자와 비밀번호를 제공해야 함을 의미합니다.

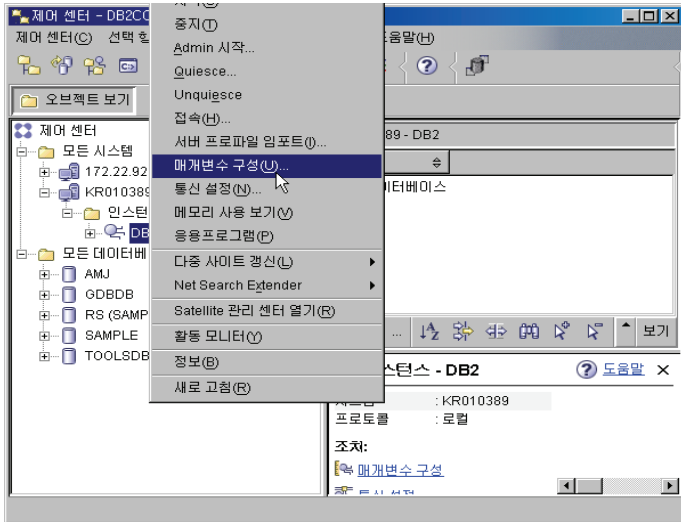


5. 확인버튼을 눌러 인증 방법을 변경합니다. 변경된 인증 방법은 인스턴스를 재시작해야 적용됩니다.

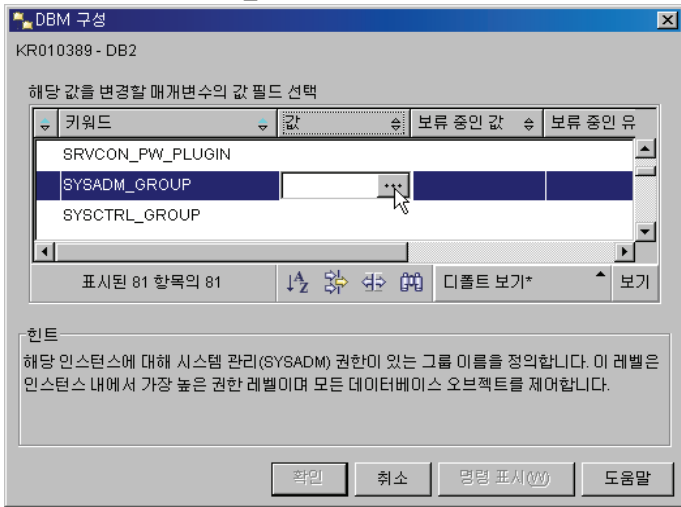
10 제어센타를 이용한 권한 및 특권 부여

Point 제어센타를 이용하여 권한 및 특권을 부여합니다.

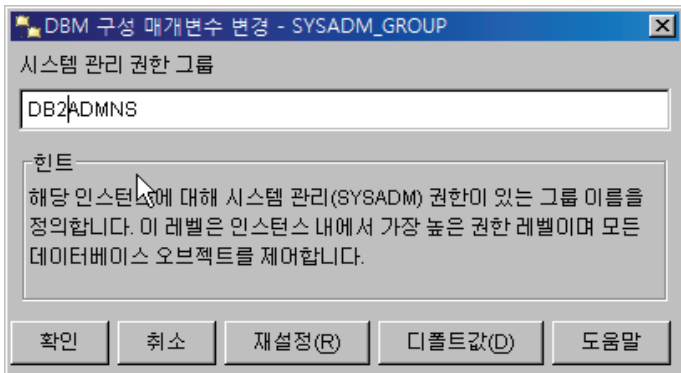
- 제어센타에서 모든 시스템 => <시스템 이름> => 인스턴스 => DB2 를 선택한후 오른쪽 버튼을 클릭하여 "매개변수 구성"을 선택합니다.



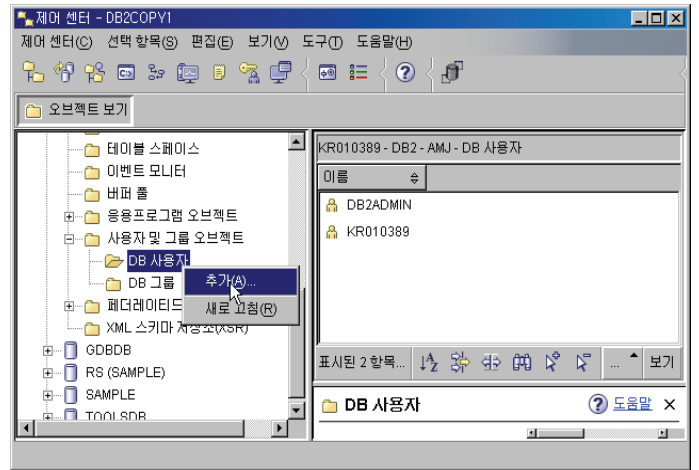
- 관리 유형에서 SYSADM_GROUP을 선택하고 값을 클릭합니다.



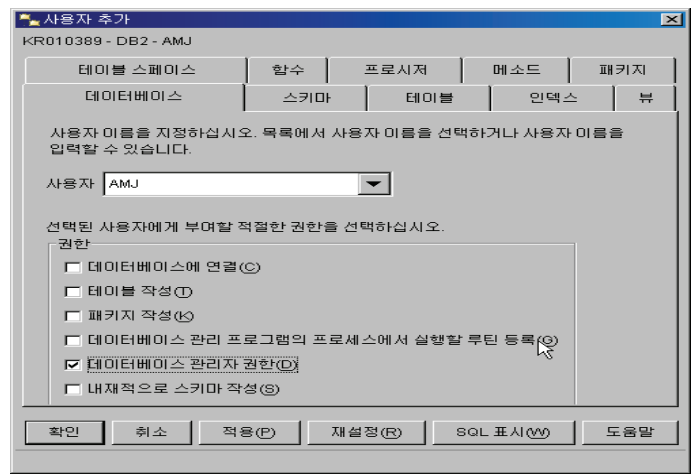
- 시스템 관리 권한 그룹을 지정하면 그 그룹에 속하는 사용자가 SYSADM 권한을 부여받게 됩니다. 관리 그룹은 DBADMNS를 지정합니다. DBADMNS는 반드시 윈도우의 OS 그룹으로 생성되어 있어야 합니다.



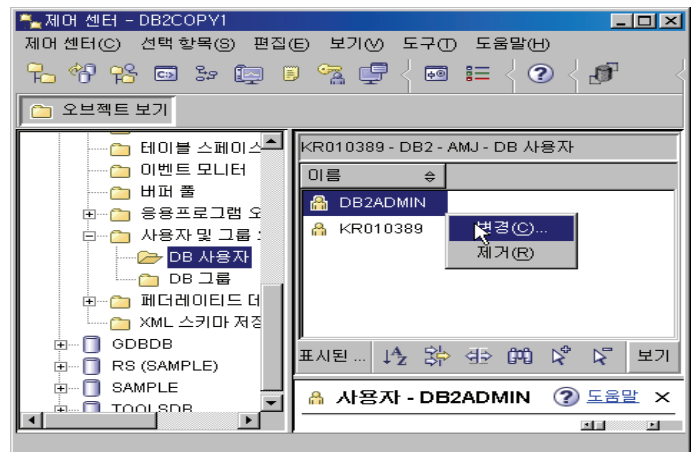
- 제어센타에서 모든 데이터베이스 하위에 있는 "SAMPLE" 데이터베이스를 펼칩니다. 사용자 및 그룹 오브젝트 하위에 있는 DB 사용자를 클릭하여 오른쪽 마우스를 클릭한 후 추가버튼을 클릭합니다.



- 원하는 사용자를 선택한 후 부여할 권한을 클릭합니다. 만약 테이블 오브젝트에 대한 특권을 부여할 경우 테이블 탭을 선택한 후 부여할 특권을 선택합니다. 이 때 사용자는 OS 사용자여야 합니다.



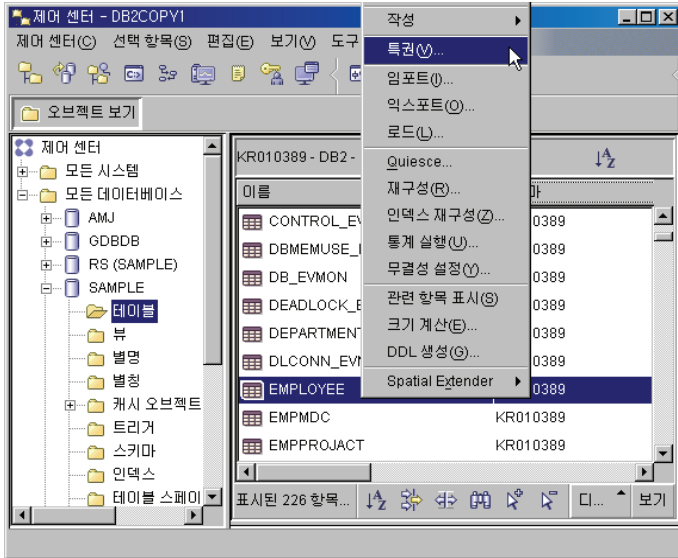
- 만약 기존 사용자의 특권을 수정할 경우 오른쪽 패널에 있는 사용자를 선택한 후 변경을 클릭한 후 5번과 동일하게 하시면 됩니다.



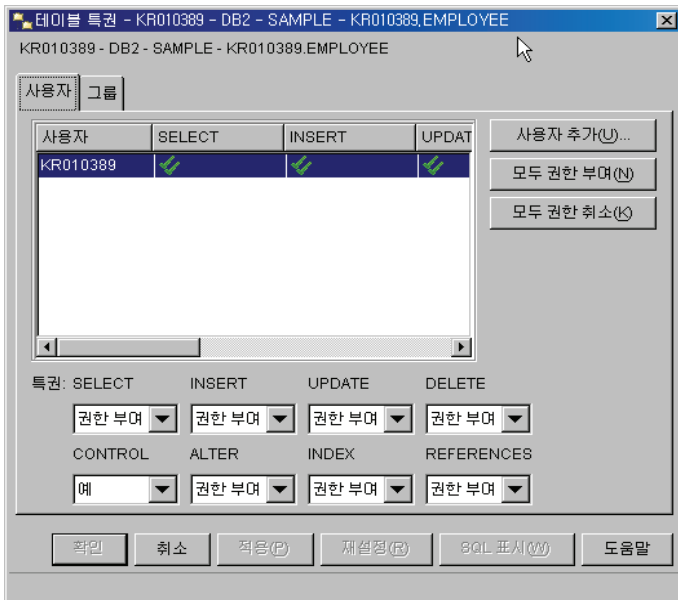
11 제어센타를 이용한 권한 및 특권 부여

Point 제어센타를 이용하여 권한 및 특권을 부여합니다.

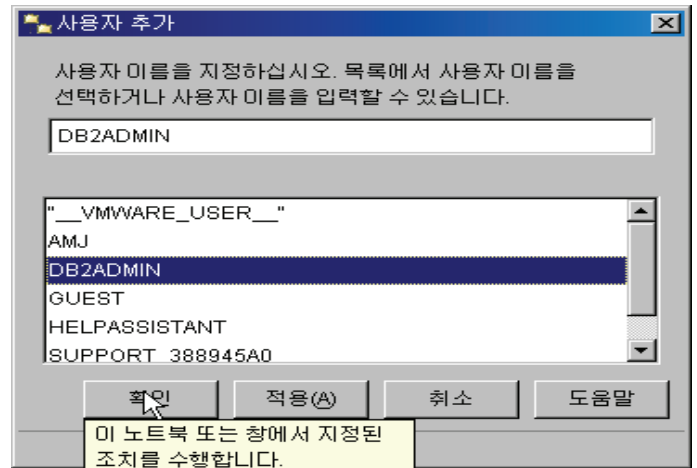
- 제어센타에서 특권을 부여하거나 변경할 오브젝트를 선택한 후 오른쪽 버튼을 눌러 특권을 클릭합니다. 이 곳에서는 EMPLOYEE의 특권을 변경하도록 합니다.



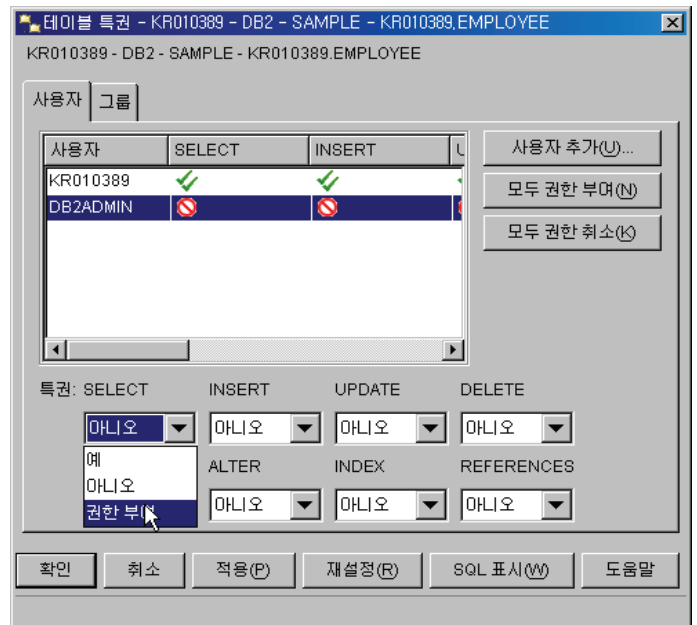
- 새로운 사용자에게 EMPLOYEE 테이블에 대한 특권을 부여할 경우 사용자 추가버튼을 클릭합니다.



- 특권을 부여할 사용자 이름을 선택한 후 확인버튼을 클릭합니다.



- 사용자를 선택하고 SELECT 특권을 부여합니다. 권한 부여를 선택하면 SELECT 특권을 부여받을 뿐 아니라 다른 사람에게 부여할 수 있는 특권도 부여받게 됩니다.



- 확인버튼을 클릭합니다.

@ Security 관련 Q/A

- ❶ 오라클은 "CREATE USER~" 라는 명령어를 이용하여 사용자를 생성하여 접속합니다. DB2는 인증과 권한이 어떻게 이루어지며 패스워드 파일은 어떻게 관리 되나요?

DB2는 사용자를 데이터베이스내에서 따로 생성하는 부분은 없고 OS 사용자를 사용하여 접속하고 권한을 부여합니다. DB2 외부에서 정의된 사용자를 사용하지 않고 패스워드 파일 또한 존재하지 않기 때문에 관리할 필요가 없습니다.
- ❷ 오라클은 Resource와 패스워드 관리를 위해 profile을 생성합니다. 그리고 정의된 profile을 각 사용자에게 지정하여 사용자마다 Resource를 달리 지정할 수 있으며 패스워드도 달리 관리할 수 있습니다. DB2는 어떤지요?

DB2는 패스워드를 따로 관리하지 않으므로 패스워드에 따른 파라미터를 관리할 필요는 없습니다. Resource 부분은 Database Manager ⇒ Configuration 에서 파라미터를 지정함으로써 가능하나 사용자별로 달리 지정하지는 않습니다.
- ❸ 오라클은 사용자 생성시 사용자가 사용할 수 있는 테이블 스페이스를 지정할 수 있으며 또한 특정 테이블 스페이스에서 사용자가 사용가능한 크기를 제한할 수 있습니다. DB2는 어떻게 할 수 있는지요?

DB2는 특정 Tablespace에 대한 사용 권한을 특정 사용자에게만 부여함으로써 오라클과 유사하게 제한할 수 있습니다. 다만 한 테이블 스페이스를 여러 사용자가 사용할 경우 사용자마다 사용가능한 크기를 제한할 수는 없습니다.
- ❹ 오라클의 Role과 DB2의 Role은 동일한건가요?

오라클의 Role과 DB2의 Role의 역할은 동일하며 부여/해지 방법 역시 거의 유사합니다.
- ❺ 오라클은 CONNECT라는 미리 정의된 ROLE 을 사용자에게 부여합니다. DB2는 CONNECT 라는 Role 이 존재하는지요? 만약 존재하지 않는다면 어떻게 Connect 권한을 사용자에게 부여할 수 있는지요?

DB2는 오라클과 달리 미리 정의된 Role은 없습니다. 그러나 오라클의 사전에 정의된 Role 들 , 예를 들면 CONNECT, DBA, IMP_FULL_DATABASE 의 Role에 해당하는 데이터베이스 권한은 존재합니다. 즉 데이터베이스 권한으로 CONNECT, DBADM, LOAD 등이 존재하여 이 권한을 사용자에게 부여하면 됩니다.



Unit 10 : 데이터 이동 비교

오라클 사용자를 위한 DB2 가이드

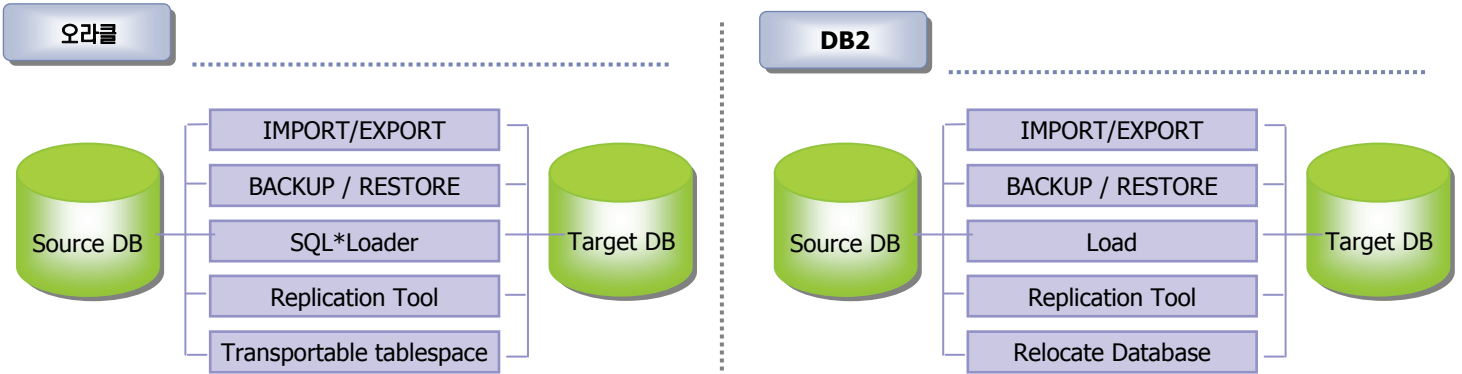
데이터 이동 방법 비교

EXPORT/IMPORT/LOAD 유틸리티 비교

DB2 EXPORT / IMPORT /LOAD 소개



Point 데이터베이스내에 존재하는 데이터를 다른 플랫폼 또는 다른 데이터베이스로 이동하는 방법에는 Import, Export 및 Load 유틸리티, 백업 및 복구, 복제등이 있습니다. 이 곳에서는 오라클과 DB2에서 제공하는 데이터 이동 방법에 대해 알아봅니다.



IMPORT/EXPORT 툴을 이용한 데이터 이동 :

- DATA PUMP IMPORT/EXPORT 유틸리티를 이용하여 전용 포맷인 DUMP 파일 형태로 데이터를 이동합니다.
- 이기종간의 데이터베이스에 대한 데이터

RMAN (Recovery Manager)을 이용한 데이터 이동:

- 백업 및 복구를 이용하여 데이터를 이동합니다.

SQL*Loader 유틸리티를 이용한 데이터 이동 :

- 외부 파일을 오라클 데이터베이스 테이블로 데이터를 적재합니다.

복제 솔루션을 이용한 데이터 이동 :

- 복제 솔루션을 이용하여 데이터베이스를 복제합니다.

transportable tablespace를 이용한 데이터 이동:

- 오라클 10g 부터 제공하는 기능으로 OS Byte Order 가 동일 할 경우 서로 다른 플랫폼일지라도 테이블 스페이스 이동이 가능합니다.

IMPORT/EXPORT 툴을 이용한 데이터 이동 :

- IMPORT/EXPORT 유틸리티를 이용해 특정 테이블에 대한 데이터를 이동합니다.
- 표준 데이터 형식인 ASC,DEL 형식 또는 DB2 전용 형식인 IXF 를 지원합니다. .

BACKUP / RESTORE 를 이용한 데이터 이동:

- 백업 및 복구를 이용하여 데이터를 이동합니다.

LOAD 유틸리티를 이용한 데이터 이동 :

- LOAD 유틸리티를 이용하여 특정 테이블, 특정 스키마에 대한 데이터를 적재할 수 있습니다.
- HPU (High Performance Unload) 제품을 이용하여 데이터베이스의 데이터를 고속으로 빠르게 적재합니다.

복제 솔루션을 이용한 데이터 이동 :

- 복제 솔루션을 이용하여 데이터베이스를 복제합니다.


reloadedb 를 이용한 데이터베이스 이동 :

- 데이터베이스를 같은 시스템 내 또는 다른 시스템에 재배치하여 이동시킬 수 있습니다.




오라클의 IMPORT와 EXPORT 유틸리티와 DB2의 IMPORT/EXPORT는 기능상 유사합니다. 그러나 오라클의 IMPORT/EXPORT 는 오라클 전용 포맷인 DUMP 파일 형태로 데이터를 이동하는 것으로 이기종 데이터베이스간에 데이터를 이동시킬 수 없습니다. DB2는 전용 포맷인 IXF 파일형태뿐 아니라 ASCII 파일 형태로 데이터를 이동할 수 있으므로 이기종 데이터베이스간에 데이터 이동이 가능합니다.

02 DB2 데이터 이동 방법

Point  DB2는 다양한 데이터 이동 방법을 가지고 있습니다. 이 곳에서는 데이터 이동과 관련된 유틸리티를 설명합니다. 이 표를 참고하여 요구사항에 맞는 유틸리티를 선택합니다.

	Load	db2move
목적	대용량 데이터를 새 테이블 또는 기존 테이블로 효율적으로 이동할 수 있습니다.	<ul style="list-style-type: none"> • IMPORT/EXPORT 옵션과 함께 사용할 경우 DB2 데이터베이스간에 대량의 테이블을 편리하게 이동할 수 있습니다. • COPY 옵션과 함께 사용할 경우 소스 데이터베이스로에서 타겟 데이터베이스로 스키마 템플릿을 복사할 수 있습니다.
Cross platform 호환여부	가능	가능
Best Practice	성능이 가장 최우선될 경우	cross platform 에 대한 백업 및 복구를 지원하지 않을 경우 IMPORT, EXPORT 옵션을 이용하여 데이터베이스를 복제할 수 있습니다.
	IMPORT	EXPORT
목적	외부 파일로부터 데이터를 읽어 테이블, 뷰, nickname에 삽입하고자 할 경우 사용됩니다.	<ul style="list-style-type: none"> • 데이터베이스로부터 데이터를 다양한 형태의 외부 파일로 추출해 내고자 할 경우 사용됩니다.
Cross platform 호환여부	가능	가능
Best Practice	<p>IMPORT 는 다음과 같은 경우일 경우 LOAD 유틸리티를 대신 할 수 있는 툴입니다.</p> <ul style="list-style-type: none"> • 타겟 테이블이 뷰일 경우 • 타겟 테이블에 제약조건이 존재하여 Load시 "Set Integrity Pending" 상태를 원하지 않을 경우 • 타겟 테이블에 트리거가 존재하며 트리거가 수행되기를 원할 경우 	<p>외부 파일로 데이터를 저장하여 향후 다른 테이블에 데이터를 이동하고자 할 경우 사용됩니다.</p> <p>High Performance Unload (HPU) 는 고성능 유틸리티이나 별도로 구매해야 합니다.</p> <p>EXPORT 유틸리티는 XML 칼럼을 지원합니다.</p>
	ADMIN_COPY_SCHEMA 프로시저	REDIRECT, GENERATE SCRIPT 옵션과 함께 사용된 RESTORE
목적	단일 스키마내의 모든 오브젝트의 복사본을 생성하여 새로운 스키마에 속하는 오브젝트로 재생성할 수 있습니다. copy 옵션을 이용하여 데이터와 함께 복사할 것인지 아닐지를 지정할 수 있습니다.	<ul style="list-style-type: none"> • 기존 백업 이미지로부터 스크립트를 이용하여 한 시스템에서 다른 시스템으로 전체 데이터베이스를 복사할 경우 사용됩니다.
Cross platform 호환여부	가능	제한적
Best Practice	스키마 템플릿을 생성하여 기존 소스 스키마에 영향을 주지 않은 채 새로 인덱스를 만들어 테스트를 수행할 수 있습니다.	백업 이미지가 존재할 경우 가장 적합합니다.
	db2reloadedb 명령어	Split Mirror
목적	데이터베이스 이름의 재정의, 데이터베이스 위치 변경, 데이터베이스의 일부분을 같은 시스템 또는 다른 시스템에 위치시키고자 할 경우 사용됩니다.	복제본, Standby 또는 백업 데이터베이스를 생성하는데 사용됩니다.
Cross platform 호환여부	가능하지 않음	가능하지 않음
Best Practice	<ul style="list-style-type: none"> • 백업 또는 복구작업이 오래 걸릴 경우 이용될 수 있습니다. • 백업 및 복구의 대안책으로 데이터베이스의 복사본을 생성하거나 이동할 수 있습니다. • 테스트 환경을 구성할 경우 빨리 데이터베이스의 복제본을 생성할 수 있습니다. • 테이블 스페이스 컨테이너를 새로운 스토리지 디바이스 그룹으로 이동시킬 수 있습니다. 	<ul style="list-style-type: none"> • down time을 최소화하기 위해 primary 서버가 down 되었을 경우 standby 시스템을 생성합니다. • 테스트 환경을 구성할 경우 데이터베이스의 복제본을 가장 빠른 방법으로 생성할 수 있습니다.

Point  다음은 오라클과 DB2에서 데이터 이동에 가장 많이 사용되는 Export 와 Import 유틸리티를 비교합니다.

EXPORT / IMPORT 비교

	오라클	DB2
설명	<ul style="list-style-type: none"> • 논리적인 백업에 사용되는 유틸리티로, 바이너리 파일 형태로 기록합니다. • 데이터베이스가 가동중인 상태에서 실행합니다. • 전체 데이터베이스, 특정 사용자(특정 스키마), 특정 테이블, 데이터베이스 등을 익스포트 시킬 수 있습니다. • 오라클 전용 데이터 형식인 덤프 파일을 이용하여 데이터를 이동합니다. • Conventional Path 로더 및 Direct Path 로더 중 선택할 수 있습니다. 	<ul style="list-style-type: none"> • SQL 문 또는 Xquery를 사용하여 데이터를 추출/삽입해 내는 유틸리티입니다. • 데이터베이스가 가동중인 상태에서 실행합니다. • 특정 테이블 데이터를 이동합니다. • 표준 데이터 형식인 ASC,DEL 형식 또는 DB2 전용 형식인 IXF 를 지원합니다. • 오라클처럼 전체 데이터베이스, 특정 스키마등을 EXPORT/IMPORT 하려면 EXPORT/IMPORT 와 함께 db2move 라는 유틸리티를 사용합니다. • 오라클의 Conventional Path 로더와 유사합니다.
구문	<ul style="list-style-type: none"> • IMP <USERID>/<PASSWD> FILE=<입력 파일명> • EXP <USERID>/<PASSWD> FILE=<입력 파일명> 	<ul style="list-style-type: none"> • IMPORT FROM <입력 파일명> OF DEL MESSAGES <메시지 파일 이름> INSERT INTO <목표 테이블 명> • EXPORT TO <출력 파일명> OF <DEL WSF IXF> SELECT 문

예문

오라클

EXPORT 구문 :

```
$ exp hr/hr tables=employee query=""where salary <=100" file=emp.dmp
```

IMPORT 구문 :

```
imp hr/hr file=ADMINISTRATOR.sql fromuser=eworld touser=eworld
```



DB2

EXPORT 구문 :

```
db2=> export emp.ixf of IXF MESSAGES amj.msgs
SELECT * FROM scott.employee where salary <= 100
```

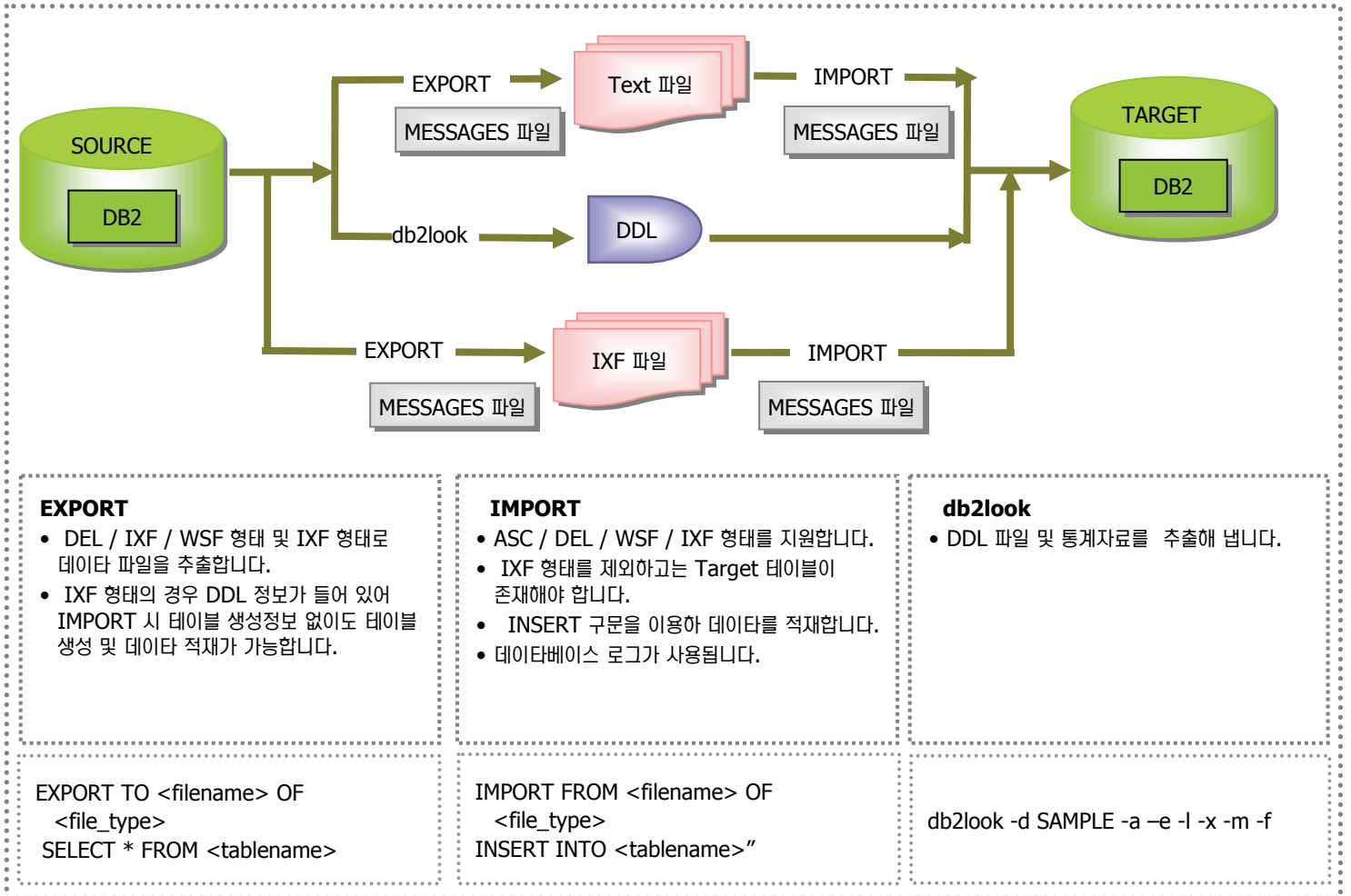
IMPORT 구문:

```
db2=> import from emp.ixf MESSAGES amj.msgs
INSERT INTO amj.employee
```



오라클의 IMPORT와 EXPORT 유틸리티와 DB2의 IMPORT/EXPORT는 기능상 유사합니다. 그러나 오라클의 IMPORT/EXPORT 는 오라클 전용 포맷인 DUMP 파일 형태로 데이터를 이동하는 것으로 이기종 데이터베이스간에 데이터를 이동시킬 수 없습니다. DB2는 전용 포맷인 IXF 파일형태뿐 아니라 ASCII 파일 형태로 데이터를 이동할 수 있으므로 이기종 데이터베이스간에 데이터 이동이 가능합니다. 다만 오라클의 IMPORT 와 EXPORT 는 전체 데이터베이스 및 사용자 별, 테이블 별로 데이터를 추출하여 적재할 수 있으나 DB2의 IMPORT/EXPORT는 하나의 테이블만 선택할 수 있습니다. 만약 오라클의 IMPORT/EXPORT 처럼 전체 데이터베이스 및 스키마별, 테이블 별로 하고자 할 경우 db2move 라는 유틸리티를 사용하면 오라클의 IMPORT/EXPORT 와 동일하게 수행할 수 있습니다.

Point DB2의 EXPORT 와 IMPORT 는 데이터를 이기종간에 이동하기 위한 가장 간편한 방법입니다. 다음은 DB2 Export 와 Import 에 대해 알아보도록 합니다.



* 부연설명

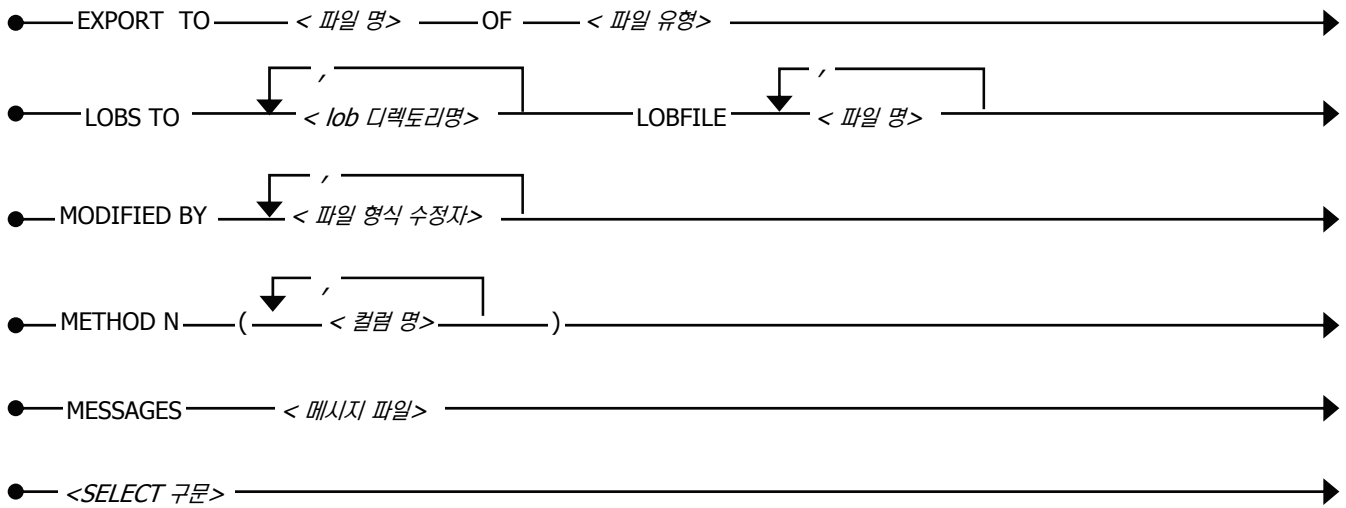
DB2에서 테이블의 데이터를 파일로 저장하거나 파일의 데이터를 테이블에 적재하는 유틸리티에서 사용되는 파일의 유형에는 ASC, DEL, WSF, IXF 파일등이 있습니다. IXF 는 DB2 전용 파일 포맷으로 데이터와 컬럼에 대한 제어 정보를 가지고 있어 적재하고자 하는 테이블이 존재하지 않을 경우에도 IXF 파일을 읽어 테이블을 생성하여 데이터를 적재할 수 있습니다.

파일 유형 :

유형	설명
ASC	Non-Delimited ASCII 파일로, IMPORT / LOAD 시 사용됩니다.
DEL	Delimited ASCII 파일로, EXPORT/IMPORT/LOAD 시 사용됩니다.
WSF	Work Sheet Format 으로 Lotus 1-2-3, Symphony 등에 사용됩니다
IXF	Integrated eXchange Format 으로 EXPORT/IMPORT/LOAD 시에 사용되며 데이터를 삽입할 테이블이 존재하지 않아도 IXF 파일에 저장된 테이블 구조를 읽어 테이블을 생성한 후 데이터를 적재합니다.
CURSOR	SELECT 문의 결과 집합을 저장한 구조체로 LOAD 에 사용됩니다.

Point  다음은 DB2 EXPORT 유틸리티 명령문입니다.


EXPORT 구문



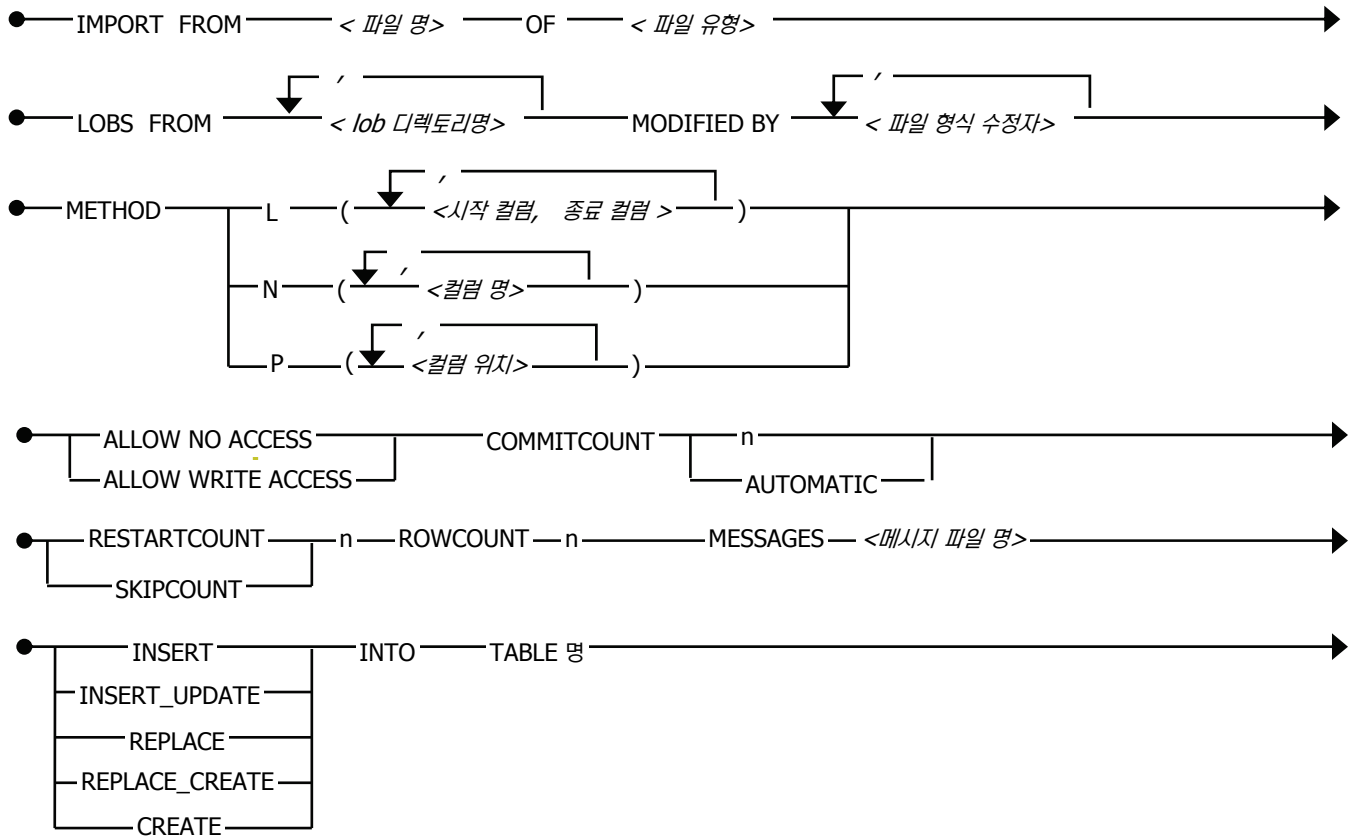
참고 : 이외에도 다양한 옵션이 제공되니 도움말을 참고하시기 바랍니다.

옵션 설명 :

모드	설명
TO < 파일명 >	출력 데이터 파일명을 지정합니다.
OF < 파일 유형 >	DEL, WSF, IXF 중의 한 가지를 지정하여 파일 유형을 지정합니다. DB2간 데이터 이동일 경우 IXF 파일 유형을 사용하면 편리하며 이기종 데이터베이스간의 데이터 이동일 경우 DEL 파일을 사용하면 편리합니다.
LOBS TO < 디렉토리명 >	LOB 컬럼이 저장될 디렉토리명을 지정합니다.
LOBFILE <파일명 >	LOB 컬럼이 저장될 파일명을 지정합니다. 3자리 숫자의 확장자가 자동으로 생성됩니다.
MODIFIED BY < 파일 형식 수정자 >	다양한 옵션으로 데이터 파일의 형식을 제어합니다.
METHOD N	컬럼의 이름으로 데이터를 지정합니다.
MESSAGES	작업 결과를 기록하는 메시지 파일명을 지정합니다.
<SELECT 문>	데이터를 추출할 SELECT 문을 지정합니다.

Point  다음은 DB2 IMPORT 유틸리티 명령문입니다.

IMPORT 구문

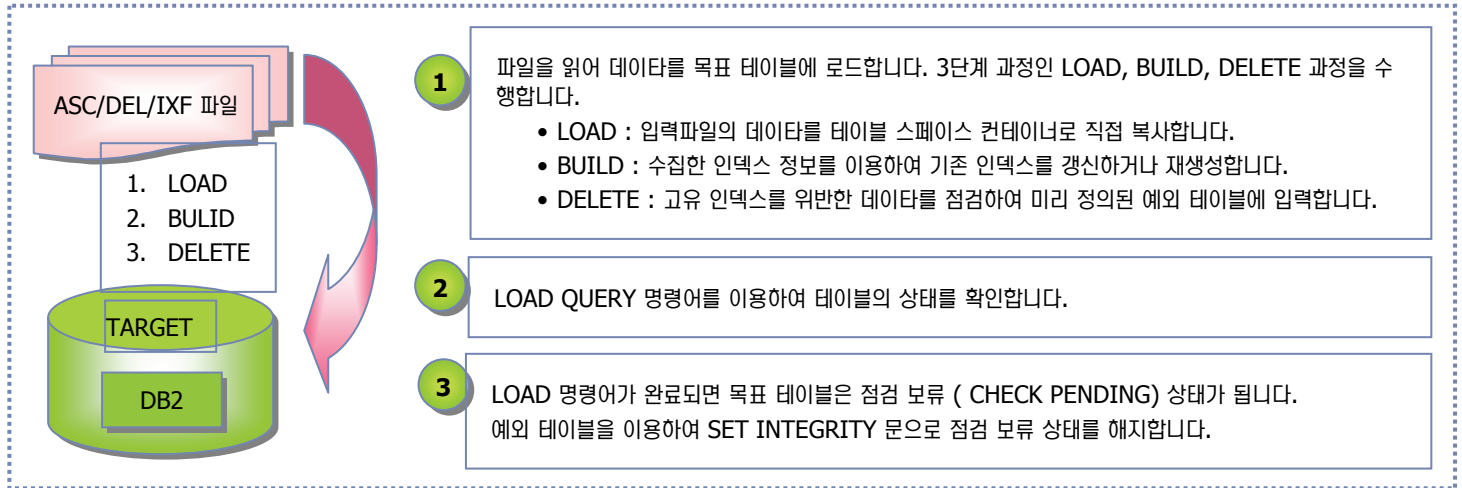


참고 : 이외에도 다양한 옵션이 제공되니 도움말을 참고하시기 바랍니다.

옵션 설명

모드	설명
FROM <입력 파일명>	입력 데이터 파일명을 지정합니다.
OF < 파일 유형 >	ASC, DEL, WSF, IXF 중 한 가지를 지정합니다.
METHOD L/N/P	필드의 시작과 종료 위치의 바이트 옵션, 컬럼 이름, 필드 번호로 컬럼별 데이터를 지정합니다.
COMMITCOUNT	n건을 처리할때마다 COMMIT 을 실행합니다.
RESTARTCOUNT n	n+1 번째 데이터부터 처리합니다.
ROWCOUNT n	n건의 데이터만 처리합니다.
MESSAGES	작업 결과를 기록하는 메시지 파일명을 지정합니다.
INSERT	기존의 목표 테이블에 데이터를 추가합니다.
INSERT_UPDATE	기존의 목표 테이블에 데이터를 추가 또는 갱신합니다.
REPLACE	기존의 목표 테이블의 데이터를 로그를 발생시키지 않고 TRUNCATE 한 후 INSERT 문으로 데이터를 추가합니다.
CREATE	목표 테이블을 생성하고 데이터를 입력합니다.
INTO <테이블명>	목표 테이블 이름을 지정합니다.

Point DB2의 LOAD 는 IMPORT 와 유사한 틀로 파일로부터 대량의 데이터를 기존의 테이블에 고속으로 저장하는 유틸리티입니다. LOAD 명령어 와 SET INTEGRITY 명령어가 사용됩니다.



LOAD 특징

- 많은 양의 데이터를 빨리 효과적으로 적재할 수 있습니다.
 - format 된 페이지를 직접 데이터베이스내에 기록합니다.
 - 트리거를 일으키지 않습니다.
 - referential checking 을 수행하지 않습니다.
- 다음과 같이 4가지 단계를 이용하여 LOAD 유틸리티를 수행합니다.
 - Load, Build, Delete 및 Index Copy
- Online load 가 가능합니다.
 - 기존 테이블 데이터를 LOAD 시에도 읽을 수 있습니다.
- 통계자료를 생성할 수 있습니다.
- exception 데이터를 별도의 테이블로 복사할 수 있습니다 .
- INPUT 값으로 PIPE 또는 CURSOR 를 이용할 수 있습니다.
- LOB 데이터 및 UDT를 다룰 수 있습니다.
- ASC,DEL, IXF 파일 형태를 모두 지원합니다.

1

```
LOAD FROM <filename> OF <filetype>
INSERT INTO <table-name>
```

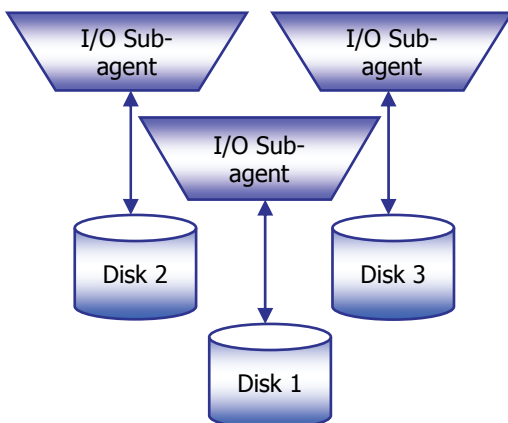
2

```
LOAD QUERY TABLE <table-name>
```

3


```
SET INTEGRITY FOR <테이블 명> IMMEDIATE
CHECKED FOR EXCEPTION IN <목표 테이블 명>
USE <예외 테이블 명>
```

LOAD Self Tuning 특징



- DB2 Load 유틸리티는 자동으로 다음과 같은 속성을 기반으로 LOAD 작업을 튜닝합니다.

- I/O sub-agents
 - 테이블 공간의 컨테이너, CPU 등에 기반하여 선택합니다.
- Index 유지보수 모드
 - Incremental versus Rebuild
 - "INDEXING MODE AUTOSELECT" 를 사용합니다.
- 메모리 사용량
 - UTIL_HEAP_SZ의 빈공간을 사용합니다.
- 서버 리소스 사용율
 - UTIL_IMPACT_PRIORITY를 사용합니다.

Point  LOAD 유틸리티의 첫번째 과정으로 입력 파일의 데이터를 테이블 스페이스 컨테이너로 직접 복사합니다. 컨테이너에 추가된 데이터는 데이터베이스 로그 파일에 기록되지 않으며 데이터의 추가로 인한 INSERT 트리거도 실행되지 않습니다.

LOAD 단계



```
LOAD FROM kes.del OF DEL SAVECOUNT 1000
MODIFIED BY DUMPFIL /work/load.dmp
MESSAGES load.msges
INSERT INTO amj.emp
FOR EXCEPTION amj.empexp
```

1 LOAD 단계

1. 파일의 데이터를 amj.emp 테이블이 위치한 테이블 스페이스의 컨테이너로 직접 복사합니다.
2. SAVECOUNT 옵션을 이용하여 내부적으로 복사 완료 중간 시점을 지정합니다. 이 곳에서는 지정된 1000 행마다 consistency 포인트를 지정합니다. 만약 LOAD가 이 단계에서 실패할 경우 이 시점 이후부터 복사를 계속할 수 있습니다.
3. 데이터 유형과 호환되지 않은 행 및 NULL 값이 허용되지 않는 컬럼에 NULL 값이 포함된 행 등 유효하지 않은 데이터를 dump 파일로 지정된 파일인 load.dmp 파일에 기록합니다.
4. 목표 테이블에 존재하는 모든 인덱스에 대한 인덱스 키값과 대응되는 RID 정보를 미리 수집합니다.

입력 모드 :

모드	설명
INSERT	목표 테이블에 입력 파일의 데이터를 추가합니다.
REPLACE	목표 테이블의 기존 데이터를 데이터베이스 로그 파일에 기록하지 않고 Truncate 한 후에 입력 파일의 데이터로 대체합니다.

2 BUILD 단계

1. LOAD 단계에서 수집된 인덱스 정보를 이용하여 기존의 인덱스를 갱신하거나 재생성합니다.

인덱스 반영 모드 :


모드	설명
인덱스 갱신	목표 테이블에 존재하는 모든 인덱스에 새로 추가된 행에 대한 인덱스 데이터를 추가합니다.
인덱스 재생성	<p>목표 테이블에 존재하는 모든 인덱스를 제거하고 다시 생성합니다.</p> <ul style="list-style-type: none"> • AUTO : DB2가 자동으로 UPDATE 와 REBUILD 모드 중에서 선택합니다. • UPDATE : 목표 테이블에 존재하는 모든 인덱스에 추가된 데이터의 인덱스 정보를 추가합니다. • REBUILD : 목표 테이블에 존재하는 모든 인덱스를 재생성합니다. • DEFERRED : 목표 테이블에 존재하는 모든 인덱스의 재생성 작업을 실행하지 않고 보류합니다.

3 DELETE 단계

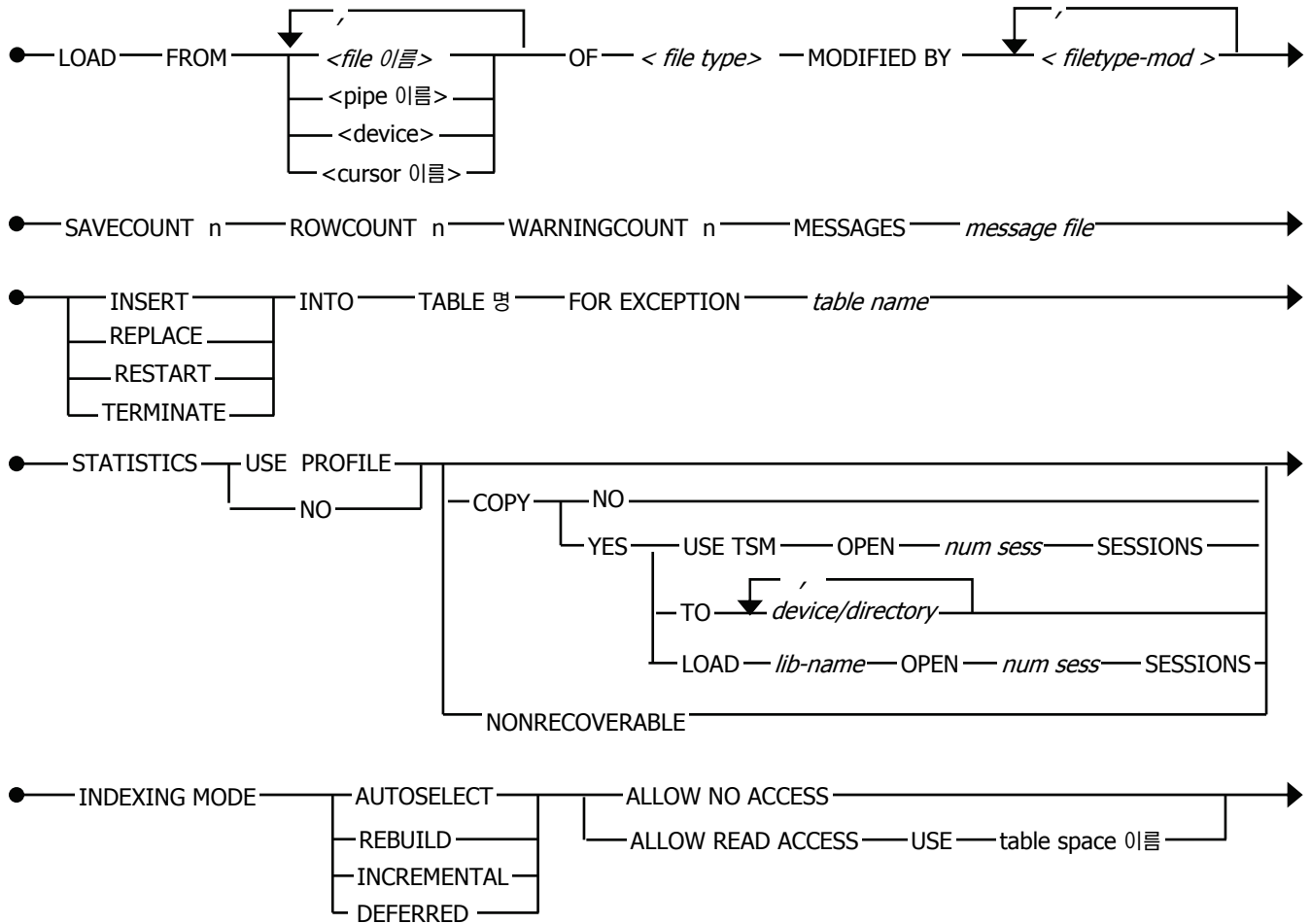
1. LOAD 단계에서 추가된 데이터 중에서 고유 인덱스의 규정을 위반한 행들을 점검하여 예외 테이블인 amj.empexp 에 삽입합니다.



LOAD 유틸리티가 실패하여 TERMINATE 명령어를 이용하여 취소할 경우 INSERT 와 REPLACE 모드에 따라 달라집니다. INSERT 모드 일 경우에는 LOAD 명령어를 실행하기 이전의 상태로 복구되고 REPLACE 모드를 사용할 경우에는 테이블의 데이터가 모두 삭제됩니다.

Point  다음은 DB2 LOAD 유틸리티 명령문입니다.


LOAD 구문



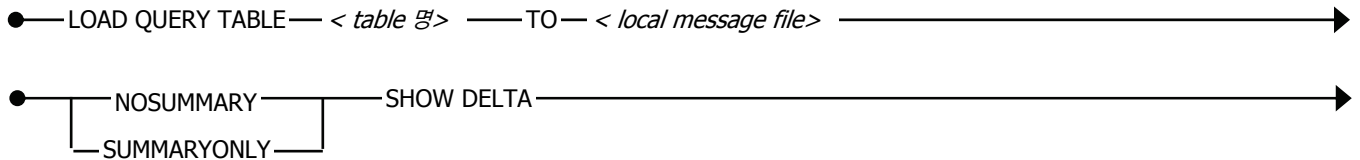
참고 : 이외에도 다양한 옵션이 제공되고 있으니 도움말을 참고하시기 바랍니다.

옵션 설명

모드	설명
FROM <입력파일 명>	입력 데이터 파일명을 지정합니다.
OF <파일 유형>	ASC,DEL,IXF 중의 한가지를 지정합니다.
SAVECOUNT n	n건을 복사할 때마다 복사 완료 정보를 보관합니다.
ROWCOUNT n	n건의 데이터만 처리합니다.
MESSAGES <파일명>	작업 결과를 기록하는 메시지 파일명을 지정합니다.
INSERT / REPLACE	목표 테이블에 데이터를 추가 또는 교체 합니다.
RESTART/TERMINATE	중단된 LOAD 작업을 다시 시작하거나 종료합니다.
INTO <테이블명>	목표 테이블명을 지정합니다.
FOR EXCEPTION <테이블명>	예외 테이블명을 지정합니다.
COPY <모드>	백업 실행 여부를 지정합니다.
INDEXING MODE <모드>	인덱스 재생성 모드를 지정합니다.
ALLOW READ ACCESS	Load 명령어가 실행되기 이전의 데이터에 대한 읽기 액세스를 허용합니다.

Point  로드 유틸리티가 로드 조작 중에 데이터베이스 일관성을 위해 변경하는 테이블의 상태값을 확인하는 명령어로 LOAD 명령어의 현재 실행 단계 및 LOAD 단계에서 복사 완료된 데이터의 건수와 재생성 또는 갱신이 완료된 인덱스의 개수도 확인할 수 있습니다.

LOAD QUERY 구문



참고 : 이외에도 다양한 옵션이 제공되고 있으니 도움말을 참고하시기 바랍니다.

옵션 설명 :

모드	설명
<테이블명>	LOAD 명령어가 실행중인 목표 테이블명을 지정합니다.
TO <메시지 파일명>	LOAD 명령어가 현재까지 실행한 상황을 저장할 메시지 파일명을 지정합니다.
NOSUMMARY	LOAD 명령어가 처리한 데이터 건수에 대한 요약 정보 (읽은 행 수, 건너뛴 행 수, 입력된 행 수, 거부된 행 수, 삭제된 행 수, 커밋된 행 수, 경고 개수)를 제공하지 않습니다.
SUMMARYONLY	LOAD 명령어가 처리한 데이터 건수에 대한 요약 정보만 제공합니다.
SHOW DELTA	최근의 LOAD QUERY 명령어 실행 이후에 변경된 정보만 제공합니다.

예문

load query 예 :

```
/home/inst95> db2 load query table amj.EMP summaryonly
```

```

Number of rows read           = 1100
Number of rows skipped        = 0
Number of rows loaded         = 1100
Number of rows rejected       = 0
Number of rows deleted        = 0
Number of rows committed     = 1100
Number of warnings            = 0


```

```

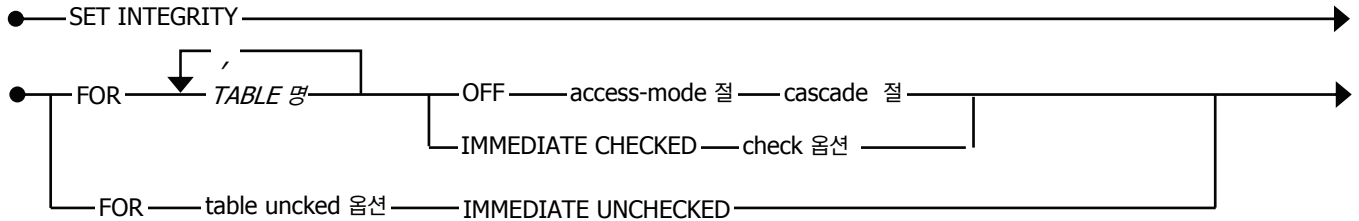
Tablestate :
Load in Progres

```

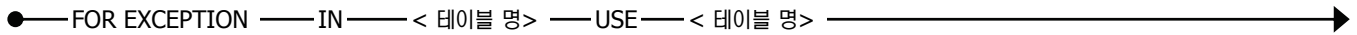
11 SET INTEGRITY 명령어

Point  LOAD 유틸리티는 테이블의 외부키와 점검 제한 조건을 점검하지 않으므로 LOAD 명령어가 완료된 후에 목표 테이블은 '점검 보류 상태 (CHECK PENDING)' 상태가 됩니다. SET INTEGRITY 을 수행하여 점검 보류 상태를 해지합니다.

SET INTEGRITY 구분



CHECK 옵션 :



참고 : 이외에도 다양한 옵션이 제공되고 있으니 도움말을 참고하시기 바랍니다.

옵션 설명 :

모드	설명
FOR <테이블 명>	목표 테이블 명을 지정합니다.
IMMEDIATE CHECKED	즉시 점검하도록 요청합니다.
FOR EXCEPTION	예외 테이블이 사용되는 것을 알려줍니다.
FOR <테이블 명>	예외 행이 검출된 테이블 명입니다.
USE <테이블 명>	예외 테이블명입니다.

예문

set integrity 예 :

```
/home/inst95> db2 ? SQL0668N
```

```
SQL0668N 이유 코드 "<reason-code> 로 인해 테이블 "<table-name>" 에서 조작이 허용되지 않습니다.
```

설명 :

테이블 "<table-name>" 에 대한 액세스가 제한됩니다. 원인은 "<reason-code>"에 기초합니다.

1. 테이블이 점검 보류 상태입니다. 테이블의 무결성이 적용되지 않았으므로 테이블 내용이 유효하지 않을 수 있습니다. 종속 테이블이 점검 보류 상태이면 점검 보류 상태가 아닌 상위 테이블이나 기본 테이블에서 조작을 수행할 경우에도 이 오류를 수신할 수 있습니다.

```
/home/inst95> db2 "SELECT const_checked FROM syscat.tables WHERE tabschema='AMJ' AND tablename='EMP'"
```

```
CONST_CHECKED
```

```
NNYYYYYYYYYYYYYYYYYYYYYYYYYYYY
```


1 레코드가 선택됨

```
/home/inst95> db2 "set integrity for emp immediate checked for exception in emp use empexp"
```

```
/home/inst95> db2 "select id,substr(name,1,6) name, mydept, sex from empexp"
```

→ 예외 테이블에 외부키와 점검 제한 조건을 위반한 행이 입력되었는지 확인합니다.

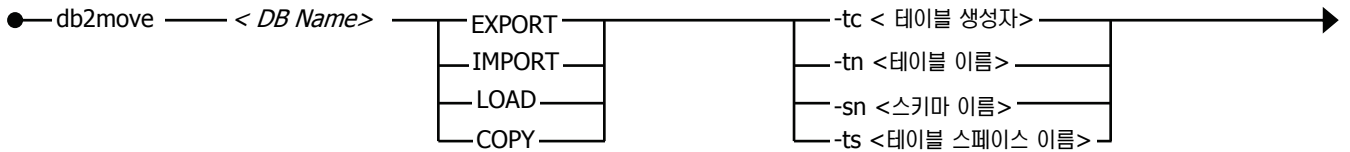
12 IMPORT와 LOAD 의 차이점

Point  다음은 IMPORT 와 LOAD 의 차이점입니다. IMPORT 와 LOAD 는 모두 외부 데이터 파일을 읽어 데이터를 데이터베이스로 적재할 수 있는 유틸리티입니다.

Import	Load
많은 양의 데이터를 적재할 때 LOAD 보다 느립니다.	많은 양의 데이터를 적재할 때 IMPORT 보다 빠릅니다.
제한된 intra-partition 병렬성을 이용합니다.	항상 intra-partition 병렬성을 이용합니다.
계층적 데이터를 지원합니다.	계층적 데이터를 지원하지 않습니다.
IXF 형태일 경우 테이블, 계층, 인덱스가 자동 생성됩니다.	테이블 및 인덱스는 형태에 상관없이 존재해야 합니다.
Importing 을 Summary Table로 지원하지 않습니다.	Support for loading into summary tables
WSF 형식이 지원됩니다.	WSF 형식이 지원되지 않습니다.
BINARYNUMERICS/PACKEDDECIMAL/ZONEDDECIMAL 이 지원되지 않습니다.	BINARYNUMERICS / PACKEDDECIMAL / ZONEDDECIMAL 지원됩니다.
모든 행이 로그됩니다.	최소한의 로깅이 수행됩니다.
트리거가 지원됩니다.	트리거가 지원되지 않습니다.
"GENERATED ALWAYS"로 선언된 컬럼을 덮어 쓸 수 없습니다.	"GENERATED ALWAYS"로 선언된 컬럼을 덮어 쓸 수 있습니다.
Import 시에 모든 제약조건이 검증됩니다.	Load 연산동안에는 Uniqueness만이 검증되며 모든 다른 제약조건들은 "SET INTEGRITY"를 통해 체크됩니다.
import시에 Key 값이 Index 테이블에 한번에 하나씩 삽입됩니다.	데이터가 모두 적재된 후에 Key 값이 정렬되어 인덱스가 생성됩니다.
통계자료의 변경을 요구할 경우 RUNSTATS 유틸리티가 Import 후에 수행되어야 합니다.	테이블내의 모든 데이터가 교체되면 LOAD 연산동안 통계자료가 수집됩니다.
DB2 Connect 를 이용하여 host (mainframe) 데이터베이스로 import 가 가능합니다.	host (mainframe) 데이터베이스로 Load할 수 없습니다.
테이블, 뷰, 닥네임 등으로 Import 가 가능합니다.	테이블만 Load 작업이 가능합니다.

Point db2move 유틸리티를 이용하여 테이블 별, 테이블 스페이스 별, 스키마 별 또는 전체 데이터베이스에 대한 IMPORT, EXPORT, LOAD 작업을 수행할 수 있습니다. 오라클의 IMPORT/EXPORT 툴에 해당한다고 보시면 됩니다.

db2move 구문



참고 : 이외에도 다양한 옵션이 제공되오니 도움말을 참고하시기 바랍니다.

옵션 설명

옵션	파라미터	설명
-tc	테이블 생성자	디폴트는 모든 생성자입니다. 생성한 사용자에 의해 생성된 테이블을 EXPORT 합니다. EXPORT에 대해서만 적용되는 옵션입니다.
-tn	테이블 이름	디폴트는 모든 사용자 테이블입니다. 지정한 테이블을 EXPORT 합니다. EXPORT에 대해서만 적용되는 옵션입니다.
-sn	스키마 이름	디폴트는 모든 사용자 테이블입니다. 지정한 스키마를 EXPORT 합니다.
-ts	테이블스페이스 이름	디폴트는 모든 테이블 스페이스입니다. 지정한 테이블 스페이스에 속하는 테이블들이 EXPORT 합니다.

필요 파일 및 자동 생성되는 파일

	설명
EXPORT 시	input파일은 필요치 않으며 db2move.lst 파일, tabnnn.ixf, tabnnn.msg, tabnnnc.yyy 파일이 자동으로 생성됩니다.
IMPORT 시	input 파일로 db2move.lst 파일, tabnnn.ixf, tabnnn.msg, tabnnnc.yyy 파일이 필요하며 IMPORT.out 파일과 tabnnn.msg 파일이 자동 생성됩니다.

예문

db2move 를 이용한 EXPORT 예 :

```
/home/inst95> db2move sample export
```

```
/home/inst95>db2move sample export -tc us*rid2 -tn tname1, *tname2
```

→ us로 시작하여 rid2 로 끝나는 모든 사용자가 생성한 테이블과 테이블 이름이 tname1 과 tb로 시작하여 name2로 끝나는 테이블들을 EXPORT 합니다.

db2move 를 이용한 IMPORT 예 :

```
/home/inst95>db2move sample import -io replace -u inst95 -p ****
```

→ SAMPLE 데이터베이스내의 모든 테이블을 replace 모드로 IMPORT 합니다.

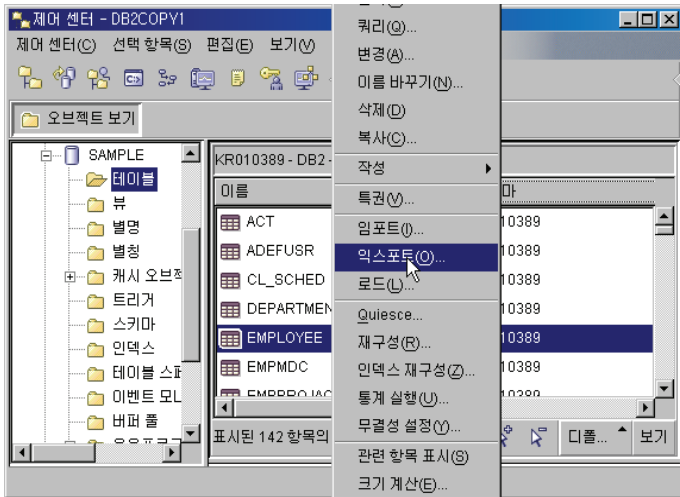
```
home/inst95>db2move DBSRC COPY -sn AMJ -co TARGET_DB dbtgt USER inst95 using ****
```

→ Source 데이터베이스의 AMJ 스키마를 Target 데이터베이스로 복사합니다.

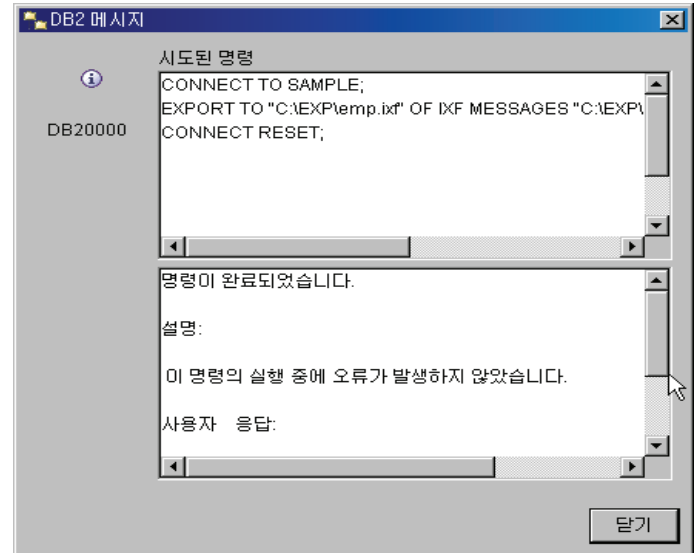
14 제어센타를 이용한 EXPORT

Point 제어센타를 이용하여 DB2 데이터베이스에 있는 테이블을 EXPORT 합니다.

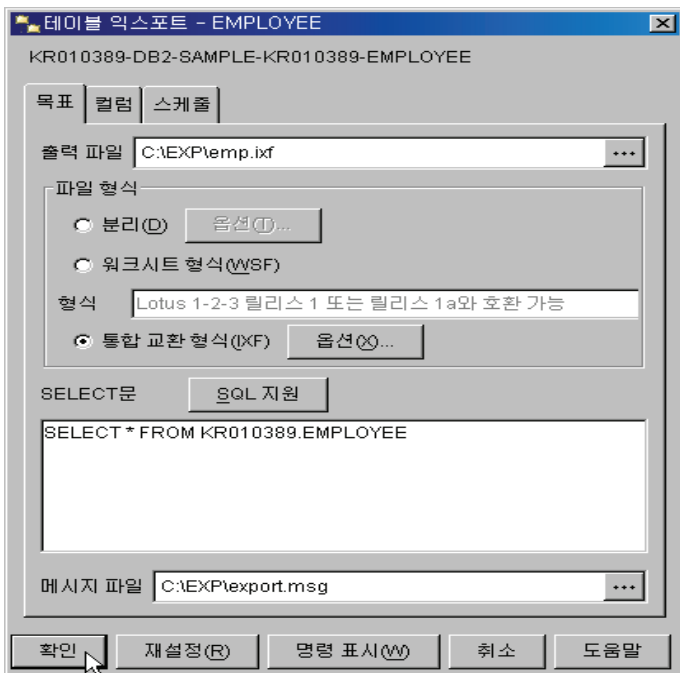
1. 제어센타에서 EXPORT 할 테이블인 EMPLOYEE을 선택하여 오른쪽 마우스를 선택한 후 익스포트를 클릭합니다.



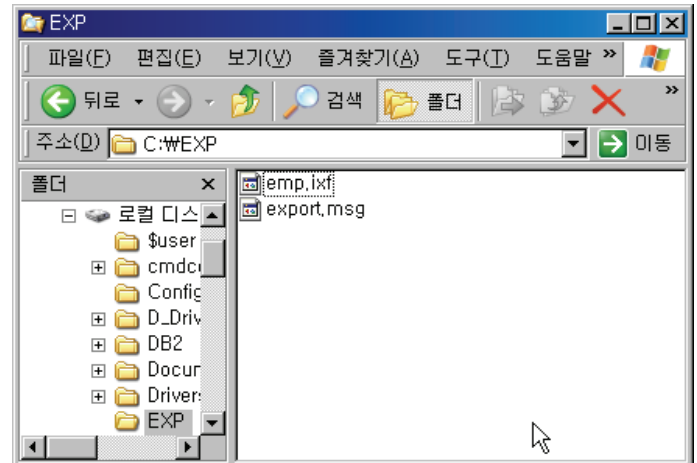
3. EXPORT 가 제대로 수행이 되었는지 확인합니다.




2. 다음과 같이 목표 탭이 나타나면 데이터를 export 할 파일을 지정하고 파일 형식을 선택합니다. 파일 형식은 ASCII, WSF, IXF 중 선택 가능합니다. DB2에서는 IXF 형태를 많이 사용하므로 IXF를 선택합니다. 그리고 메시지 파일에 경고 및 여러 발생시 메시지를 기록할 파일을 선택하고 확인 버튼을 클릭합니다.



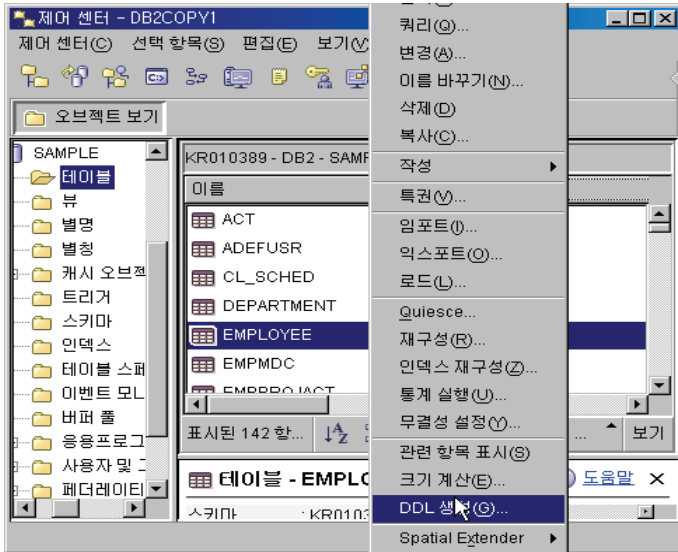
4. 탐색기에서 Export 한 파일과 메시지 파일이 생성되었는지 확인합니다.



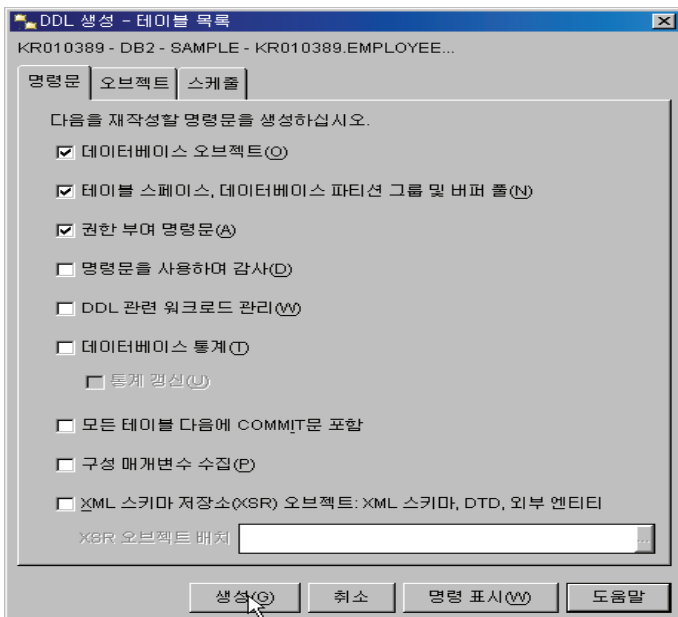
15 제어센터를 이용한 IMPORT

Point  제어센터를 이용하여 DB2 데이터베이스에 있는 테이블로 IMPORT 합니다.

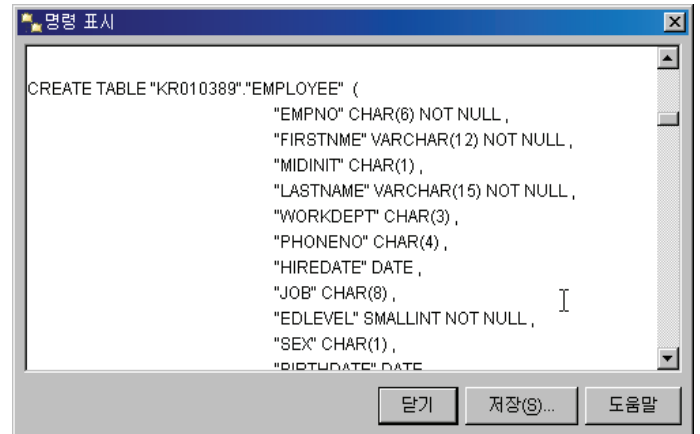
1. AMJ.EMPLOYEE 라는 테이블에 데이터를 IMPORT 하도록 합니다. 이때 AMJ.EMPLOYEE 테이블은 존재하지 않으므로 기존 KR010389.EMPLOYEE 테이블과 동일한 구조로 AMJ.EMPLOYEE 테이블을 생성하도록 합니다. 다음과 같이 EMPLOYEE 를 선택한 후 오른쪽 버튼을 클릭하여 DDL 생성을 선택합니다.



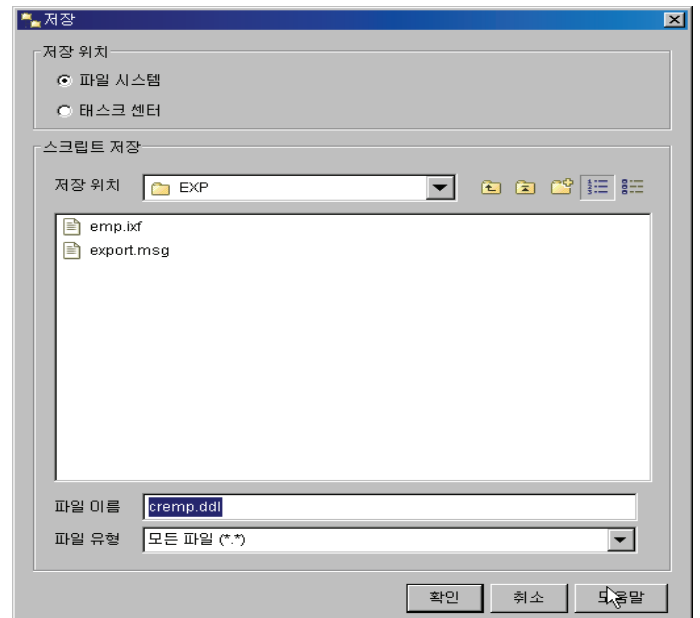
2. 명령문 탭에서 오브젝트 및 스페이스, 권한 부여에 대한 속성도 DDL 문에 적용할 경우 선택하며 오브젝트 탭으로 이동하여 생성할 테이블이 제대로 선택되었는지 확인한 후 생성 버튼을 클릭합니다.




3. 다음과 같이 명령 표시창이 나타나면 테이블 생성 명령문이 제대로 나타나는지 확인한 후 저장 버튼을 눌러 파일로 저장합니다.



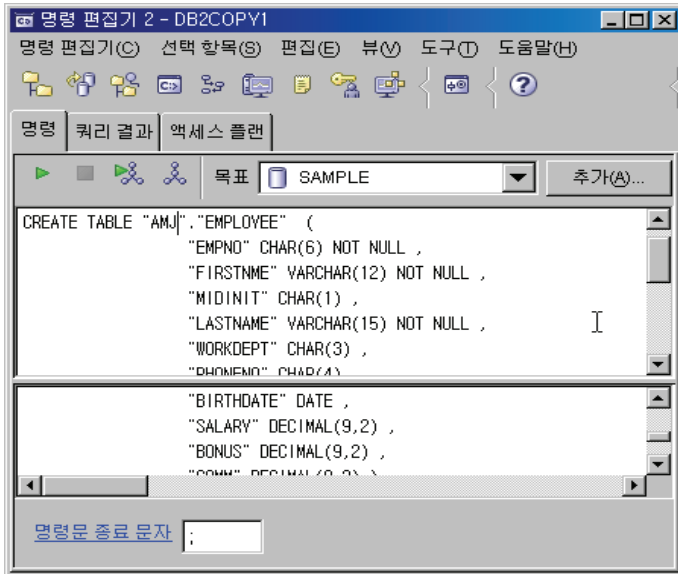
4. 저장 창이 나타나면 저장위치와 파일 이름을 지정한 후 "확인버튼"을 클릭합니다.



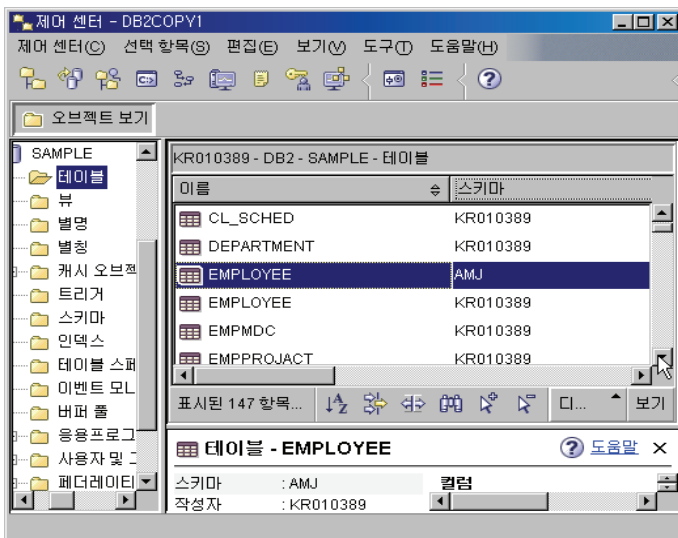
16 제어센타를 이용한 IMPORT

Point  제어센타를 이용하여 DB2 데이터베이스에 있는 테이블로 IMPORT 합니다.

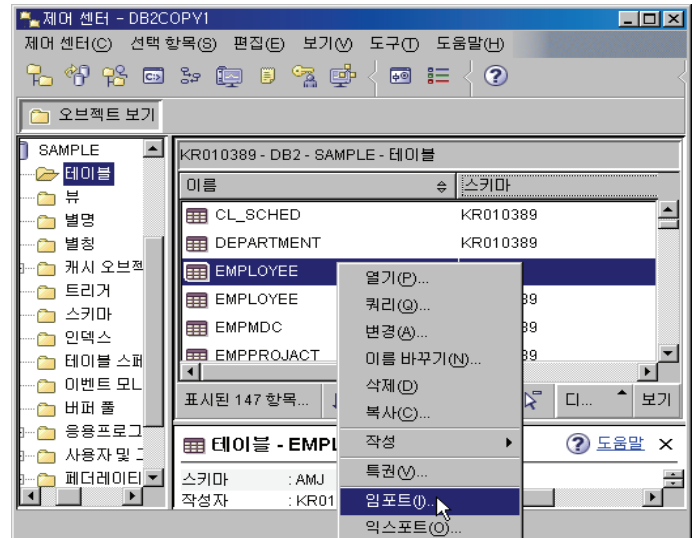
5. DDL 파일을 이용하여 다음과 같이 스키마를 변경한 후 테이블을 생성합니다. 제어센타 메뉴에서 도구를 선택한 후 명령 편집기를 선택합니다. 목표에 추가 버튼을 클릭하여 SAMPLE 데이터베이스를 선택한 후 확인 버튼을 클릭합니다. 앞에서 생성한 파일을 열어 스키마 부분을 변경한 후 아래와 같이 실행하도록 합니다.



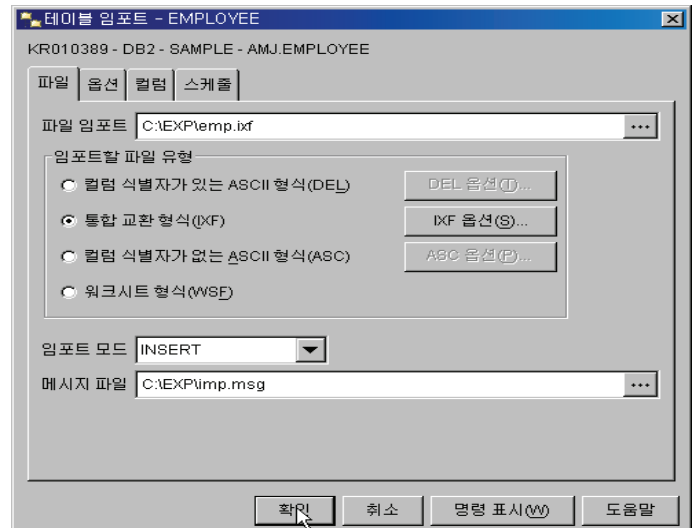
6. 다시 제어센타로 돌아가서 테이블을 선택한 후 오른쪽 마우스를 클릭하여 "새로고침"을 선택합니다. 오른쪽 패널에 AMJ.EMPLOYEE 테이블이 생성되었는지 확인합니다.




7. 새로 생성한 테이블을 선택하여 앞에서 추출한 데이터를 IMPORT 합니다. AMJ.EMPLOYEE 를 선택하여 오른쪽 마우스를 클릭한 후 임포트를 선택합니다.



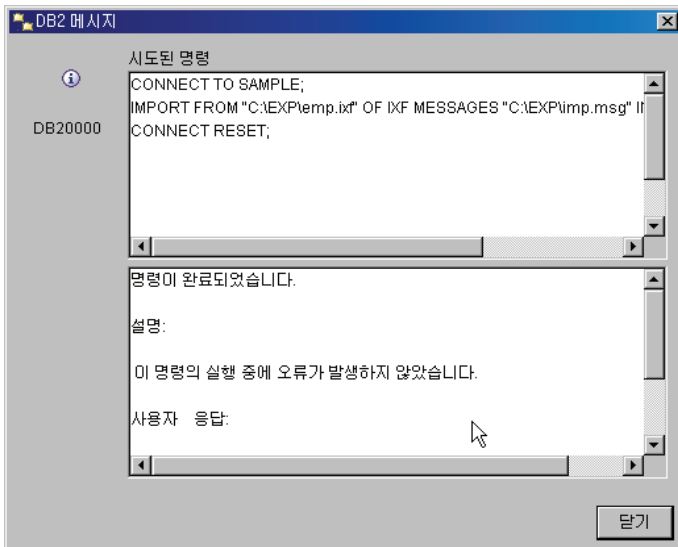
8. 임포트할 파일로 앞에서 export한 파일을 선택합니다. 임포트할 파일 유형을 "통합 교환 형식(IXF)"로 지정하고 임포트 모드를 "INSERT"로 선택한 후 메시지 파일을 아래와 같이 입력합니다. (이곳에서 "CREATE" 모드를 선택할 경우 앞에서 AMJ.EMPLOYEE 테이블을 생성하는 절차는 생략해도 됩니다.)



17 제어센타를 이용한 IMPORT

Point  제어센타를 이용하여 DB2 데이터베이스에 있는 테이블로 IMPORT 합니다.


9. 메시지 창에서 메시지를 확인하여 에러가 발생되었는지 확인합니다.



10. 제어센타에서 AMJ.EMPLOYEE 를 더블 클릭하여 테이블의 데이터가 제대로 삽입되었는지 확인합니다.



18 명령어를 이용한 LOAD

Point  명령어를 이용하여 DB2 데이터베이스에 있는 테이블로 LOAD 합니다.

1. 목표 테이블을 생성하고 기본 데이터를 입력하는 SQL 문을 작성하여, crtab.db2로 저장합니다. 일반 에디터 명령문을 이용하여 다음과 같이 작성합니다.

```
DROP TABLE AMJ.EMP;
DROP TABLE AMJ.DEPT;
CREATE TABLE AMJ.DEPT
(
    id          smallint      not null
  ,name        varchar(20)    not null
  ,budget      int
) ;
ALTER TABLE AMJ.DEPT ADD
    CONSTRAINT dept_pk PRIMARY KEY(id);

CREATE TABLE AMJ.EMP
(
    id          smallint      not null
  ,name        varchar(30)    not null
  ,sex         char(1)
  ,mydept      smallint
  ,salary      smallint
  ,email       varchar(30)    not null
) ;
ALTER TABLE AMJ.EMP
    ADD CONSTRAINT emp_pk PRIMARY KEY(id);
ALTER TABLE AMJ.EMP
    ADD CONSTRAINT emp_uk UNIQUE(email);
ALTER TABLE AMJ.EMP
    ADD CONSTRAINT emp_fk
    FOREIGN KEY(mydept) REFERENCES AMJ.DEPT;
ALTER TABLE AMJ.EMP
    ADD CONSTRAINT emp_cc
    CHECK(sex='M' or sex='F');
```

```
INSERT INTO AMJ.DEPT VALUES
(1,'DB2 SALES',2000), (2,'TECHLINE',3000),
(3,'DB2 FTSS',4000);
INSERT INTO AMJ.EMP VALUES
(1,'모지란','F',1,10,'jrmo@kr.ibm.com'),
(2,'AMJ','F',2,50,'amj@kr.ibm.com'),
(3,'JGL','F',3,20,'jgl@kr.ibm.com'),
(4,'CKC','F',3,30,'ckc@kr.ibm.com');
```

2. 앞에서 작성한 SQL 문을 CLP 창에서 수행하여 테이블을 생성하고 데이터를 삽입합니다.

```
c:\WLABWUNIT10>db2 connect to sample
c:\WLABWUNIT10>db2 -tvf crtab.db2
```

3. 입력할 데이터를 amj.del 인 데이터 파일에 저장합니다. 약 24개의 행을 저장합니다.

```
c:\W>notepad c:\WLABWUNIT10\amj.del
```

```
11 '송재빈','M',1,10,'jbsong@kr.ibm.com'
11 '홍난희','F',2,20,'nhhong@kr.ibm.com'
13 '장동철','m',3,30,'dcjang@kr.ibm.com'
14 , , 'F',3,40,'aka@kr.ibm.com'
15 '한심한','M',2,45,'shhan@kr.ibm.com'
16 '안명찬','M',1,10,'mcahn@kr.ibm.com'
17 '박소자','F',2,20,'sjpark@kr.ibm.com'
18 '하하','M',2,20,'haha@kr.ibm.com'
19 '민순하','m',3,30,'shmin@kr.ibm.com'
20 , , 'F',3,40,'aaa@kr.ibm.com'
21 '정상숙','M',4,45,'ssjung@kr.ibm.com'
22 '김윤숙','M',5,10,'yskim@kr.ibm.com'
23 '허정민','M',1,10,'jmhu@kr.ibm.com'
24 '김건자','F',2,20,'kjkim@kr.ibm.com'
25 '지성','m',3,30,'js@kr.ibm.com'
26 , , 'F',3,40,'jslee@kr.ibm.com'
27 '이화','F',2,45,'leeh@kr.ibm.com'
28 '모순자','M',2,10,'sjmo@kr.ibm.com'
29 '서희','M',1,10,'hees@kr.ibm.com'
30 '이서희','F',2,20,'hees@kr.ibm.com'
31 '유재민','m',3,30,'jmyu@kr.ibm.com'
32 , , 'F',2,40,'amj@kr.ibm.com'
33 '김준호','M',3,45,'jhkim@kr.ibm.com'
'34','이성호','M',4,45,'shlee@kr.ibm.com'
```


부연설명 :

위의 파일을 로드할 경우 로드되지 않는 행은 다음과 같습니다.

ID 11번은 Primary Key 위반, 30번은 email 이 29번과 동일하므로 Unique 위반으로 EXCEPTION 테이블에 기록되며 2개의 행이 삭제됩니다.

14,20,26,32는 2번 Column의 NULL 값 입력 불가능, 34번은 SMALLINT Column에 Char 데이터를 입력하므로 DUMP 파일에 기록됩니다. 총 5개의 행이 거부되고 DUMP 파일에 기록됩니다.

19 명령어를 이용한 LOAD

Point  명령어를 이용하여 DB2 데이터베이스에 있는 테이블로 LOAD 합니다.

4. 추가되지 못한 행들을 확인하기 위해 예외 테이블인 amj.empexp 를 미리 생성합니다. crexp.db2 를 CLP 창에서 수행합니다.

```
c:\WLABWUNIT10> notepad crexp.db2
```

```
DROP TABLE AMJ.EMPEXP;
CREATE TABLE AMJ.EMPEXP LIKE AMJ.EMP;
ALTER TABLE AMJ.EMPEXP ADD COLUMN ts
TIMESTAMP;
ALTER TABLE AMJ.EMPEXP ADD COLUMN msg
CLOB(32K);
```

```
c:\WLABWUNIT10> db2 connect to sample
c:\WLABWUNIT10> db2 -tvf crexp.db2
```

5. 목표 테이블의 인덱스를 확인합니다.

```
c:\W>db2 describe indexes for table AMJ.EMP show detail
```

인덱스 스키마	이름	규칙	고유한 컬럼 수	컬럼 이름
AMJ	EMP_PK	P	1	+ID
AMJ	EMP_UK	U	1	+EMAIL

6. 목표 테이블의 현재 데이터와 Exception 테이블의 데이터를 확인합니다.

```
c:\W>db2 SELECT id,name,mydept,sex FROM AMJ.EMP
```

```
 ID   NAME   MYDEPT  SEX
  1   모지란     1      F
  2   AMJ      2      F
  3   JGL      3      F
  4   CKC      3      F
4 레코드가 선택되었습니다.
```

```
c:\W>db2 SELECT count(*) FROM AMJ.EMPEXP
```

```
 1
-----
 0
1 레코드가 선택되었습니다.
```

7. amj.del 파일을 읽어 AMJ.EMP 테이블에 데이터를 로드하는 LOAD 명령어를 작성하여 load.db2 에 저장한 후 스크립트를 실행합니다. 또 다른 창에서 "LOAD QUERY TABLE" 명령문을 실행하여 LOAD 상태를 체크합니다. 입력데이터가 많지 않은 관계로 로드가 빨리 끝나 LOAD 최종 상태만 보일 수 있습니다.

```
c:\W>notepad load.db2
```

```
LOAD FROM "c:\WLABWUNIT10\amj.del"
" OF DEL
MODIFIED BY CHARDEL""
DUMPFIL="C:\WLABWUNIT10\amj.dmp"
MESSAGES "C:\WLABWUNIT10\amj.msg"
INSERT INTO AMJ.EMP
FOR EXCEPTION AMJ.EMPEXP;
```

```
c:\W>db2 -tvf load.db2
```

```
읽은 행 수           = 24
건너뛴 행 수        = 0
로드된 행 수         = 19
거부된 행 수         = 5
삭제된 행 수         = 2
커미트된 행 수       = 24
```


SQL3107W 메시지 파일에 적어도 하나의 경고 메시지가 있습니다.

8. "LOAD QUERY TABLE" 명령문을 실행한 결과를 확인합니다.

```
c:\W>db2 query load table amj.emp summaryonly
```

```
SQL3532I 로드 유틸리티가 현재 "DELETE" 단계에 있습니다.
SQL3534I 로드 삭제 단계는 거의 "100"퍼센트 완료되었습니다.
읽은 행 수           = 24
건너뛴 행 수        = 0
로드된 행 수         = 19
거부된 행 수         = 5
삭제된 행 수         = 2
커미트된 행 수       = 24
경고 수              = 5
테이블 상태:
무결성 설정 보류
로드 진행
```


20 명령어를 이용한 LOAD

Point  명령어를 이용하여 DB2 데이터베이스에 있는 테이블로 LOAD 합니다.

9. 로드작업이 완료되면 amj.msg 파일을 열어 테이블에 삽입되지 못한 데이터가 무엇인지, 그 이유가 무엇인지 확인합니다.

c:\W>notepad amj.msg

```
SQL3515W 유틸리티가 "2008-04-14 17:21:45.050601"에서 "LOAD" 단계를 완료했습니다.
SQL3500W 유틸리티가 "2008-04-14 17:21:45.054285"에서 "BUILD" 단계를 시작 중입니다.
SQL3213I 인덱싱 모드는 "REBUILD"입니다.
SQL3515W 유틸리티가 "2008-04-14 17:21:45.195728"에서 "BUILD" 단계를 완료했습니다.
SQL3500W 유틸리티가 "2008-04-14 17:21:45.275520"에서 "DELETE" 단계를 시작 중입니다.

SQL3509W 유틸리티가 테이블에서 "2"개의 행을 삭제했습니다.
SQL3515W 유틸리티가 "2008-04-14 7:21:45.417370"에서 "DELETE" 단계를 완료했습니다.
```

10. amj.dmp.load.X 파일을 열어 테이블에 삽입되지 못한 유효하지 못한 데이터를 확인합니다.

c:\W>notepad amj.dmp.*

```
14 ,      ,F' ,3 ,40 ,'aka@kr.ibm.com'
20 ,      ,F' ,3 ,40 ,'aaa@kr.ibm.com'
26 ,      ,F' ,3 ,40 ,'jslee@kr.ibm.com'
32 ,      ,F' ,2 ,40 ,'amj@kr.ibm.com'
'34','이성호' ,M' ,4 ,45 ,'shlee@kr.ibm.com'
```

11. AMJ.EMPEXP 테이블을 조회하여 고유 인덱스를 위반한 행이 있는지 확인합니다.

c:\W>db2 SELECT id,name,mydept,sex,email
FROM AMJ.EMPEXP

ID	NAME	SEX	EMAIL
30	이서희	F	hees@kr.ibm.com
11	홍난희	F	'nhhong@kr.ibm.com

12. AMJ.EMP 테이블을 조회해 봅니다. 다음과 같이 에러가 발생합니다.

C:\W>db2 select * from amj.emp
ID NAME SEX MYDEPT SALARY

SQL0668N 이유 코드 "1"(으)로 인해 테이블 "AMJ.EMP"에서 조작이 허용되지 않습니다. SQLSTATE=57016

13. 앞에서 발생한 "SQL0668N" 에러와 이유 코드 1의 원인을 파악합니다.

c:\W>db2 ? SQL0668N

테이블이 무결성 설정 보류 권한 없음 상태입니다. 테이블의 무결성이 적용되지 않았으므로 테이블 내용이 유효하지 않을 수 있습니다. 무결성 설정 보류 권한 없음 상태가 아닌 기본 테이블 또는 상위 테이블에 대한 조작은 종속 테이블이 무결성 설정 보류 권한 없음 상태에 있으면 이 오류를 수신할 수 있습니다.

14. 시스템 카탈로그에서 테이블에 대한 정보를 확인하면, 테이블에 대한 제한 조건이 점검되지 않았다는 것을 확인할 수 있습니다.

c:\W>db2 SELECT const_checked FROM SYSCAT.TABLES
WHERE TABSCHEMA='AMJ' AND tablename='EMP'

```
CONST_CHECKED
NNYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
1 레코드가 선택되었습니다.
```

15. '점검 보류' 상태를 해결하려면, SET INTEGRITY 명령어로 외부키와 점검 제한 조건을 위반한 행을 점검합니다. 위반한 행은 예외 테이블에 자동으로 저장됩니다.


c:\W>db2 SET INTEGRITY FOR AMJ.EMP IMMEDIATE
CHECKED FOR EXCEPTION IN AMJ.EMP
USE AMJ.EMPEXP

SQL3602W 데이터 점검 처리시 제한조건 위반이 발견되어 예외 테이블로 이동시켰습니다. SQLSTATE=01603

16. 예외 테이블을 조회하여 외부키와 점검 제한 조건을 위반한 행이 입력되었는지 확인합니다.

c:\W>db2 SELECT id,name,mydept,sex,email
FROM AMJ.EMPEXP

ID	NAME	MYDEPT	SEX	EMAIL
30	이서희	2	F	hees@kr.ibm.com
11	홍난희	2	F	nhhong@kr.ibm.com
13	장동철	3	m	dcjang@kr.ibm.com
19	민순하	3	m	shmin@kr.ibm.com
21	정상숙	4	M	ssjung@kr.ibm.com
22	김윤숙	5	M	yskim@kr.ibm.com
25	지성	3	m	js@kr.ibm.com
31	유재민	3	m	jmyu@kr.ibm.com

Point  다음은 오라클의 SQL*Loader와 DB2의 Load 작업을 수행하는 db2move, IMPORT, LOAD 유틸리티를 비교합니다.

SQL*LOADER 및 db2move 유틸리티 비교

	오라클의 SQL*Loader	db2move / IMPORT / LOAD
설명	<ul style="list-style-type: none"> SQL*Loader 란 외부 파일의 데이터를 ORACLE 의 테이블에 넣기 위한 유틸리티입니다. Control File의 정보를 이용하여 데이터를 적재합니다. 	<ul style="list-style-type: none"> DB2는 IMPORT, LOAD, db2move 유틸리티 모두 외부 파일의 데이터를 DB2 테이블에 넣을 수 있습니다. Control File 이 필요치 않습니다.
공통점	<ul style="list-style-type: none"> 외부 파일을 읽어 데이터를 RDBMS 에 적재합니다. 	
차이점	<ul style="list-style-type: none"> 오라클의 SQL*Loader는 Control File이 존재해야 하나 DB2의 경우에는 필요치 않으며 적재할 대상 및 속도에 따라 유틸리티를 선택할 수 있습니다. 	

예문

오라클

```
SQLldr userid=scott/tiger control='ulcase1.ctl'
log='ulcase1.log' DIRECT=TRUE
```



DB2

```
db2move sample import -io replace -u inst95 -p ****
```

22 db2move 유틸리티를 이용한 예제

Point db2move 유틸리티를 이용하여 데이터베이스 전체 데이터를 추출하고 적재합니다.

1. 시작 => 실행에서 "db2cmd" 명령어를 수행하여 DB2 CLP 창을 엽니다. DB2 CLP 창에서 db2move 명령어를 수행하여 유틸리티의 도움말을 참고합니다. .

```

C:\WEXP>db2move

Application code page not determined, using ANSI codepage 136

-----db2move - database movement tool-----

Usage: "db2move.<dbname>.<action>.<options>"

<dbname> is the name of the database.

<action> is one of:
  EXPORT - Export all tables that meet the filtering criteria
           (If no <options> specified, export all tables).
           Internal staging information is stored in file
           Messages are stored in 'EXPORT.msg'.
  
```

2. db2move 유틸리티를 이용하여 SAMPLE 데이터베이스 전체 테이블을 추출합니다. 현재 디렉토리에 추출한 파일이 생성되므로 디렉토리를 새로 생성한 후 생성한 디렉토리에서 db2move 유틸리티를 수행하는 것이 좋습니다.

```

C:\WEXP>db2move sample export

Application code page not determined, using ANSI
***** DB2MOVE *****

Action: EXPORT

Start time: Tue Mar 04 13:44:32 2008

Connecting to database SAMPLE ... successful! Se

Binding package automatically ... C:\WPROGRA~1\WIBM
successful!

Binding package automatically ... C:\WPROGRA~1\WIBM
successful!
  
```

3. 윈도우의 "dir" 명령어를 이용하여 현재 디렉토리에 새로 생성된 파일들을 확인합니다.

```

C:\WEXP>dir/w
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: F0C9-11EF

C:\WEXP 디렉토리

[.]                [..]                ctemp.ddl          db
emp.ixf             export.msg          EXPORT.out         im
tab1.ixf            tab1.msg            tab10.ixf         ta
tab11.ixf           tab11.msg           tab12.ixf         ta
tab13.ixf           tab13.msg           tab14.ixf         ta
tab14a.001.lob     tab15.ixf           tab15.msg         ta
tab16.msg           tab17.ixf           tab17.msg         ta
tab18.msg           tab19.ixf           tab19.msg         ta
tab2.msg            tab20.ixf           tab20.msg         ta
tab3.msg            tab4.ixf            tab4.msg          ta
tab5.msg            tab6.ixf            tab6.msg          ta
tab7.msg            tab8.ixf            tab8.msg          ta
tab9.ixf            tab9.msg            tab9a.001.lob
  
```

4. 일반 Editor 프로그램을 이용하여 db2move.lst 파일을 열어 내용을 확인합니다. 해당 테이블 데이터가 어떠한 파일로 추출되었는지 알 수 있습니다. 예를 들면 AMJ.EMPLOYEE 파일은 tab4.ixf 파일로 추출되었으며 메시지 파일은 tab4.msg 파일이라는 것을 유추할 수 있습니다.

```

db2move.lst - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

!*"KR010389"."ACT"!tab1.ixf!tab1.msg!
!*"KR010389"."CL_SCHED"!tab2.ixf!tab2.msg!
!*"KR010389"."DEPARTMENT"!tab3.ixf!tab3.msg!
!*"AMJ"."EMPLOYEE"!tab4.ixf!tab4.msg!
!*"KR010389"."EMPLOYEE"!tab5.ixf!tab5.msg!
!*"KR010389"."EHMPDC"!tab6.ixf!tab6.msg!
!*"KR010389"."EHMPROJACT"!tab7.ixf!tab7.msg!
!*"KR010389"."EMP_PHOTO"!tab8.ixf!tab8.msg!
!*"KR010389"."EMP_RESUME"!tab9.ixf!tab9.msg!
!*"SYSTOOLS"."HMOM_ATM_INFO"!tab10.ixf!tab10.msg!
!*"SYSTOOLS"."HMOM_COLLECTION"!tab11.ixf!tab11.msg!
!*"KR010389"."IN_TRAY"!tab12.ixf!tab12.msg!
!*"KR010389"."ORG"!tab13.ixf!tab13.msg!
!*"SYSTOOLS"."POLICY"!tab14.ixf!tab14.msg!
!*"KR010389"."PROJACT"!tab15.ixf!tab15.msg!
!*"KR010389"."PROJECT"!tab16.ixf!tab16.msg!
  
```

23 db2move 유틸리티를 이용한 예제

Point db2move 유틸리티를 이용하여 데이터베이스 전체 데이터를 추출하고 적재합니다.

5. 앞에서 추출한 데이터베이스 전체 데이터를 "TEST"라는 새로운 데이터베이스에 적재하도록 합니다. "TEST" 데이터베이스를 다음과 같이 생성합니다. 만약 "애플리케이션이 다른 데이터베이스에 연결되어 있습니다"라는 에러메시지가 나타나면 terminate 명령어를 이용하여 기존 연결 프로그램 및 백그라운드 프로세스를 중지한 후 다시 데이터베이스 생성 명령어를 수행합니다.

```

C:\WEXP>db2 terminate
DB200001  TERMINATE 명령이 완료되었습니다.

C:\WEXP>db2 create db test
DB200001  CREATE DATABASE 명령이 완료되었습니다.

C:\WEXP>
  
```

6. 일반 EDITOR 프로그램을 사용하여 db2move.lst 파일을 열어 확인한 후 만약 스키마를 변경할 경우 스키마 이름을 변경하여 저장합니다.

```

db2move.lst - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
!*"KR010389"*"ACT"*tab1.ixf!tab1.msg!
!*"KR010389"*"CL_SCHED"*tab2.ixf!tab2.msg!
!*"KR010389"*"DEPARTMENT"*tab3.ixf!tab3.msg!
!*"AMJ"*"EMPLOYEE"*tab4.ixf!tab4.msg!
!*"KR010389"*"EMPLOYEE"*tab5.ixf!tab5.msg!
!*"KR010389"*"EMPMDC"*tab6.ixf!tab6.msg!
!*"KR010389"*"EMPPROJACT"*tab7.ixf!tab7.msg!
!*"KR010389"*"EMP_PHOTO"*tab8.ixf!tab8.msg!
!*"KR010389"*"EMP_RESUME"*tab9.ixf!tab9.msg!
!*"SYSTOOLS"*"HM0N_ATM_INF0"*tab10.ixf!tab10.msg!
!*"SYSTOOLS"*"HM0N_COLLECTION"*tab11.ixf!tab11.msg!
!*"KR010389"*"IN_TRAY"*tab12.ixf!tab12.msg!
!*"KR010389"*"ORG"*tab13.ixf!tab13.msg!
!*"SYSTOOLS"*"POLICY"*tab14.ixf!tab14.msg!
!*"KR010389"*"PROJACT"*tab15.ixf!tab15.msg!
!*"KR010389"*"PROJECT"*tab16.ixf!tab16.msg!
  
```

KR010389를 TEST로 변경하여도 무방합니다.

7. 앞에서 추출한 ixf 파일을 이용하여 새로 생성한 데이터베이스 "TEST"에 적재합니다. 이 때 db2move 는 db2move.lst 파일을 이용하여 어떤 테이블에 어떤 파일을 읽어 적재할 것인지 정보를 얻습니다.

```

C:\WEXP>db2move test import

Application code page not determined, using ANSI codepage 1363

***** DB2MOVE *****
Action:  IMPORT

Start time:  Tue Mar 04 15:20:31 2008

Connecting to database TEST ... successful

Binding package automatically ... C:\WPROGRA~1\IBM\SQLLIB\BND\DB2
successful!

Binding package automatically ... C:\WPROGRA~1\IBM\SQLLIB\BND\DB2
successful!

* IMPORT: table "KR010389"."ACT"
  -Rows read:      18
  -Inserted:      18
  -Rejected:       0
  -Committed:     18
  
```

8. EMPMDC 테이블에서 SQL3311N 에러 메시지와 함께 오류가 발생된 것을 확인할 수 있습니다. "db2 ? SQL3311N" 명령어를 수행하여 에러의 발생원인을 확인합니다. 아래와 같이 SQL3311N의 에러 발생 원인 및 이유 코드 3에 대해 설명되어 있습니다. "MDC 키가 저장되지 않았습니다."

```

C:\WEXP>db2 ? SQL3311N

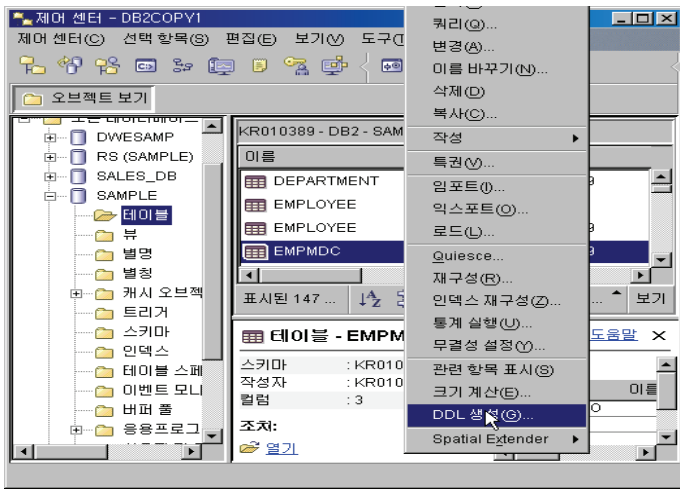
SQL3311N 이 PC/IXF 파일은 임포트 CREATE 모드에서 지원되지
  합니다. 이유 코드 = "<reason-code>".

설명:
이유 코드:
1   인덱스 컬럼 이름에 16진수 값 0x2B 또는 0x2D가 있는
    정보가 저장되지 않았습니다.
2   XML 컬럼은 지원되지 않습니다.
3   MDC 키가 저장되지 않았습니다.
4   테이블 파티션 키가 저장되지 않았습니다.
  
```

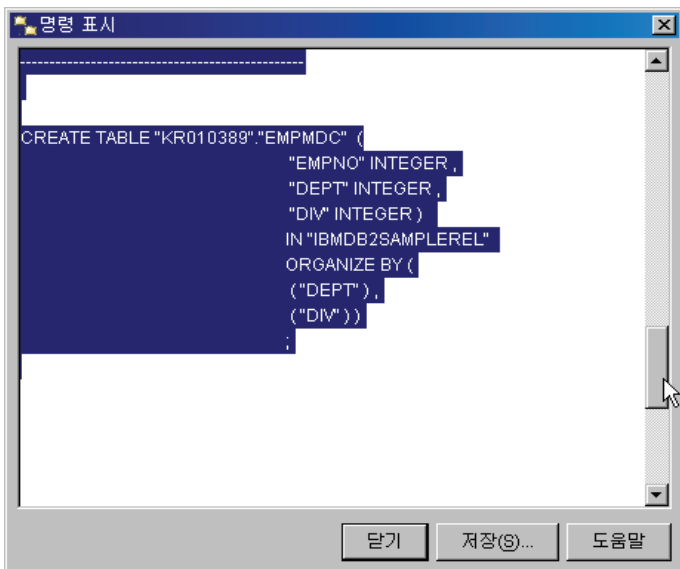
24 db2move 유틸리티를 이용한 예제

Point db2move 유틸리티를 이용하여 데이터베이스 전체 데이터를 추출하고 적재합니다.

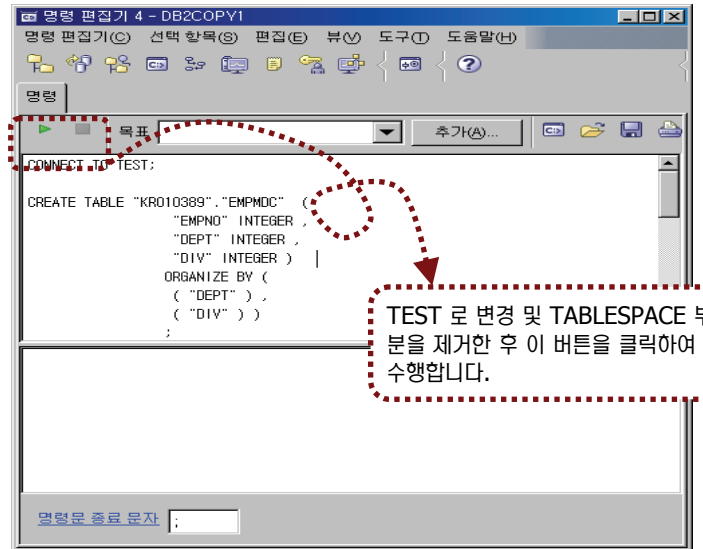
9. SQL3311N 에러 및 아우 코드 2는 PC/IXF 파일의 경우 대상 테이블에 테이블이 존재하지 않아도 데이터를 생성후 데이터를 적재할 수 있습니다. 이러한 모드를 "CREATE" 모드라 하는데 MDC 테이블의 경우에는 PC/IXF 파일 형태의 경우에도 대상 테이블이 존재해야 데이터를 적재할수 있는 경우입니다. 앞에서 수행한 것처럼 EMPMDC 의 DDL 파일을 생성하여 TEST 데이터베이스에 생성합니다. 제어센터에서 SAMPLE 데이터베이스를 선택 => 테이블 선택 => 오른쪽 패널에서 EMPMDC 를 선택한 후 오른쪽 마우스를 클릭하여 DDL 생성을 선택합니다.



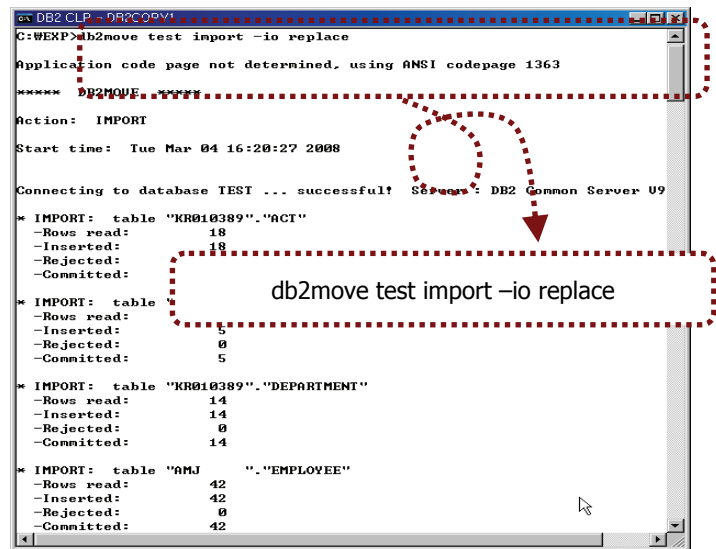
10. 다음과 같이 명령 표시가 나타나면 EMPMDC 생성문을 복사합니다.




11. 현재 SAMPLE 데이터베이스의 데이터들을 TEST 데이터베이스로 이동하는 것이므로 앞에서 복사한 명령문 중 "CONNECT TO SAMPLE" 을 "CONNECT TO TEST"로 변경하고 TABLESPACE가 지정되었을 경우 IN 부분을 아래와 같이 삭제한 후 생성합니다.



12. 다시한번 db2move 유틸리티를 이용하여 적재하도록 합니다. 그러나 이번에 IMPORT 모드는 REPLACE로 지정하여 기존 데이터와 적재하고자 하는 데이터가 동일할 경우 덮어쓰는 모드로 지정합니다. EMPMDC 를 포함하여 모든 테이블에 데이터가 적재된 것을 확인할 수 있습니다.

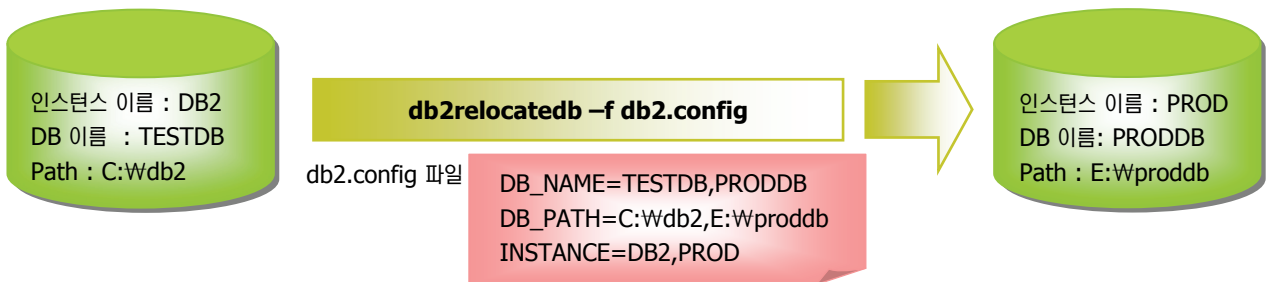


Point  다음과 같이 db2relocatedb 유틸리티 및 split mirror 를 이용하여 데이터를 이동할 수 있습니다.

db2relocatedb 유틸리티

db2relocatedb 유틸리티

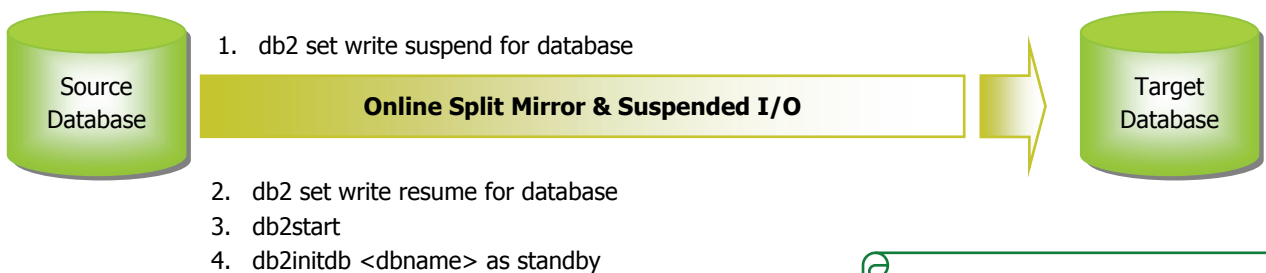
- 데이터베이스를 물리적으로 이동 및 복사합니다.
- 백업 또는 복구가 요구되지 않습니다.
- 오프라인으로 수행됩니다.
- 데이터베이스 이름, Path , 인스턴스 이름, Log Pat, 테이블 스페이스 컨테이너 이름등을 변경할 수 있습니다.



새로운 서버로 재배포하기 위한 절차 :

1. 수동으로 DB 디렉토리를 새로운 서버로 복사합니다.
2. 다음과 같이 목표 위치로 파일들을 복사합니다.
 - copy C:\Wdb2\Wdb2\WNODE0000\SQL00001 E:\Wproddb\Wproddb2\WNODE0000\SQL00001
3. db2relocatedb 명령어를 새로운 서버에서 수행합니다.

SplitMirror



가장 빠른 방법입니다.

- 즉각적인 DB의 복제가 가능.
- 백업이 필요치 않습니다.

📍 데이터 이동 관련 Q/A

- 📍 오라클 import 와 export 는 DB 전체를 Import , Export 할 수 있습니다. DB2의 Import 와 export 는 테이블별로 데이터를 이동하는 것 같은데 오라클과 같이 하려면 어떻게 해야되는건가요?
앞서 예제에서도 보여드린 것처럼 db2move 유틸리티를 이용하면 전체 데이터베이스의 데이터를 이동할 수 있습니다.
- 📍 오라클은 DIRECT PATH 와 Conventional Path 가 존재하여 빨리 추출하고 데이터를 로드할 수 있습니다. DB2도 이러한 옵션이 있는지요?
DB2는 안타깝게도 이러한 옵션은 존재하지 않습니다 .다만 Data 적재시 LOAD 를 이용할 경우 데이터 페이지에 직접 기록함으로써 속도가 더 빨라집니다.
- 📍 오라클의 SQL*Loader 이용하여 데이터를 로드할 때 제어 파일이 필요합니다. DB2도 db2move 를 이용하여 데이터를 로드할 경우 이러한 제어 파일이 필요한가요?
db2move 유틸리티는 제어파일이 필요 없습니다. 다만 db2move 로 export 할때 생성되는 db2move.lst 파일이 필요합니다. db2move.lst 파일에는 각 파일이 어떤 테이블에 삽입될 것인지에 대한 정보가 들어 있습니다. 이 파일을 이용하여 데이터를 로드합니다.



Unit 11 : 백업 및 복구 비교

오라클 사용자를 위한 DB2 가이드

로깅 방식 비교

DB2의 로깅 방식 설정 방법

백업 비교

DB2 백업 소개

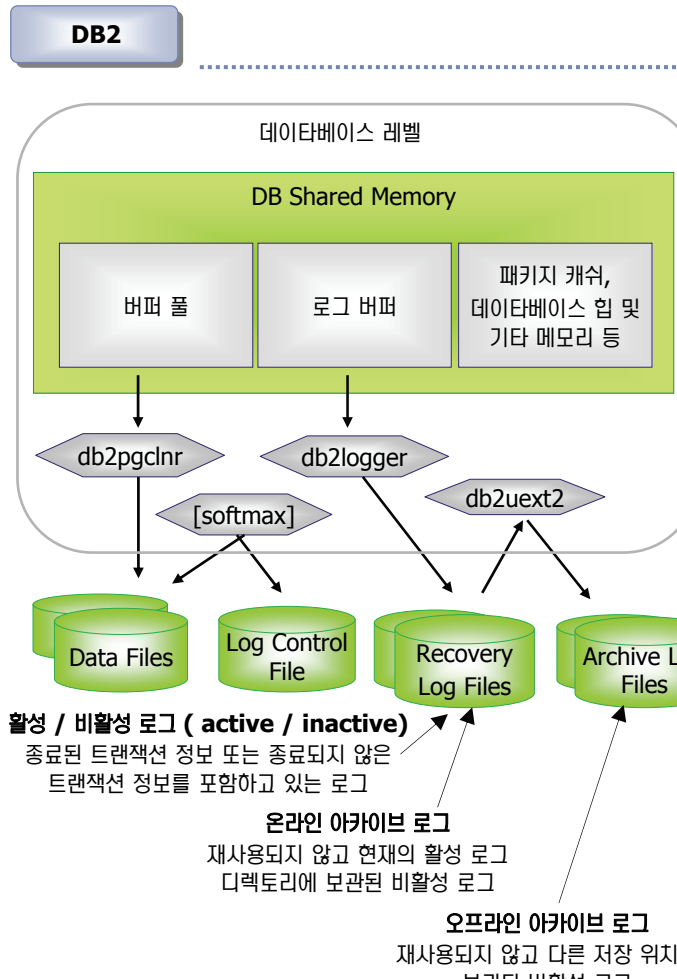
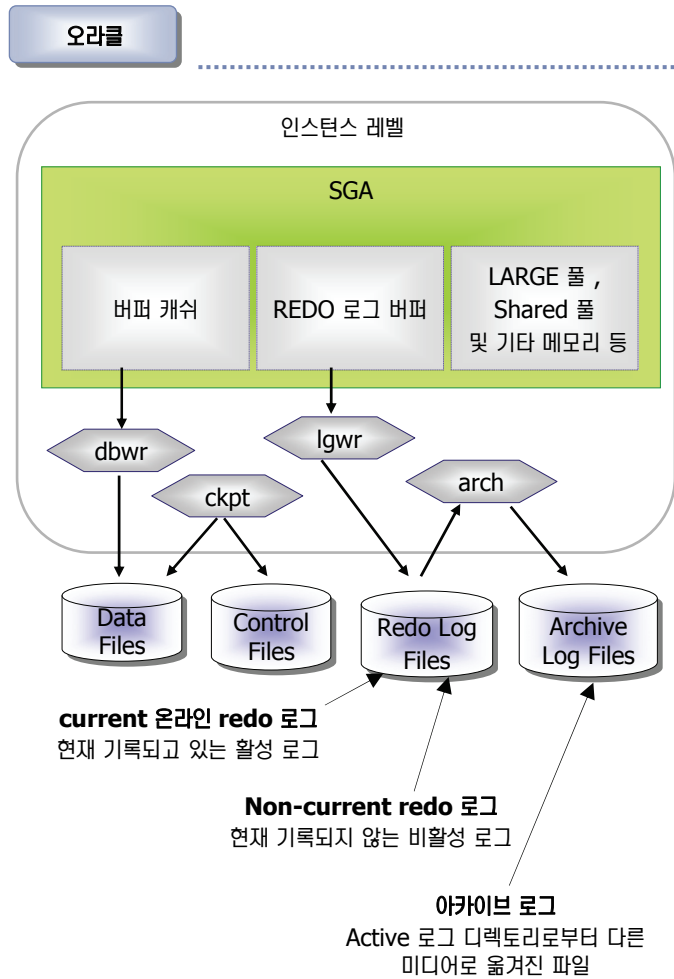
복구 비교

DB2 복구 소개



01 로깅 비교

Point 로깅 파일은 데이터의 모든 변경사항을 기록하여 향후 데이터베이스 복구를 위해 사용됩니다. 다음은 오라클과 DB2에서 사용되는 로깅 방법과 로그 파일의 종류를 비교합니다.



로깅 비교

오라클 로깅 프로세스	DB2 로깅 프로세스
<ul style="list-style-type: none"> lgwr 프로세스는 redo buffer 에 있는 변경된 사항 및 트랜잭션을 redo log 파일에 기록합니다. ckpt 프로세스는 데이터베이스의 정합성을 위해 control 파일과 데이터 파일에 각 트랜잭션별로 부여된 순차적 번호인 SCN 중 마지막 SCN (System Change Number)를 기록합니다. Archiver 는 온라인 redo log 파일을 아카이브 로그 파일로 복사합니다. 	<ul style="list-style-type: none"> db2logger 는 로그 버퍼 중 종료된 트랜잭션을 로그 파일에 기록합니다. softmax 는 데이터베이스 파라미터로 Crash Recovery를 위해 필요한 로그 파일의 수를 지정하며 page cleaner를 일으켜 지정된 것보다 더 오래된 페이지는 디스크에 기록하도록 합니다. db2uext2는 DB2에서 제공하는 archive로 TSM (Tivoli Storage Manager), tape media, 또는 OS 의 copy 명령어를 이용하여 archive 합니다.
오라클 리두 로그 버퍼	DB2 로그 버퍼
<ul style="list-style-type: none"> REDO 로그 버퍼는 변경된 데이터 또는 트랜잭션이 기록되는 장소로 이 공간이 가득 차거나 트랜잭션이 COMMIT 되면 로그 파일로 반영됩니다 	<ul style="list-style-type: none"> 로그 버퍼는 변경된 데이터 또는 트랜잭션이 기록되는 장소로 이 공간이 가득 차거나 트랜잭션이 COMMIT 되면 로그 파일로 반영됩니다.
오라클 로그 제어 파일	DB2 로그 제어 파일
<ul style="list-style-type: none"> 가장 높은 SCN (System Change Number)를 control 파일에 기록하여 복구 과정에 사용합니다. control 파일에는 백업에 대한 정보가 저장되어 있어 RMAN 이용시 사용됩니다. 	<ul style="list-style-type: none"> 시스템 장애후 데이터 베이스가 재시작할때 인스턴스는 로그 파일에 저장된 트랜잭션 정보를 적용하여 데이터 정합성을 보장합니다. 로그 파일의 어느 레코드가 적용될 필요가 있는지에 대한 정보는 Log Control 파일인 SQLOGCTL.LFH.1 과 SQLOGCTL.LFH.2 에 저장됩니다. Log Control파일이 두개 존재하여 한 파일이 깨질 경우 다른 파일이 이용됩니다.

02 로깅 방식 비교

Point DB2와 오라클이 로그 파일을 기록하는 방법에는 순환 로깅과 아카이브 로깅 방식이 있습니다. 다음은 로깅 방식을 비교합니다.

로깅 방식 비교

오라클	DB2	설명
NoArchive Mode	Circular 로깅 (순환 로깅)	<ul style="list-style-type: none"> 변경된 트랜잭션의 이력 데이터를 저장하지 않으며 로그 파일이 재사용되므로 추가적인 저장 공간이 필요하지 않습니다. 백업을 받은 시점까지만 복구가 가능합니다. 오프라인 백업만 지원되며 온라인 백업이 지원되지 않습니다. 전체 데이터베이스 백업 및 복구만 가능합니다. Crash Recovery 에 사용됩니다.
Archive Mode	Archive 로깅	<ul style="list-style-type: none"> 변경된 트랜잭션의 이력 데이터를 저장하며 Archived 로그를 위해 저장 공간이 계속적으로 필요합니다. 온라인 백업이 가능하며 장애가 발생한 시점까지 완전 복구가 가능합니다. 데이터베이스, 테이블 스페이스, 데이터 파일(오라클의 경우) 등을 백업할 수 있습니다.

로그 관련 파라미터 지정 방법 비교

	오라클	DB2 로깅 전환 및 확인 방법
로깅 모드 설정 방법	<ul style="list-style-type: none"> LOG_ARCHIVE_START=TRUE 	<ul style="list-style-type: none"> UPDATE DB CFG FOR <DB 명> USING LOGARCHMETH1 DISK:C:WARCH_LOGS
로그 파일 위치 지정	<ul style="list-style-type: none"> LOG_ARCHIVE_DEST=/oracle/arch 	<ul style="list-style-type: none"> UPDATE DB CFG FOR <DB 명> USING NEWLOGPATH /log
로그 파일 이름 규칙	<ul style="list-style-type: none"> LOG_ARCHIVE_FORMAT=arch11_%s.log 	<ul style="list-style-type: none"> 로그파일의 이름은 S0000000.LOG, S0000001.LOG 등 자동으로 순차적으로 부여됩니다.
설정 방법	<ul style="list-style-type: none"> init.ora 파일의 파라미터인 위의 파라미터를 지정합니다. 	<ul style="list-style-type: none"> 위와 같은 데이터베이스 구성 파라미터를 설정하여 아카이브 로그 위치 및 활성 로그 위치등을 지정합니다.


로그 관련 명령어 비교

	오라클	DB2 로깅 전환 및 확인 방법
로그 모드 확인 방법	<ul style="list-style-type: none"> "ARCHIVE LOG LIST" 명령어 또는 v\$DATABASE 뷰를 통해 NOARCHIVE 모드인지 ARCHIVE 모드인지 확인합니다. 	<ul style="list-style-type: none"> Database Configuration 파라미터인 LOGARCHMETH1 이 OFF 가 아닌 값인지 확인합니다. GET DB CFG FOR <DB 명> grep LOGARCHMETH1

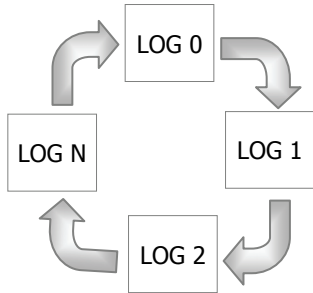
로그 파일 미러링 및 아카이빙 비교

오라클 로그 파일 미러링	DB2 로그 파일 미러링
<ul style="list-style-type: none"> REDO LOG Group 을 생성하여 한 그룹에 여러 파일을 두어 REDO Log 파일을 미러링합니다. 	<ul style="list-style-type: none"> Database Configuration 파라미터인 LOGMIRRORPATH 를 지정하여 Log 파일의 복사본을 저장할 파일 시스템을 지정합니다. UPDATE DB CFG FOR <DB 명> USING MIRRORLOGPATH /mirrorlog
오라클 로그 아카이빙	DB2 로그 아카이빙
<ul style="list-style-type: none"> 오라클내에서 로그 파일을 아카이빙하기 위한 방법은 없습니다. 아카이브 프로세스가 사용되어 현재 Active 로그를 아카이빙 합니다. 	<ul style="list-style-type: none"> DB2는 다음과 같이 다양한 방법을 통해 로그를 아카이브합니다. DB2에서 제공하는 UserEXIT 프로그램을 사용한 자동 아카이빙 Tivoli Storage Manager 를 이용한 아카이빙 3rd Party 제품을 이용한 아카이빙

03 Circular 로깅 방식 비교

Point  오라클과 DB2는 고정된 개수의 로그 파일을 순환하며 재사용하는 순환 로깅 방식을 지원합니다. 오라클에서는 NOARCHIVE, DB2 에서는 Circular 로깅이라 합니다.

오라클



- 1 init.ora 파일의 파라미터인 LOG_ARCHIVE_START 값을 FALSE 로 설정합니다.

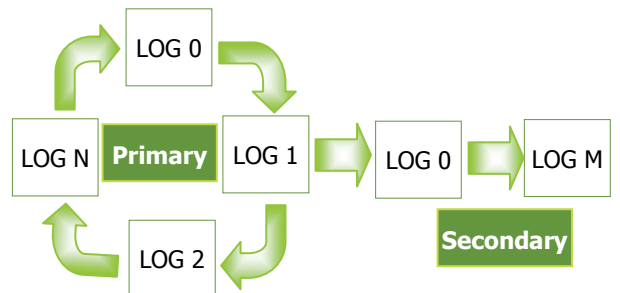
```
LOG_ARCHIVE_START=FALSE
```

- 2 데이터베이스 생성시 또는 변경시 설정된 LOGFILE의 정보를 읽어 생성합니다.

```
ALTER DATABASE ADD LOGFILE GROUP 10 ('/log/log1c.rdo', '/log/log2c.rdo');
```

- 3 UOW가 시작되면 0번 로그는 활성 로그가 되어 기록을 시작하고 0번을 모두 채우면 1번이 활성 로그가 되어 0번 로그에 이어서 기록을 시작합니다.
- 4 활성 로그에 있는 트랜잭션이 모두 종료되고 디스크의 데이터 파일에 반영이 되면 그 활성 로그는 비활성 로그가 되어 다음번에 재사용됩니다.
- 5 N번 로그가 활성 로그이고 더 이상 기록할 공간이 없으면 다음 번호의 로그 파일을 요청합니다. 이 시점에서 0번 로그가 비활성 로그였다면 N번 로그에 이어 재사용됩니다. 만약 0번 로그도 활성로그일 경우 로그 공간 부족 현상이 발생하면서 진행 중이던 트랜잭션은 모두 롤백됩니다.

DB2




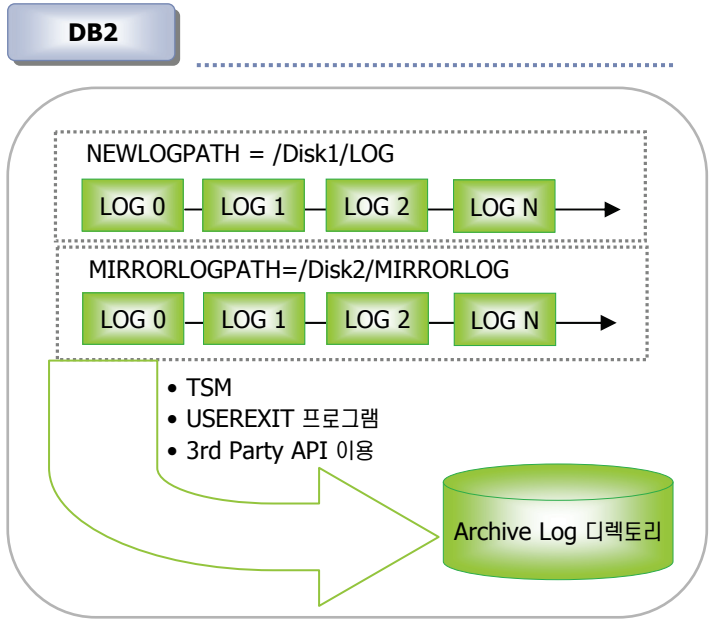
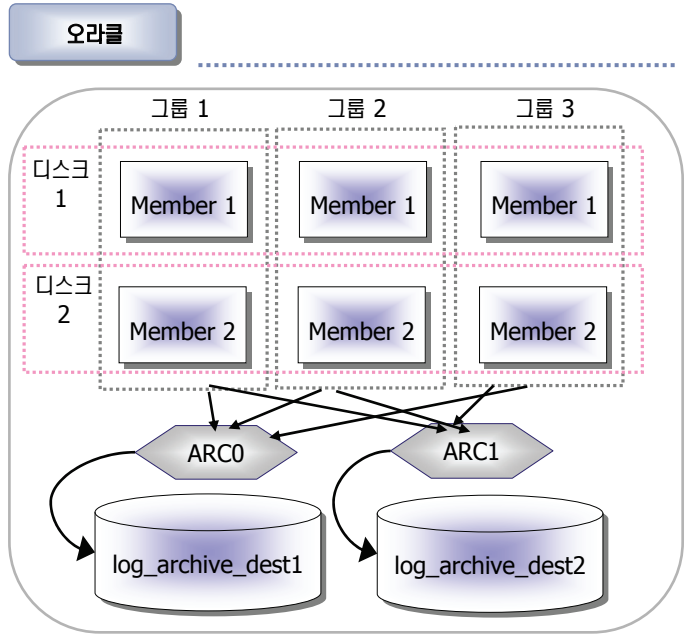
- 1 데이터베이스가 활성화되면 LOGPRIMARY 데이터베이스 구성 변수에 설정된 개수의 로그 파일들이 로그 디렉토리에 새롭게 생성됩니다. 생성된 로그 파일의 번호는 0번 부터 시작합니다.
- 2 데이터베이스 Configuration 파라미터인 LOGARCHMETH1 값을 OFF 로 설정합니다. 디폴트는 OFF 입니다.
Update db cfg for <DB 명> using LOGARCHMETH1 OFF
- 3 UOW가 시작되면 0번 로그는 활성 로그가 되어 기록을 시작하고 0번을 모두 채우면 1번이 활성 로그가 되어 0번 로그에 이어서 기록을 시작합니다.
- 4 활성 로그에 있는 트랜잭션이 모두 종료되고 디스크의 데이터 파일에 반영이 되면 그 활성 로그는 비활성 로그가 되어 다음번에 재사용될 수 있습니다.
- 5 N번 로그가 활성 로그이고 더 이상 기록할 공간이 없으면 다음 번호의 로그 파일을 요청합니다. 이 시점에서 0번 로그가 비활성 로그였다면 N번 로그에 이어 재사용됩니다. 만약 0번 로그도 활성로그일 경우 LOGSECOND 에서 지정한 로그가 사용되어 전체 활성 로그의 수는 N+1이 됩니다.
- 6 LOGSECOND가 지정한 M개의 로그도 모두 활성 로그가 된 상태에서 더 이상 기록할 공간이 없으면 로그 공간 부족 현상이 발생하면서 진행 중이던 트랜잭션은 모두 롤백됩니다. 즉 활성 로그의 최대 개수는 LOGPRIMARY 구성 변수와 LOGSECOND 구성 변수의 합을 초과할 수 없습니다.
- 7 LOGSECOND 로 생성된 M개의 로그들이 비활성 로그가 되면 비동기식으로 엔진에 의해 제거되고 다시 LOGPRIMARY 의 값인 N개의 로그가 남게 됩니다.

참고)

로깅모드를 변경한 후에는 데이터베이스가 BACKUP PENDING 상태가 됩니다. BACKUP PENDING 이 되면 BACKUP 을 수행하여 BACKUP PENDING 에서 빠져 나오도록 합니다.

04 아카이브 로깅 방식 비교

Point  오라클과 DB2는 Online 백업 및 테이블 스페이스 단위 백업, 시점 복구등을 위해 아카이브 로깅 방식을 지원합니다.



1 init.ora 파일의 파라미터인 LOG_ARCHIVE_START 값을 TRUE 로 설정하고 아카이브 로그 파일을 저장할 디렉토리 및 이름 규칙을 지정합니다.

```
LOG_ARCHIVE_START=TRUE
LOG_ARCHIVE_DEST = "C:\WoradbWarchive"
LOG_ARCHIVE_FORMAT = %S.ARC
```

2 데이터베이스 생성시 또는 변경시 설정된 LOGFILE의 정보를 읽어 생성합니다.

```
ALTER DATABASE ADD LOGFILE GROUP 10
('/log/log1c.rdo', '/log/log2c.rdo');
```

3 LGWR은 변경된 데이터를 그룹 1의 Log Member 1에게 기록합니다. 이때 Member 2는 복제본으로 Log Member1에 문제 발생 시 사용됩니다.

4 Archive 프로세스를 통해 로그 파일들이 지정된 archive 디렉토리에 복제됩니다.

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

2 데이터베이스 Configuration 파라미터인 LOGARCHMETH1 값을 OFF 가 아닌 값으로 설정합니다.

```
Update db cfg for <DB 명> using LOGARCHMETH1
DISK:/u/archived_log
```

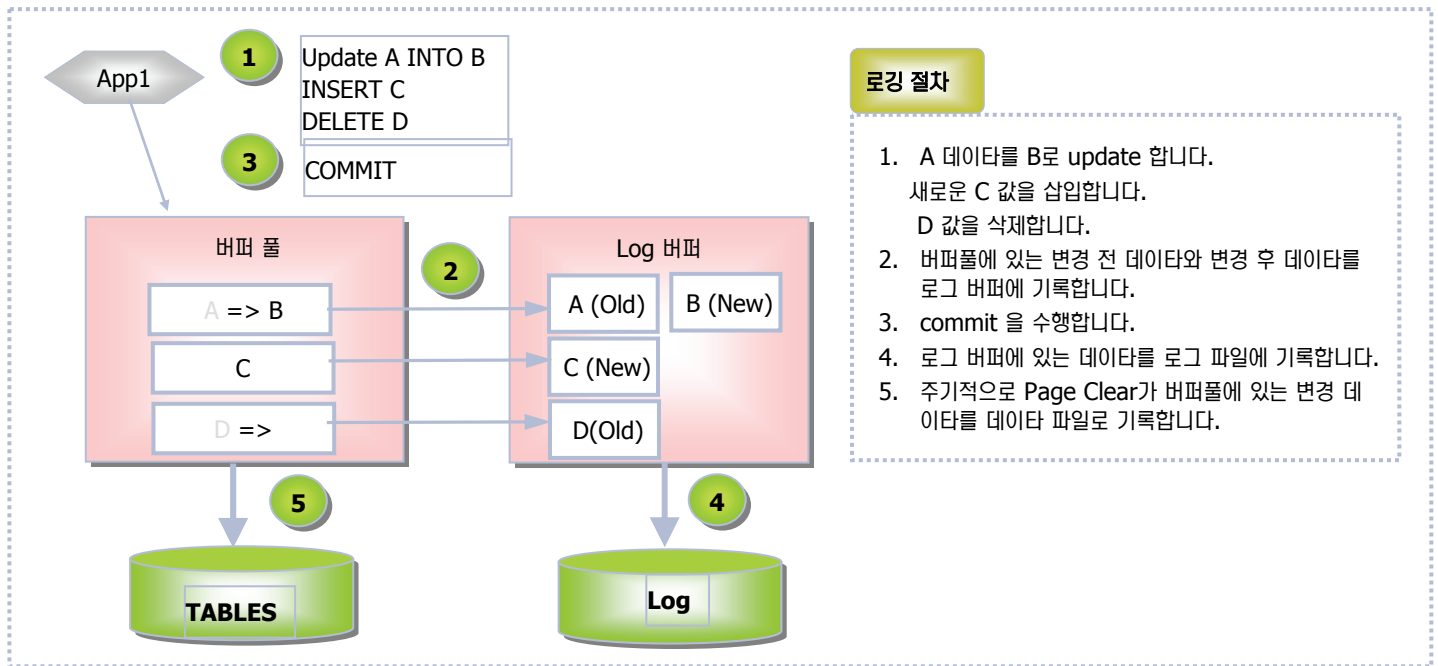
3 LGWR은 변경된 데이터를 NEWLOGPATH 와 MIRRORLOG의 로그 파일에 기록합니다. MIRRORLOG는 로그 파일의 복제본을 저장하는 위치로 NEWLOGPATH에 문제 발생시 사용됩니다.

4 비활성 로그가 생성되면 USEREXIT 프로그램에 의해 자동으로 아카이브 디렉토리로 지정된 /u/archived_log 디렉토리로 이동합니다.

* 부연설명

- 오라클의 LOG GROUP 및 Member 를 이용하여 로그 파일을 미러링 (Multiplexing 이라고 하기도 함)처럼 DB2도 MIRRORLOGPATH를 이용하여 로그 파일을 미러링 합니다. 이는 로깅방식에 상관없이 지정 가능합니다.
- 오라클 아카이빙은 성능에 영향을 줍니다. 즉 redo log 파일은 아카이브 로그로 기록하려는 동안 데이터베이스의 활동을 기다리거나 지연시킬 수 있습니다. 또한 여러 아카이브 디렉토리가 존재한다면 수행해야 할 작업이 더 많아집니다. 이를 위해 오라클에서는 아카이브 프로세스를 여러 개 생성하도록 가이드합니다. 하지만 DB2는 아카이브 프로세스가 존재하지 않고 외부 프로그램을 이용하여 아카이브하므로 성능에 영향을 주지 않으며 또한 여러가지 방식으로 로그 파일을 아카이브할 수 있습니다.

Point 데이터 변경이 일어나면 버퍼풀의 데이터가 변경됩니다. 변경 이전의 값과 변경 이후의 값은 로그 버퍼 및 로그 파일에 기록되어 트랜잭션의 롤백 작업에 사용됩니다.



로깅 파라미터

- 로깅 관련 파라미터는 Database Configuration Parameter 로 다음과 같이 Database Configuration Parameter를 변경하여 적용합니다.
`db2=> update db cfg for <DB 명> using <LOG PARAMETER> <Value>`

로깅 파라미터	설명
LOGPRIMARY	1차 로그의 개수입니다. 데이터베이스의 변경 내용을 기록하며 데이터베이스가 활성화될때 미리 할당됩니다 .
LOGSECOND	2차 로그의 개수입니다. 1차 로그가 모두 사용되면 필요에 따라 한개씩 할당됩니다.
LOGFILSIZ	로그 파일 한 개의 크기로 4K 페이지 단위로 설정합니다.
LOGBUFSZ	로그 파일을 위한 버퍼의 크기로 4K 페이지 단위로 설정합니다.
LOGARCHMETH1, LOGARCHMETH2	비활성 로그의 현재의 로그 디렉토리가 아닌 위치에 아카이브 하도록 합니다. 기본값은 OFF 이며 LOGRETAIN, USEREXIT, DISK, TSM, VENDER 중에서 설정합니다. DISK 를 이용할 경우 아카이브명은 <DISK:절대경로> 의 형태로 설정합니다.
NEWLOGPATH	Active 로그 파일이 위치하는 경로명을 지정합니다.
MIRRORLOGPATH	LOG 파일의 복제본을 저장하는 공간으로 Active Log Path 에 문제 발생시 이 위치에 저장된 로그 파일이 사용됩니다. 한가지 주의 할 점은 NEWLOGPATH 와 MIRRORLOGPATH는 물리적으로 다른 디스크에 놓는 것이 좋습니다.
MINCOMMIT	지정한 개수의 COMMIT 요청을 수행할 때까지 로그 파일에 기록하는 것을 지연합니다. 최대 지연 시간은 1초입니다.
SOFTMAX	응급 복구시에 사용되는 활성 로그의 양을 지정합니다.
CHNGPGS_THRESH	버퍼풀에서 디스크로 반영되지 않은 변경된 페이지의 최대 비율을 설정합니다.

주의 사항) 기존 LOGRETAIN, USEREXIT 파라미터는 LOGARCHMETH 로 대체될 예정입니다. 되도록이면 이 파라미터를 이용하여 Archive 모드로 변경하여 주시기 바랍니다.

Point 명령어를 이용하여 DB2 로깅을 변경합니다. DB2는 Circular 로깅을 디폴트로 사용하며 Archive 모드로 변경시에는 DB Configuration 파라미터를 변경해야 합니다.

1. SAMPLE 데이터베이스에 접속한 후 DB CFG의 파라미터를 확인하여 DB가 현재 Circular 인지 Archive 모드인지 확인합니다.

```
c:\W> db2 connect to sample
c:\W> db2 get db cfg for sample | find "LOGARCHMETH"
```

```
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) = OFF
두 번째 로그 아카이브 메소드(LOGARCHMETH2) = OFF
```

2. 로그 파일에 대한 경로 또는 NEWLOGPATH에서 지정한 경로를 확인하여 활성 로그 파일이 위치하는 장소를 확인합니다.

```
c:\W> db2 get db cfg for sample | find "로그 파일에 대한"
```

```
로그 파일에 대한 변경된 경로(NEWLOGPATH) =
로그 파일에 대한 경로 =
C:\WDB2\WNODE0000\SQL00004\SQLLOGDIR\W
```

3. Archive 로그와 활성화 로그 파일을 저장할 디렉토리를 파일 시스템에 생성한 후 DB CFG의 파라미터인 LOGARCHMETH1과 NEWLOGPATH의 값을 다음과 같이 설정합니다.

```
c:\W> mkdir c:\Warch_logs c:\Wdb2_logs
c:\W> db2 update db cfg for sample using
LOGARCHMETH1 DISK:c:\Warch_logs
c:\W> db2 update db cfg for sample using
NEWLOGPATH c:\Wdb2_logs
```

```
DB20000I UPDATE DATABASE CONFIGURATION 명령이 완료되었습니다.
```

```
SQL1363W 즉각적인 수정을 위해 제출된 하나 이상의 매개변수가 동적으로 변경되지 않았습니다. 이러한 구성 매개변수의 경우, 변경사항이 적용되려면 먼저 이 데이터베이스에서 모든 응용프로그램의 연결을 끊어야 합니다.
```

4. UPDATE DB CFG 후에 SQL1363W 경고 메시지를 받은 경우, 이는 변경값이 동작되지 않았다는 의미이므로 접속된 모든 애플리케이션을 종료하고 데이터베이스를 재활성화합니다.

```
C:\W> db2 force applications all
C:\W> db2 terminate
C:\W> db2 deactivate db sample
```

5. 데이터베이스를 재활성화하면 SQL1116N 에러가 발생합니다.

```
C:\W> db2 activate db sample
```

```
SQL1116N BACKUP PENDING 때문에 데이터베이스 "SAMPLE"에 연결하거나 활성화할 수 없습니다. SQLSTATE=57019
```

6. 로깅 모드를 변경하면 데이터베이스 FULL 백업이 필요합니다. backup 명령어를 실행하여 데이터베이스의 "BACKUP PENDING" 상태를 해제하고 다시 재활성화합니다. 그리고 NEWLOGPATH에 NODE0000 디렉토리 및 파일이 생성되었는지 확인하고 LOGARCHMETH1에는 DB2\W\SAMPLE\NODE0000\WC0000000 디렉토리가 생성되었는지 확인합니다.

```
C:\W> db2 backup db sample
C:\W> db2 activate db sample
C:\W> dir db2_logs
C:\W> dir arch_logs
```

7. 다양한 트랜잭션을 실행시키면 로그 파일에 변경 데이터가 기록되고, 비활성 로그가 되면 자동으로 LOGARCHMETH1 변수가 지정한 경로명으로 이동합니다. DB2는 지정한 경로명 하위 디렉토리로 <INSTANCE명>\W\<DB명>\W\<NODE0000> 디렉토리를 자동으로 생성하며 그 디렉토리에 아카이브 로그 파일을 놓습니다.

```
C:\Warch_logs\WDB2\W\SAMPLE\NODE0000\WC0000000>
```

```
C:\Warch_logs\WDB2\W\SAMPLE\NODE0000\WC0000000 디렉터리
[.] [..] S0000000.LOG S0000001.LOG
S0000002.LOG S0000003.LOG S0000004.LOG
5개 파일 16,502,850 바이트
2개 디렉터리 7,064,027,136 바이트 남음
```

8. 3번에서 설정한 활성 로그 파일이 놓이는 디렉토리로 이동하여 활성 로그가 생성되었는지 확인합니다.

```
C:\W> dir C:\WDB2\WNODE0000\SQL00004\SQLLOGDIR\W
```

```
C:\WDB2\WNODE0000\SQL00004\SQLLOGDIR 디렉터리
[.] [..] S0000005.LOG S0000006.LOG
S0000007.LOG S0000008.LOG S0000009.LOG
5개 파일 20,520,960 바이트
```

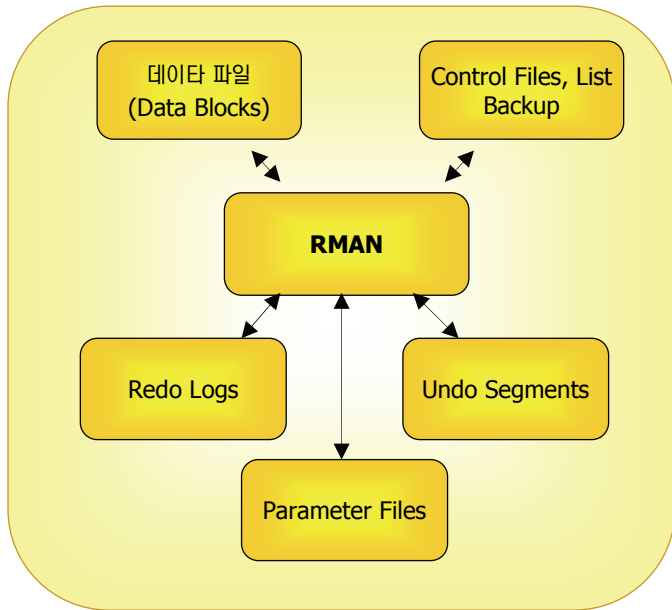
9. 활성 로그 파일의 시작 번호를 확인하면 마지막 아카이브 로그의 번호를 알 수 있습니다.

```
C:\W> db2 get db cfg for sample | find "처음에 사용되는 로그"
```

```
처음에 사용되는 로그 파일 = S0000005.LOG
```

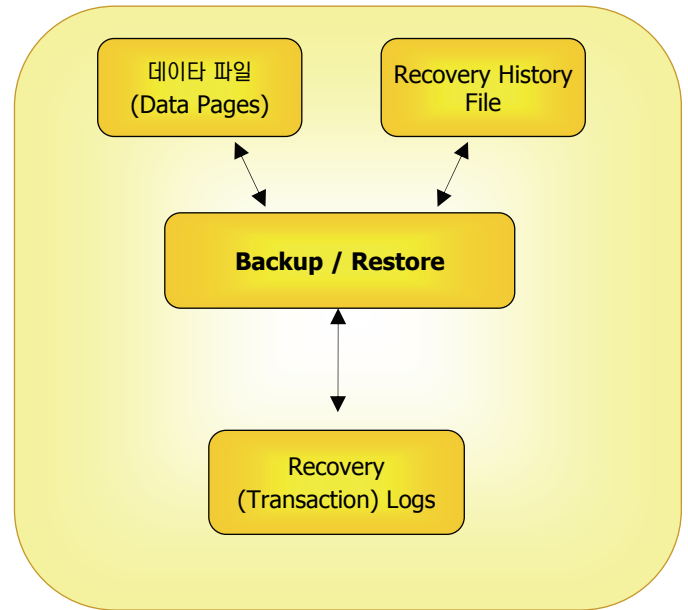
Point 다음은 오라클에서 백업 및 복구를 담당하는 RMAN(Oracle Recovery Manager) 와 DB2의 Backup 및 Restore 유틸리티에 대해서 알아보도록 합니다.

오라클



- RMAN 은 백업 및 복구 과정을 수행하는 툴로 다음과 같은 작업을 수행합니다.
 - 데이터베이스, 테이블 스페이스, 데이터 파일, control 파일, archived log 파일 백업 및 복구
 - 백업 이미지 압축
 - Incremental 백업 수행
 - 로그 백업
 - 백업 이력에 대한 정보 저장

DB2



- DB2 는 백업 및 복구 유틸리티를 통해 백업 및 복구를 수행하며 다음과 같은 작업을 수행합니다.
 - 데이터베이스, 테이블 스페이스 백업 및 복구
 - 백업 이미지 압축
 - Incremental 백업 수행

* 부연설명


- 오라클은 Version 9i 이전에는 RMAN 툴이 존재하지 않아 백업 및 복구에 OS 명령어를 이용하여 백업 및 복구를 수행하였습니다. 여전히 OS 명령어를 이용한 백업 및 복구가 존재하지만 DBA의 수동 작업이 많아 현재는 RMAN을 많이 사용하고 있습니다.
- 오라클의 RMAN에는 백업 이력에 대한 정보를 저장하는 공간이 있어 백업에 대한 이력 데이터를 관리합니다. DB2 역시 백업 및 복구, 로드등 데이터베이스의 작업 내역을 파일로 저장하여 관리합니다. DB2에서는 "list history all for <DB 명>"을 통해 작업 내역을 확인할 수 있습니다.

Point 데이터베이스에서 가장 중요한 기능중의 하나는 백업입니다. 이는 향후 데이터베이스의 문제점 발생시 복구에 대한 대비책으로 다음은 오라클과 DB2에서 사용되는 백업 방법을 비교합니다.

	오라클	DB2
백업 종류 비교	<ul style="list-style-type: none"> 오프라인 백업 (오라클에서는 Cold 백업이라고도 불림) <ul style="list-style-type: none"> 데이터베이스에 접속된 사용자가 없는 상태에서 백업하는 모드로 백업이 진행되는 동안 데이터의 변경은 발생되지 않습니다. 온라인 백업 (오라클에서는 Hot 백업이라고도 불림) <ul style="list-style-type: none"> 사용자가 데이터베이스에 접속하고 있는 상태에서 백업하는 모드입니다. 백업이 진행되는 동안 데이터가 변경 중일 수도 있으므로, 백업 이미지 파일과 백업이 진행되는 동안 변경된 데이터에 대한 로그 파일을 함께 보관해야 복구에 사용할 수 있습니다. 아카이브 로깅에서만 지원됩니다. 	
백업 대상 비교	<ul style="list-style-type: none"> 데이터베이스 전체 테이블 스페이스 데이터 파일, Control 파일, ARCHIVE LOG, SPFILE 등 	<ul style="list-style-type: none"> 데이터베이스 전체 테이블 스페이스
백업 범위	<ul style="list-style-type: none"> Full <ul style="list-style-type: none"> 데이터베이스의 모든 데이터와 제어 정보를 백업합니다. Cumulative Incremental Backup (Level 0 ~ Level 4) <ul style="list-style-type: none"> Level 에 따라 최근의 Full 백업 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다. Differential Incremental Backup (default) <ul style="list-style-type: none"> Level 에 따라 최근의 FULL 백업 또는 Incremental 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다. 	<ul style="list-style-type: none"> Full <ul style="list-style-type: none"> 데이터베이스의 모든 데이터와 제어 정보를 백업합니다. Incremental <ul style="list-style-type: none"> 최근의 Full 백업 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다. 오라클의 cumulative incremental backup와 유사합니다. Delta <ul style="list-style-type: none"> 최근의 Full 백업 또는 Incremental 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다. 오라클의 Differential incremental backup 와 유사합니다.
백업 이미지 압축	<ul style="list-style-type: none"> 백업 이미지 압축이 가능합니다. <pre>RMAN> 2> BACKUP AS COMPRESSED BACKUPSET 3> DATABASE;</pre>	<ul style="list-style-type: none"> 백업 이미지 압축이 가능합니다. <pre>db2=> BACKUP DATABASE SAMPLE USE TSM COMPRESS WITHOUT PROMPTING</pre>
백업 방법	<ul style="list-style-type: none"> OS COPY 명령어를 이용하여 이미지를 복사합니다. RMAN의 백업 명령어를 이용하여 백업 Set 을 생성하거나 이미지 복사를 수행합니다. 	<ul style="list-style-type: none"> 백업 명령어를 이용하여 백업 이미지를 생성합니다.
백업 권한	<ul style="list-style-type: none"> SYSDBA 권한이 필요합니다. 	<ul style="list-style-type: none"> SYSADM, SYSCTRL 또는 SYSMANT 권한이 필요합니다.
백업 관리	<ul style="list-style-type: none"> RMAN 을 이용하여 기존 백업 내역을 관리할 수 있습니다. <pre>RMAN> 2> LIST BACKUP; SQL> SELECT * FROM V\$BACKUP_FILES;</pre>	<ul style="list-style-type: none"> 복구 기록 파일을 이용하여 데이터베이스에 대한 백업, 복구, 로드 등의 작업에 대한 내역을 관리할 수 있습니다. <pre>db2=> list history backup all for <DB 명> db2=> SELECT * FROM table(sysproc.admin_list_hist()) as history</pre>
백업 이미지 저장장소	<ul style="list-style-type: none"> RMAN의 CHANNEL 을 할당하거나 또는 CONFIGURE 명령어를 이용하여 지정합니다. <pre>RMAN> ALLOCATE CHANNEL ch1 TYPE DISK;</pre>	<ul style="list-style-type: none"> BACKUP 수행시 TSM 또는 테이프, 3rd Party 제품을 이용할 것인지 지정합니다. <pre>db2=> backup db sample to /dev/rmt0 db2=> backup db sample use tsm open 2 sessions</pre>
백업 이미지 체크	<ul style="list-style-type: none"> DBVERIFY 를 이용해 데이터 파일에 오류가 있는지 확인합니다. 	<ul style="list-style-type: none"> "db2ckbcp" 명령어를 이용해 백업 이미지가 Restore 될 수 있는지 확인합니다.



오라클의 백업 방법에는 OS copy 명령어를 이용한 이미지 복사와 RMAN 백업 명령어를 이용한 이미지 백업 Set 이미지 생성 방법 두가지가 존재합니다. 그러나 이미지 복사는 OS의 Copy 명령어를 이용한 복사로 데이터 파일의 빈 Data Block 까지 복사하므로 공간 압축 효율성이 떨어지며 수동 작업이 수행되어야 하므로 불편하고 DBA 개입없는 백업이 불가능합니다. DB2 또한 OS Copy 명령어를 이용하여 백업을 수행할 수는 있으나 권고하지는 않습니다.

Point  데이터베이스에서 가장 중요한 기능중의 하나는 백업입니다. 이 곳에서는 오라클과 DB2에서 사용되는 백업 방법을 비교합니다.

오프라인/온라인 데이터베이스 백업 절차 비교

오라클	DB2
<p>1 sysdba 권한을 가지고 있는 사용자로 접속하여 NORMAL 모드로 SHUTDOWN 합니다.</p> <pre>SQL> shutdown normal;</pre> <p>2 OS Copy 명령어를 이용하여 데이터 파일, Control 파일, redo log 파일을 복사하거나 또는 RMAN 을 이용하여 백업합니다.</p> <pre>RMAN> BACKUP DATABASE;</pre>	<p>1 SYSADM 권한을 가진 사용자가 다음 명령어를 수행하여 데이터베이스에 접속한 애플리케이션을 모두 끊습니다.</p> <pre>db2> force applications all</pre> <p>2 backup 명령어를 이용하여 백업을 수행합니다.</p> <pre>db2> backup db <DB 명> [online] [include logs]</pre> <p>온라인 백업의 경우 복구를 위해 백업을 수행하는 동안 연관된 로그 파일들도 함께 보관해야 복구를 완료할 수 있습니다.</p>

Incremental 백업 절차 비교

오라클	DB2
<p>1 sysdba 권한을 가지고 있는 사용자로 접속하여 NORMAL 모드로 SHUTDOWN 합니다.</p> <pre>SQL> shutdown normal;</pre> <p>2 마지막 백업 이후에 변경된 데이터를 추적할 수 있도록 트래킹 모드를 활성화시킵니다.</p> <pre>SQL> alter database enable block change tracking using file '/rman_bkups/change.log';</pre> <p>3 INCREMENTAL, CUMULATIVE 등 다양한 옵션을 이용하여 최근의 FULL 백업 또는 INCREMENTAL 백업 이후에 변경된 부분에 대해서 백업 이미지를 생성합니다.</p> <pre>RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE; RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE; RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;</pre>	<p>1 아카이브 로깅 모드에서만 지원되므로 LOGARCHMETH1 값을 OFF 가 아닌 다른 값으로 설정합니다.</p> <pre>db2> update db cfg for <DB 명> using LOGARCHMETH1 DISK:<경로명></pre> <p>2 TRACKMOD 데이터베이스 구성 변수의 값을 "ON"으로 설정하여 마지막 백업 이후에 변경된 데이터를 추적할 수 있도록 합니다.</p> <pre>db2> update db cfg for <DB 명> using TRACKMOD ON</pre> <p>3 INCREMENTAL, DELTA, ONLINE 등 다양한 옵션을 이용하여 최근의 FULL 백업 또는 INCREMENTAL 백업 이후에 변경된 부분에 대해서 백업 이미지를 생성합니다.</p> <pre>db2> backup db <DB 명> INCREMENTAL db2> backup db <DB 명> ONLINE INCREMENTAL DELTA INCLUDE LOGS</pre>

테이블 스페이스 백업 절차 비교

오라클	DB2
<p>1 sysdba 권한을 가지고 있는 사용자로 접속하여 NORMAL 모드로 SHUTDOWN 합니다.</p> <pre>SQL> shutdown normal;</pre> <p>2 OS Copy 명령어를 이용하여 데이터 파일, Control 파일, redo log 파일을 복사하거나 또는 RMAN 을 이용하여 백업합니다.</p> <pre>RMAN> BACKUP DATABASE;</pre>	<p>1 아카이브 로깅모드에서만 지원되므로 LOGARCHMETH1 값을 OFF 가 아닌 다른 값으로 설정합니다.</p> <pre>db2> update db cfg for <DB 명> using LOGARCHMETH1 DISK:<경로명></pre> <p>2 백업 명령어의 TABLESPACE 옵션을 이용하여 한 개 이상의 테이블 스페이스에 대한 백업 이미지를 생성하며 연관된 테이블 스페이스의 경우 함께 백업합니다.</p> <pre>db2> backup db <DB 명> tablespace (<테이블 스페이스 명, <테이블 스페이스 명>)</pre>

Point DB2는 백업 명령어를 이용하여 데이터베이스의 모든 데이터와 제어 정보를 저장한 이미지 파일을 생성합니다. 백업은 상황에 따라 백업 수준, 백업 범위, 백업 모드를 지정하여 백업 전략을 세웁니다.

백업 모드

모드	설명
오프라인	데이터베이스에 접속된 사용자가 없는 상태에서 백업하는 모드입니다. 백업이 진행되는 동안 데이터의 변경은 발생하지 않으므로, 백업 이미지 파일만 보관하면 복구에 사용할 수 있습니다.
온라인	사용자가 데이터베이스에 접속하고 있는 상태에서 백업하는 모드입니다. 백업이 진행되는 동안 데이터가 변경될 수도 있으므로 백업 이미지 파일과 백업이 진행되는 동안 변경된 데이터에 대한 로그 파일을 함께 보관해야 복구에 사용할 수 있습니다. 아카이브 로깅에서만 지원됩니다.

백업 대상

대상	설명
데이터베이스	데이터베이스의 모든 테이블 스페이스를 백업합니다.
테이블 스페이스	지정한 테이블 스페이스만 백업합니다. 아카이브 로깅에서만 지원됩니다.

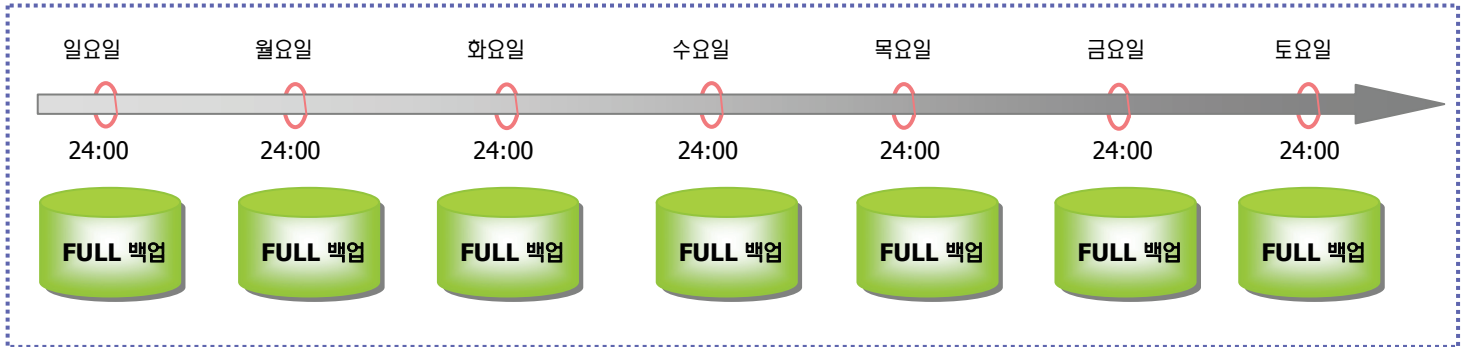
백업 범위

대상	설명
FULL	<ul style="list-style-type: none"> 데이터베이스의 모든 데이터와 제어 정보를 백업합니다. 주기적으로 FULL 백업을 수행합니다. 마지막으로 생성한 FULL 백업의 이미지는 복구에 사용되므로 안전하게 보관합니다. 온라인 백업일 경우 백업 작업 동안 사용된 로그 파일을 함께 보관합니다.
INCREMENTAL	<ul style="list-style-type: none"> 최근의 Full 백업 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다. 데이터베이스 CFG 파라미터인 LOGPARCHEMETH1 및 TRACKMOD 를 설정함으로써 지원 가능합니다. 백업 작업 동안 사용된 로그 파일은 복구를 대비하여 안전하게 보관해야 합니다.
DELTA	<ul style="list-style-type: none"> 최근의 FULL 또는 Incremental 백업 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다. 데이터베이스 CFG 파라미터인 LOGPARCHEMETH1 및 TRACKMOD 를 설정함으로써 지원 가능합니다. 백업 작업 동안 사용된 로그 파일은 복구를 대비하여 안전하게 보관해야 합니다.

백업 이미지 저장 장소

대상	설명
테이프	<ul style="list-style-type: none"> 백업 이미지를 테이프에 저장할 수 있습니다.
Named Pipe	<ul style="list-style-type: none"> 백업 이미지를 named pipe 에 저장하여 백업 및 복구할 수 있습니다.
디스크	<ul style="list-style-type: none"> 기본적인 모드로 백업 이미지를 디스크에 저장합니다.

Point DB2는 백업 명령어를 이용하여 데이터베이스의 모든 데이터와 제어 정보를 저장한 이미지 파일을 생성합니다. 다음은 데이터베이스의 모든 데이터 파일과 제어 정보를 저장하는 FULL 백업입니다.



FULL 백업 절차

- 1 주기적으로 FULL 백업을 수행합니다. 마지막으로 생성한 FULL 백업 이미지는 복구에 사용되므로 안전하게 보관합니다. 온라인 백업이라면 백업 작업 동안 사용된 로그 파일을 함께 보관해야 합니다.
- 2 순환 로깅 (Circular) 모드인 경우에는 오프라인 모드의 FULL 백업만 가능하며 아카이브 로깅 모드인 경우에는 오프라인 백업 및 온라인 백업이 모두 가능합니다. DB CFG 파라미터 중 LOGARCHMETH1를 체크하여 OFF 가 아닌 다른 값으로 설정되어 있다면 아카이브 로깅 모드입니다.

```
db2=> get db cfg for <DB 명> | find "LOGARCHMETH1"
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) = DISK:c:\warch_logs
```

- 3 백업 명령어를 이용하여 오프라인 백업 또는 온라인 백업을 수행합니다. 백업 명령어를 수행하는 세션은 내부적으로 데이터베이스에 접속하여 백업 이미지를 생성하고 백업이 완료되면 데이터베이스에 대한 접속을 종료합니다. 원격 데이터베이스의 백업을 수행할 때는 반드시 사용자 명과 암호명을 입력해야 합니다. TO 옵션을 이용하여 백업 이미지를 다른 디렉토리에 생성합니다.

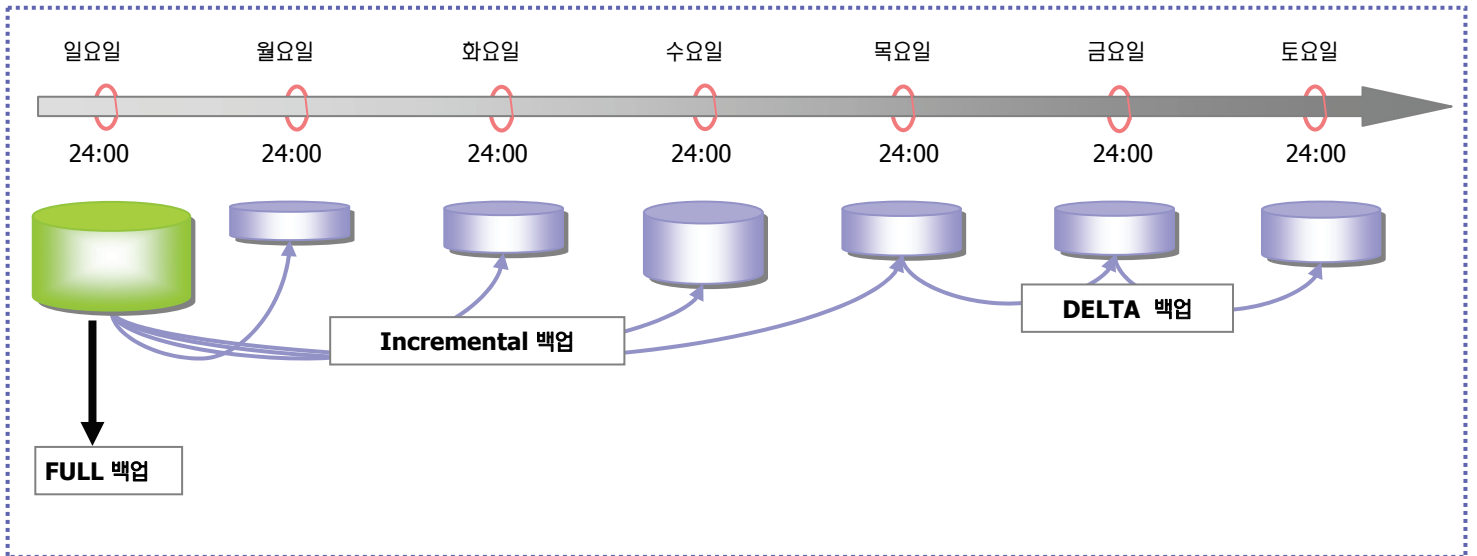
```
db2=> backup db <DB 명> USER <사용자 명> USING <암호명> TO <경로명>
```

- 4 list history 명령어를 이용하여 데이터베이스에 대한 백업 이력 사항 및 백업 이미지 정보를 확인합니다.

```
db2=> list history backup for <DB 명>
```

12 DB2 Incremental 및 DELTA 백업

Point DB2는 데이터베이스의 FULL 백업 또는 Incremental 백업 이후 변경된 데이터에 대한 정보만 저장하는 Incremental 백업을 제공합니다. Incremental 및 Delta 백업을 이용할 경우 백업 시간을 단축할 수 있습니다.



Incremental / Delta 백업 절차

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

2 아카이브 로깅 모드에서만 지원되므로 데이터베이스 구성 변수인 LOGARCHMETH1를 체크하여 OFF가 아닌 다른 값으로 설정합니다.

```
db2=> update db cfg for <DB 명> using LOGARCHMETH1 DISK:c:\warch_logs
db2=> get db cfg for <DB 명>
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) = DISK:c:\warch_logs
```

3 데이터베이스 구성 변수인 TRACKMOD를 ON으로 설정합니다.

```
db2=> update db cfg for <DB 명> using TRACKMOD ON
```

4 BACKUP 명령어에서 INCREMENTAL 옵션을 이용하여 최근의 FULL 백업 이후에 변경된 부분에 대해서만 백업 이미지를 생성합니다.

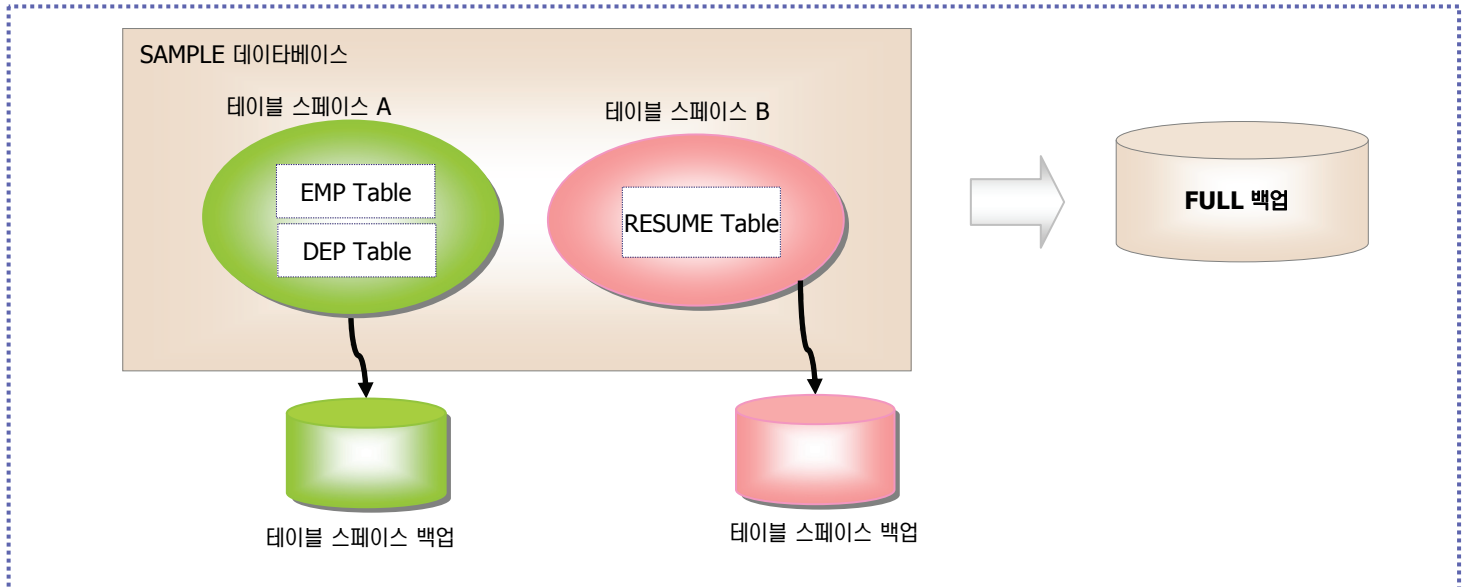
```
db2=> backup db <DB 명> incremental
db2=> backup db <DB 명> online incremental to <디렉토리명> INCLUDE LOGS
db2=> backup db <DB 명> online incremental delta
db2=> backup db <DB 명> online incremental delta to <디렉토리명> INCLUDE LOGS
```

5 list history 명령어를 이용하여 데이터베이스에 대한 백업 이력 사항 및 백업 이미지 정보를 확인합니다.

```
db2=> list history backup for <DB 명>
```

13 테이블 스페이스 백업

Point DB2는 테이블 스페이스 백업을 지원함으로써 백업에 소요되는 시간을 단축시킬 수 있습니다. 테이블 스페이스 백업은 한 개 이상의 테이블 스페이스를 함께 백업할 수 있으며 아카이브 로깅 모드에서 지원됩니다.



테이블 스페이스 백업 절차

1 카탈로그 테이블 스페이스 또는 중요한 테이블 스페이스만 별도로 백업합니다. 테이블 스페이스에 대한 백업 이미지 파일과 백업 작업 동안 사용된 로그 파일을 안전하게 보관해야 합니다.

2 아카이브 로깅 모드에서만 지원되므로 데이터베이스 구성 변수인 LOGARCHMETH1를 체크하여 OFF가 아닌 다른 값으로 설정합니다.

```
db2=> update db cfg for <DB 명> using LOGARCHMETH1 DISK:c:Warch_logs
db2=> get db cfg for <DB 명>
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) = DISK:c:Warch_logs
```

3 BACKUP DATABASE 명령에서 TABLESPACE 옵션을 함께 사용하여 한 개 이상의 테이블 스페이스에 대한 백업 이미지를 생성합니다.

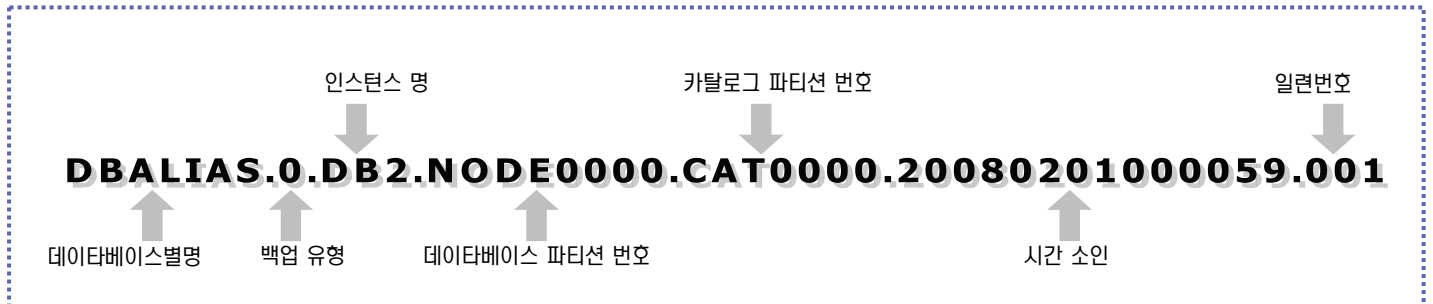
```
db2=> backup db <DB 명> tablespace (<테이블 스페이스 이름>,<테이블 스페이스 이름>)
db2=> backup db <DB 명> tablespace (<테이블 스페이스 이름>,<테이블 스페이스 이름>) online incremental
db2=> backup db <DB 명> tablespace (<테이블 스페이스 이름>,<테이블 스페이스 이름>) online incremental delta
```

4 list history 명령어를 이용하여 데이터베이스에 대한 백업 이력 사항 및 백업 이미지 정보를 확인합니다.

```
db2=> list history backup for <DB 명>
```

14 DB2 백업 이미지

Point 백업 이미지 파일 이름은 DB명, 백업 유형, 인스턴스 명, 데이터베이스 파티션 번호, 백업 완료 시간 소인, 파일 일련 번호 등의 정보를 가지고 DB2가 자동으로 생성합니다.



백업 이미지 이름 항목

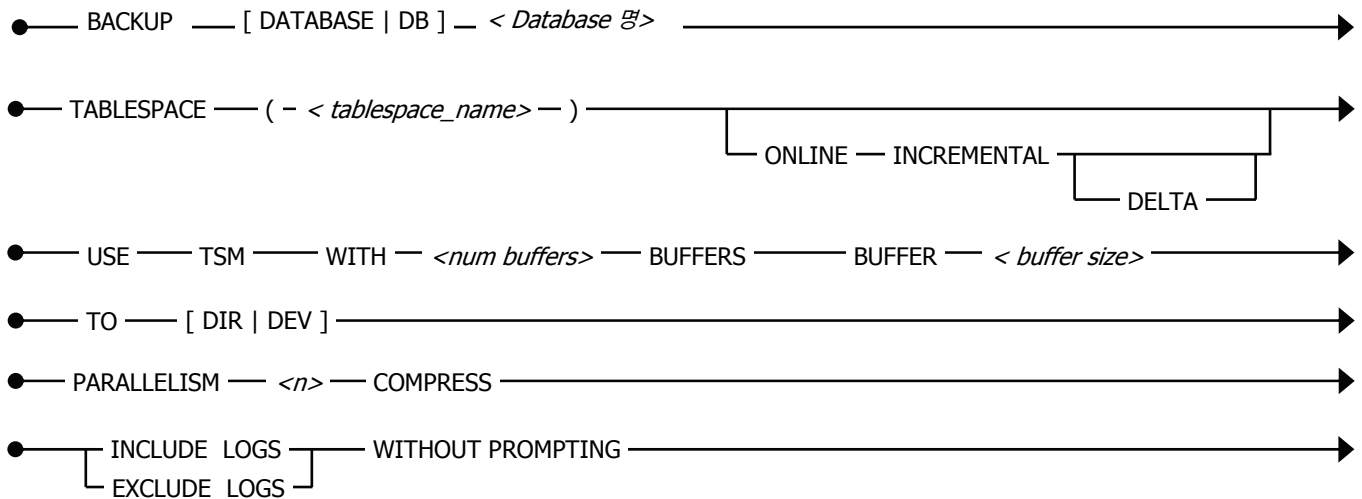
이름	설명
DB 명	지역 DB명 또는 CATALOG DB 명령어로 등록된 원격 데이터베이스 별명입니다.
백업 유형	0 은 데이터베이스 백업 이미지, 3은 테이블 스페이스 백업 이미지, 4는 LOAD 유틸리티의 COPY YES 옵션으로 생성된 백업 이미지를 나타냅니다.
인스턴스 명	데이터베이스가 존재하는 서버의 인스턴스 이름입니다. 원격 데이터베이스인 경우에는 원격 서버 머신에 백업 이미지 파일이 생성되고, 원격 인스턴스명이 표시됩니다.
데이터베이스 파티션 번호	백업된 데이터베이스 파티션의 번호입니다. DPF 환경에서는 파티션별로 백업 이미지가 생성되며, 단일 파티션인 경우에는 항상 NODE0000가 됩니다.
카탈로그 파티션 번호	카탈로그 테이블이 생성된 데이터베이스 파티션의 번호입니다. 단일 파티션인 경우에는 항상 CATN0000이 됩니다.
시간 소인	백업 작업이 완료된 시간 소인입니다. UNIX 에서는 <yyyymmddhhmmss>의 형태로 표시되고 윈도우일 경우에는 <yyyymmdd> 와 <hhmmss> 로 분리됩니다.
일련 번호	백업 이미지 파일이 두개 이상의 파일로 분할된 경우에는 동일한 이름을 가지며, 일련 번호가 다른 여러 개의 백업 이미지 파일이 생성됩니다.

* 부연설명

- 테이프 장치에 생성된 백업 이미지는 db2ckbcp 유틸리티를 이용하여 헤더 파일에 저장된 백업 정보를 확인하 수 있습니다.
- 유닉스에서는 복구시에 임의의 디렉토리에 이미지 파일을 복사하면 됩니다. 그러나 윈도우 경우에는 하위 경로명이 백업 이미지를 생성했을 때와 동일해야 합니다.

Point  다음은 DB2 Backup 명령문입니다.

Backup 구문



참고 : 이외에도 다양한 옵션이 지정되어 있으므로 도움말을 참고하시기 바랍니다.

옵션 설명 :


모드	설명
DB 명	지역 DB명 또는 Catalog DB 명령어로 등록된 원격 DB명을 지정합니다.
TABLESPACE	괄호안에 백업을 원하는 테이블스페이스의 이름들만 나열합니다.
ONLINE	온라인 모드의 백업을 실행합니다. 아카이브 로깅에서만 지원됩니다.
INCREMENTAL	INCREMENTAL 백업을 실행합니다. 아카이브 로깅에서만 지원됩니다.
DELTA	DELTA 백업을 실행합니다. INCREMENTAL 옵션과 함께 사용합니다. 아카이브 로깅에서만 지원됩니다.
TO	백업 이미지 파일이 생성되는 디렉토리를 지정할 수 있습니다. 한 개 이상의 경로명을 지정하면, 백업 이미지 파일은 여러 개의 파일로 분할되어 생성됩니다. 옵션을 지정하지 않으면, 현재 디렉토리에 한 개의 파일로 생성됩니다.
WITH ~ BUFFERS	백업 작업을 위해 데이터베이스 공유 메모리에서 할당하는 버퍼의 개수를 지정합니다.
BUFFERS	백업 작업을 위해 할당하는 버퍼의 크기를 지정합니다.
PARALLELISM	병렬적으로 읽을 수 있는 테이블 스페이스의 수를 지정합니다. 만약 명시적으로 지정되지 않을 경우 DB2가 최적의 값을 자동으로 선택합니다.
COMPRESS	백업 이미지가 압축되는 것을 지정합니다.
INCLUDE LOGS	roll forward 등 일지 시점까지 필요한 로그 파일을 백업 이미지에 함께 저장하는 옵션으로 오프라인 백업에는 적용되지 않습니다.

Point 데이터베이스에서 가장 중요한 기능 중 하나는 복구입니다. 다음은 오라클과 DB2의 복구에 대해 비교합니다.

	오라클	DB2
복구 방법 비교	<ul style="list-style-type: none"> instance 복구 <ul style="list-style-type: none"> 시스템 문제시 활성 로그 파일을 이용하여 오라클이 자동으로 복구합니다. ARCHIVEMODE 에서 불완전 (Incomplete) 복구 <ul style="list-style-type: none"> 오프라인 백업 및 로그 파일을 이용하여 데이터베이스를 특정 시점으로 복구하는 것을 불완전 복구 또는 부분 복구라 합니다. 완전 (Complete) 복구 <ul style="list-style-type: none"> NOARCHIVE Mode에서 완전 복구 <ul style="list-style-type: none"> 오프라인 백업을 가지고 복구하는 것으로 백업 받았던 시점으로 데이터베이스를 복구합니다. ARCHIVE MODE 에서 완전 복구 <ul style="list-style-type: none"> OFFLINE 백업 및 로그를 이용하여 데이터베이스 장애 발생시점 전으로 데이터베이스를 복구합니다. 	<ul style="list-style-type: none"> 크래쉬 복구 <ul style="list-style-type: none"> 시스템 문제시 활성 로그 파일을 이용하여 DB2 가 자동으로 복구합니다. 버전 복구 <ul style="list-style-type: none"> 과거 시점에 생성했던 오프라인 모드의 데이터베이스 백업 이미지를 이용하여 과거의 시점과 동일한 상태로 데이터베이스를 복구합니다. 롤포워드 복구 (roll forward) <ul style="list-style-type: none"> 버전 복구를 완료한 후에 아카이브 로그 파일을 이용하여 원하는 시점까지의 로그 파일을 적용하는 복구입니다.
복구 대상	<ul style="list-style-type: none"> 데이터베이스 전체 테이블 스페이스 데이터 파일 Control 파일, ARCHIVE LOG, SPFILE 등 	<ul style="list-style-type: none"> 데이터베이스 전체 테이블 스페이스
복구 범위	<ul style="list-style-type: none"> Full <ul style="list-style-type: none"> 데이터베이스의 모든 데이터와 제어 정보를 백업합니다. Cumulative Incremental Backup (Level 0 ~ Level 4) <ul style="list-style-type: none"> Level 에 따라 최근의 Full 백업 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다. Differential Incremental Backup (default) <ul style="list-style-type: none"> Level 에 따라 최근의 FULL 백업 또는 Incremental 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다. 	<ul style="list-style-type: none"> Full <ul style="list-style-type: none"> Full 백업 이미지를 이용하여 복구합니다. Incremental <ul style="list-style-type: none"> 최근의 Incremental 백업 이미지와 기준이 되는 Full 백업 이미지를 이용하여 복구합니다. Delta <ul style="list-style-type: none"> 최근의 DELTA 백업 이미지들과 기준이 되는 INCREMENTAL 백업 이미지, 기준이 되는 FULL 백업 이미지를 이용하여 복구합니다.
복구 권한	<ul style="list-style-type: none"> SYSDBA 권한이 필요합니다. 	<ul style="list-style-type: none"> sysadmin, sysctrl 또는 sysmaint 권한이 필요합니다.
복구 관리	<ul style="list-style-type: none"> RMAN 을 이용하여 기존 백업 내역을 관리할 수 있습니다. RMAN> 2> LIST BACKUP; SQL> SELECT * FROM V\$BACKUP_FILES; 	<ul style="list-style-type: none"> 복구 기록 파일을 이용하여 데이터베이스에 대한 백업, 복구, 로드 등의 작업에 대한 내역을 관리할 수 있습니다. db2=> list history rollforward all for <DB 명> db2=> SELECT * FROM table(sysproc.admin_list_hist()) as history
복구 방법	<ul style="list-style-type: none"> OS COPY 명령어를 이용한 이미지 복사 RMAN의 복구 명령어를 이용한 복구 	<ul style="list-style-type: none"> 복구 명령어를 이용한 복구



오라클 과 DB2의 백업 및 복구는 로깅 방법에 따라 달라집니다. 즉 로깅 방법이 Circular (DB2) 또는 NOARCHIVE 모드(오라클)일 경우 백업은 오프라인 백업만 가능하며 복구 또한 백업을 수행한 시점으로만 복구가 가능합니다. 그러나 ARCHIVE 모드일 경우에는 온라인 백업 및 Incremental 백업이 가능하며 복구 또한 장애 발생 시점 바로 전까지 가능합니다.

Point  다음은 오라클과 DB2의 복구 절차에 대해 비교합니다.

오프라인/온라인 데이터베이스 복구 절차 비교

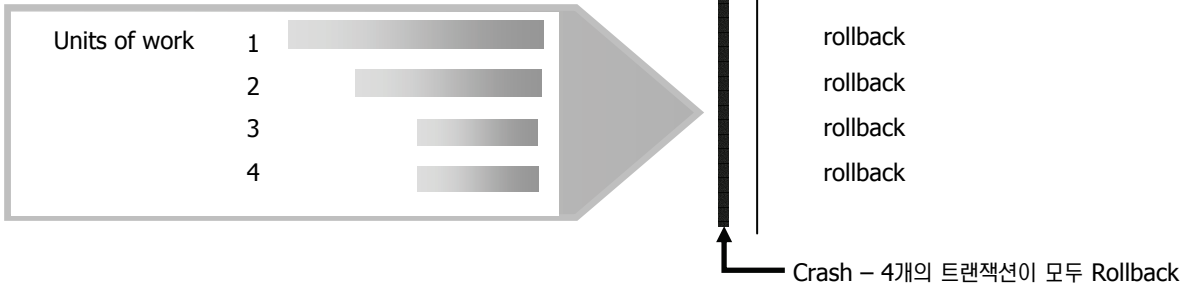
오라클	DB2
<p>1 sysdba 권한을 가지고 있는 사용자로 접속하여 NORMAL 모드로 SHUTDOWN 합니다.</p> <pre>SQL> shutdown normal;</pre>	<p>1 SYSADM 권한을 가진 사용자가 다음 명령어를 수행하여 데이터베이스에 접속한 애플리케이션을 모두 끊습니다. .</p> <pre>db2=> force applications all</pre>
<p>2 OS Copy 명령어를 이용하여 데이터 파일, Control 파일, redo log 파일을 복사하거나 또는 RMAN 의 RECOVER 명령어를 이용하여 복구합니다.</p> <pre>RMAN> RECOVER DATABASE;</pre>	<p>2 restore 명령어를 이용하여 복구를 수행합니다. 백업 이미지가 여러 개 있을 경우 TAKEN AT 옵션을 두어 선택할 수 있습니다. 또한 INTO 옵션을 사용하면 신규 데이터베이스로 복구할 수 있습니다.</p> <pre>db2=> restore db <DB 명> [online] TAKEN AT <시간></pre>
	<p>3 온라인 복구의 경우 수행하는 동안 연관된 로그 파일들도 함께 보관해야 복구를 완료할 수 있습니다.</p>

특정 시점에서의 복구 절차 비교

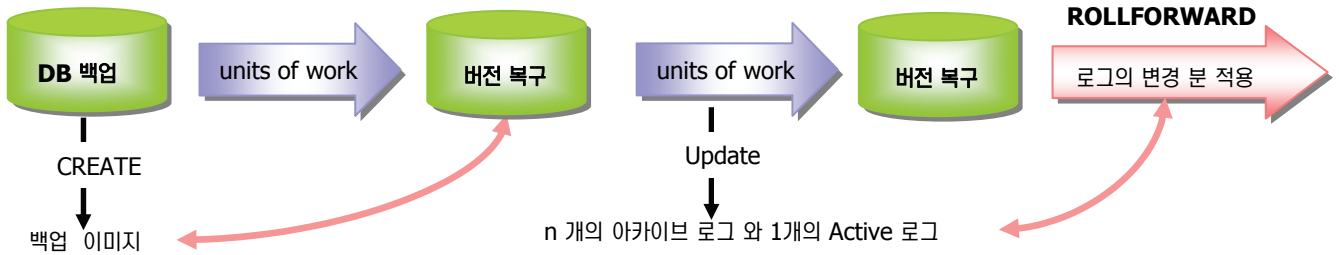
오라클	DB2
<p>1 백업 받은 데이터 파일을 원 위치로 복사합니다.</p> <pre>\$cp /BACKUP/* /u01/db01/ORADATA</pre>	<p>1 RESTORE 명령어를 이용하여 복구를 수행합니다.</p> <pre>db2=> restore db <DB 명> [online] TAKEN AT <시간></pre>
<p>2 데이터베이스를 MOUNT 합니다.</p> <pre>SQL> startup MOUNT</pre>	<p>2 Rollforward 명령어에서 ARCHIVE Log 파일을 이용하여 특정 시점으로 복구합니다.</p> <pre>db2=> rollforward db <DB 명> TO 2008-04-20-20.11.44 and complete</pre>
<p>3 데이터베이스를 복구합니다.</p> <pre>SQL> RECOVER DATABASE UNTIL TIME '2008-04-20:11:44:00';</pre>	
<p>4 RESETLOGS 를 이용하여 데이터베이스를 OPEN 합니다.</p> <pre>SQL> alter database open resetlogs;</pre>	

Point DB2는 백업 이미지와 아카이브 로그를 이용하여 원하는 시점으로 데이터베이스를 복구합니다.

Crash 복구 :



버전 복구 및 Rollforward 복구 :



복구 유형

모드	설명
크래쉬 복구	• 자연재해 등 비상시에 활성 로그 파일을 이용하여 DB2가 자동으로 실행하는 복구입니다.
버전 복구	• 과거 시점에 생성했던 오프라인 데이터베이스 백업 이미지를 이용하여 과거의 시점과 동일한 상태로 데이터베이스를 복구합니다.
롤포워드 복구	• 버전 복구를 완료한 후에 아카이브 로그 파일을 이용하여 원하는 시점까지의 로그 파일을 적용하는 복구입니다. • 아카이브 로깅에서만 지원됩니다.

복구 대상

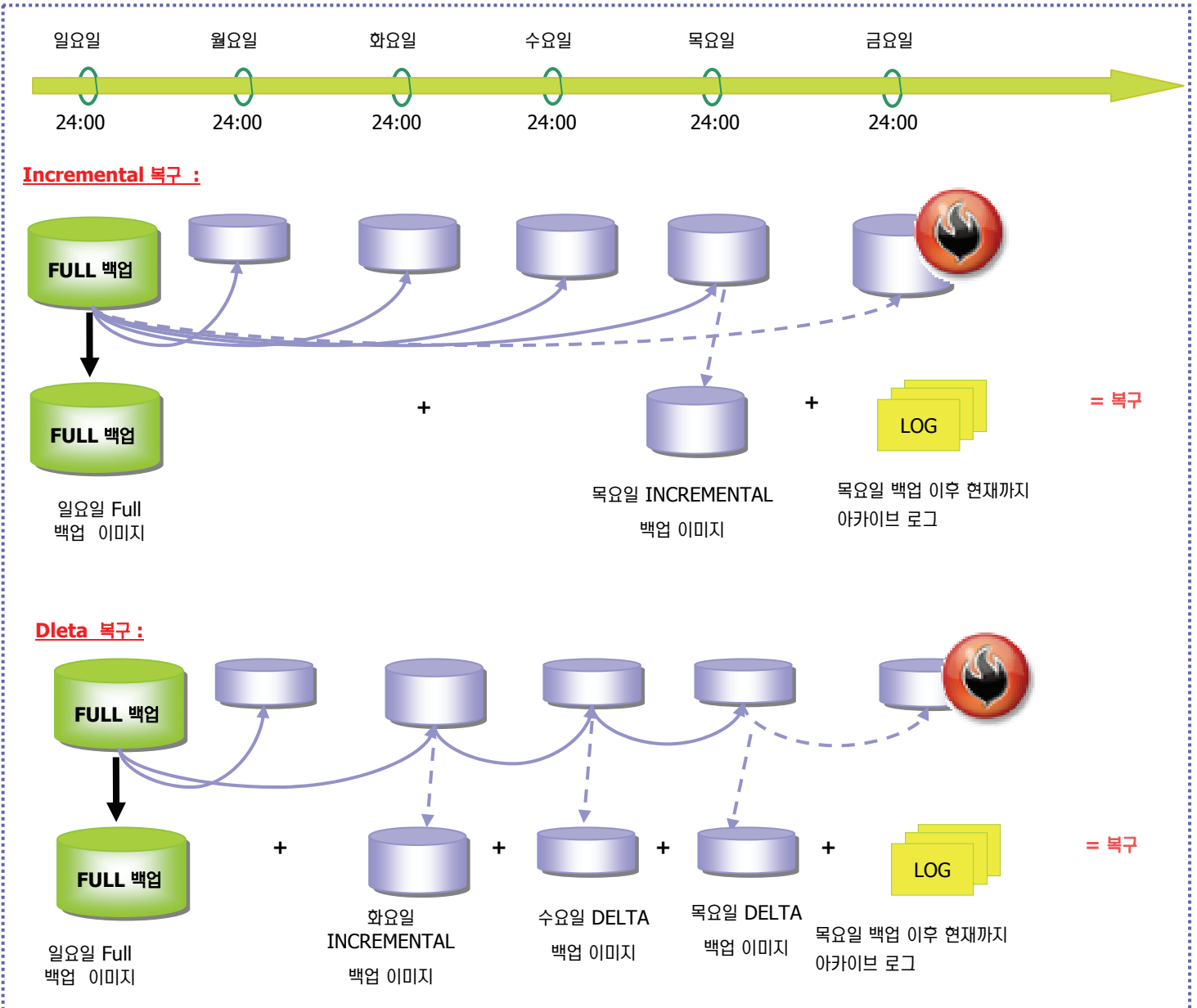
대상	설명
데이터베이스	• 데이터베이스의 모든 테이블 스페이스를 복구합니다.
테이블 스페이스	• 지정한 테이블 스페이스만 복구합니다. • 아카이브 로깅에서만 지원됩니다.

복구 범위

대상	설명
FULL	• FULL 백업 이미지를 이용하여 복구합니다.
INCREMENTAL	• 최근의 INCREMENTAL 백업 이미지와 기준이 되는 FULL 백업 이미지를 이용하여 복구합니다. • 아카이브 로깅에서만 지원됩니다.
DELTA	• 최근의 DELTA 백업 이미지들과 기준이 되는 INCREMENTAL 백업 이미지, 기준이 되는 FULL 백업 이미지를 이용하여 복구합니다. • 아카이브 로깅에서만 지원됩니다.


19 DB2 Incremental 및 Delta 복구

Point DB2는 Incremental 백업을 이용하여 백업 시간을 최소화합니다. 다음은 Incremental 및 Delta 백업 이미지를 이용한 복구입니다.



Incremental 및 Delta 복구

- 카탈로그 테이블 스페이스 또는 중요한 테이블 스페이스만 별도로 백업합니다. 테이블 스페이스에 대한 백업 이미지 파일과 백업 작업 동안 사용된 로그 파일을 안전하게 보관해야 합니다 .
- 아카이브 로깅 모드에서만 지원되므로 LOGARCHMETH1 을 OFF 가 아닌 값으로 설정합니다.
- BACKUP DATABASE 명령에서 TABLESPACE 옵션을 함께 사용하여 한 개 이상의 테이블 스페이스에 대한 백업 이미지를 생성합니다.

Point  RESTART 명령어는 DB2 장애시 크래쉬 복구를 실행하는 명령어입니다. 인스턴스가 비정상적으로 종료되는 경우에 크래쉬 복구가 필요합니다. 크래쉬 복구의 자동 실행 여부는 AUTORESTART 데이터베이스 구성 변수로 조절합니다.

RESTART 구문

- RESTART — [DATABASE | DB] — < Database 명 > →
- USER — < User Name > — USING — < Password > →
- DROP PENDING TABLESPACES — (< tablespace_name >) →
- WRITE RESUME →

참고 : 이외에도 다양한 옵션이 지정되어 있으므로 도움말을 참고하시기 바랍니다.


옵션 설명 :

모드	설명
DB 명	지역 DB명 또는 Catalog DB 명령어로 등록된 원격 DB명을 지정합니다.
USER, USING	크래쉬 복구를 실행하는 사용자 명과 암호명을 지정합니다. 원격 데이터베이스의 복구에는 반드시 USER 와 USING 옵션으로 사용자 명과 암호명을 지정해야 합니다.
DROP PENDING TABLESPACES	테이블 스페이스의 컨테이너가 손상되면 해당 테이블 스페이스는 사용할 수 없게 됩니다. 이러한 테이블 스페이스가 발견되면, 크래쉬 복구를 완료할 수 없습니다.

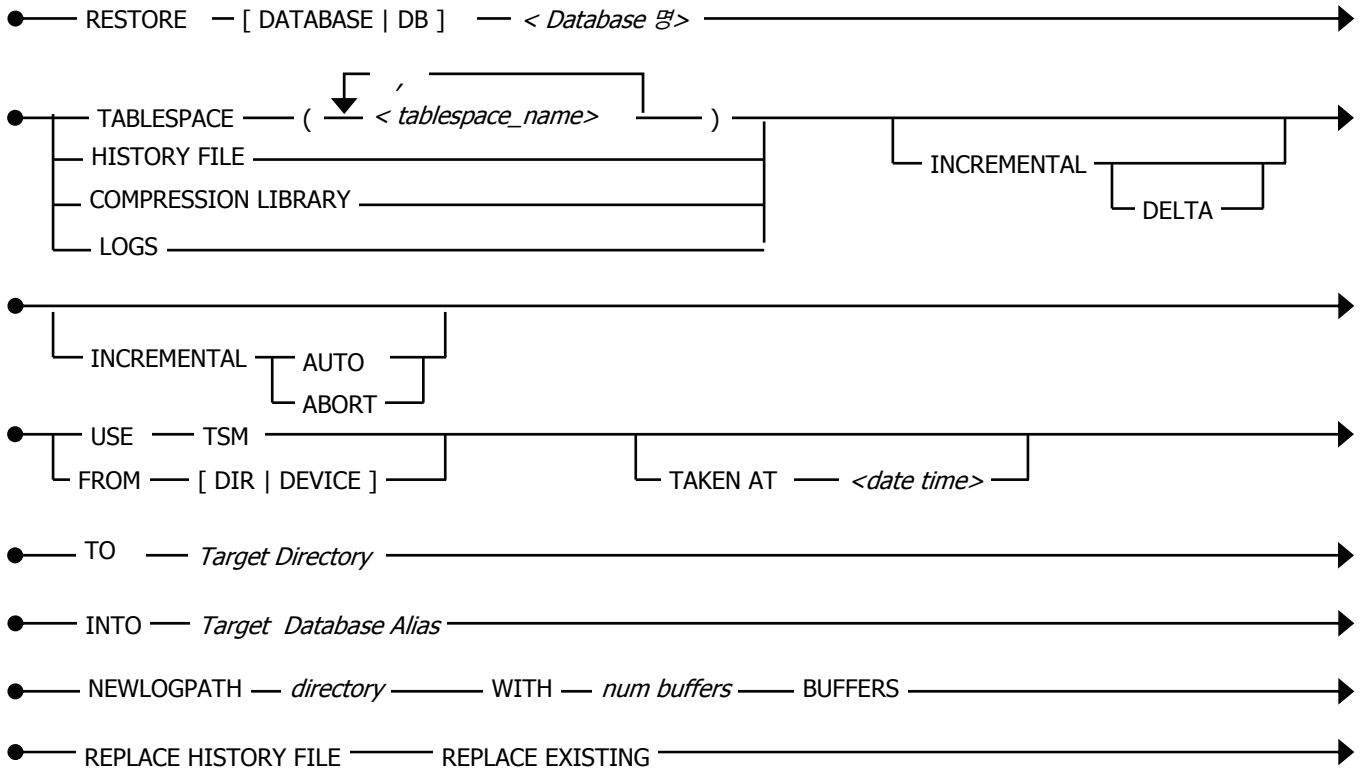


db2start 명령어를 이용하여 DB2 instance 를 시작하여도 크래쉬 복구가 수행됩니다.

21 DB2 RESTORE 명령어

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

RESTORE 구문



참고 : 이외에도 다양한 옵션이 지정되어 있으므로 도움말을 참고하시기 바랍니다.

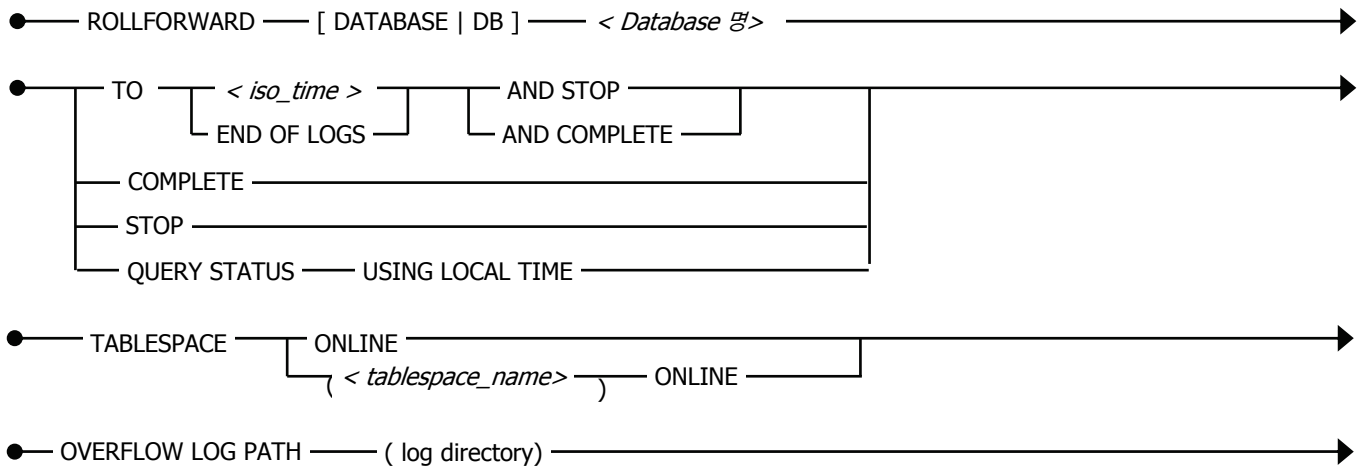
옵션 설명 :

모드	설명
DB 명	지역 DB명 또는 Catalog DB 명령어로 등록된 원격 DB명을 지정합니다.
TABLESPACE	괄호 안에 백업을 원하는 테이블 스페이스의 이름들만 나열합니다.
HISTORY FILE	복구 실행 기록 파일만 복구합니다.
ONLINE	온라인 모드로 복구합니다. TABLESPACE 복구에 지원됩니다.
INCREMENTAL	INCREMENTAL 또는 DELTA 백업 이미지의 복구를 실행합니다.
FROM	백업 이미지 파일이 존재하는 디렉토리를 지정합니다. 백업 이미지 파일이 여러 개의 파일로 분할되어 생성되었으면 한 개 이상의 경로명을 지정해야 합니다. 옵션을 지정하지 않으면, 현재 디렉토리에 한 개의 백업 이미지 파일을 이용합니다.
TAKEN AT	지정한 경로명에 동일한 DB에 대한 백업 이미지가 여러 개 있는 경우에는 백업 이미지의 생성 시간 소인을 명시하여 구분합니다.
TO	새로운 데이터베이스가 생성될 경로명을 지정합니다.
INTO	새로 생성될 데이터베이스명을 지정합니다.
WITH ~ BUFFERS	복구 작업을 위해 데이터베이스의 공유 메모리에서 할당하는 버퍼의 개수를 지정합니다.
BUFFER	복구 작업을 위해 할당하는 버퍼의 크기를 지정합니다.

22 DB2 ROLLFORWARD 명령어

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

ROLLFORWARD 구문



참고 : 이외에도 다양한 옵션이 지정되어 있으므로 도움말을 참고하시기 바랍니다.

옵션 설명 :

모드	설명
DB 명	지역 DB명 또는 Catalog DB 명령어로 등록된 원격 DB명을 지정합니다.
TO isotime	<표준 시간> 으로 표시한 시점까지의 로그 파일을 적용합니다. OS의 TIMEZONE 값을 이용한 환산이 필요합니다. <표준 시간>은 <yyyy-mm-dd-hh.mm.ss.nnnnnn>의 형태로 표현합니다.
USING LOCAL TIME	To isotime 옵션과 함께 사용됩니다. <표준시간> 대신에 현 시스템의 시간대에서 사용하는 <지역 시간>을 지정합니다. <지역 시간> 도 <yyyy-mm-dd-hh.mm.ss.nnnnnn>의 형태로 표현합니다.
TO END OF LOGS	로그 디렉토리에 존재하는 온라인 아카이브 로그 파일의 최대 번호까지 롤포워드를 실행합니다.
STOP	롤포워드를 종료하고, '롤포워드 보류' 상태를 해제합니다.
CANCEL	롤포워드를 중단하고, 데이터베이스를 다시 '리스토어 보류' 상태가 되게 합니다.
QUERY STATUS	롤포워드 작업의 진행 정보를 알려줍니다.
TABLESPACE	테이블 스페이스 수준의 롤포워드 작업시에 사용합니다.
ONLINE	테이블 스페이스 수준의 롤포워드를 온라인 모드로 실행합니다. 다른 테이블 스페이스는 액세스할 수 있습니다.

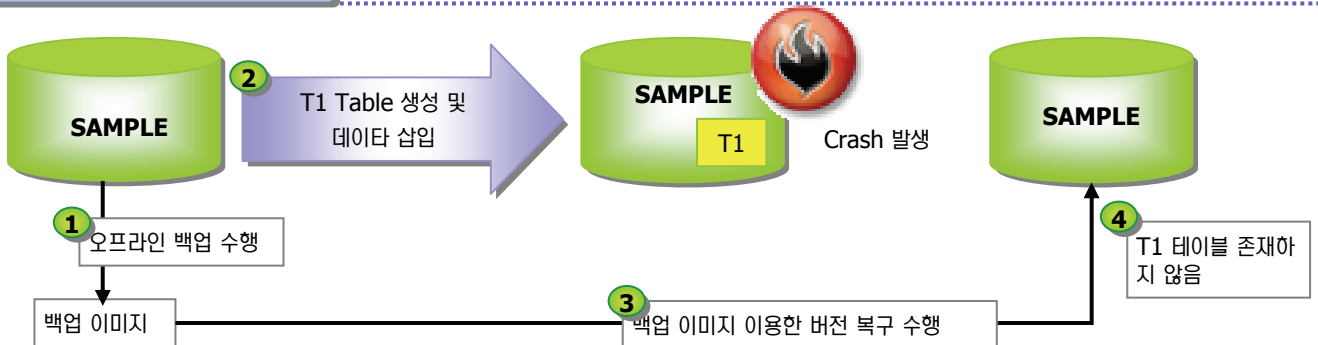
23 백업 및 복구 시나리오

Point 다음은 다음장에서 실습할 백업 및 복구에 대한 시나리오입니다. 총 5가지 시나리오를 가지고 백업 및 복구를 실습합니다.

전제 조건

- db2sampl 명령어를 이용하여 SAMPLE 데이터베이스를 생성합니다.
- 앞에 로깅 모드를 변경하는 Lab 을 수행하여 아카이브 로깅 모드와 아카이브 로그 디렉토리를 설정합니다.

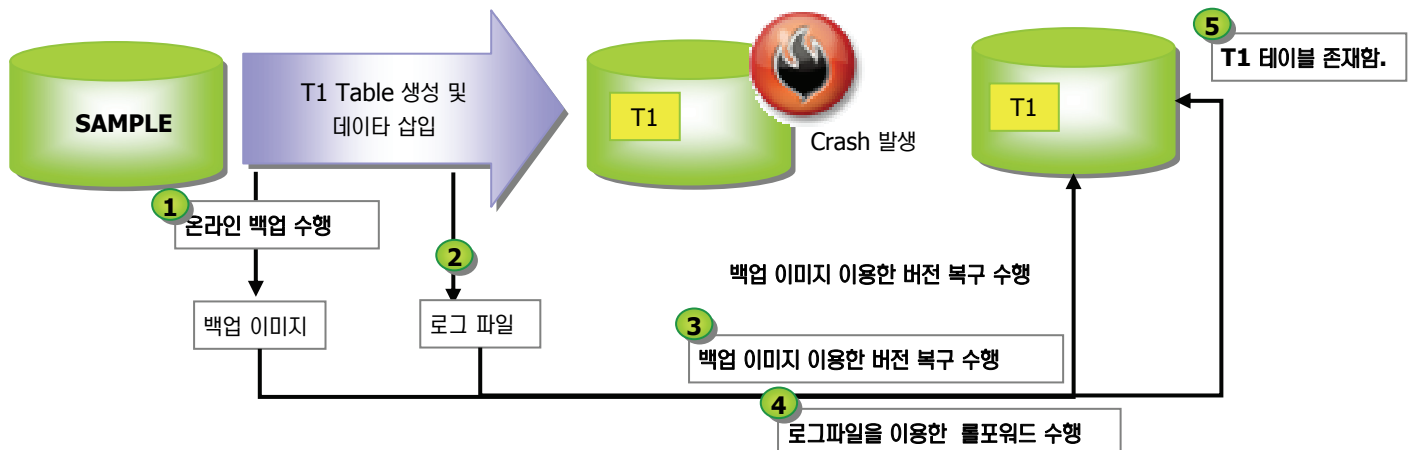
시나리오 1: 오프라인 백업 및 버전 복구



절차:

- 1 SAMPLE 데이터베이스의 오프라인 백업을 수행합니다.
- 2 T1 테이블을 생성하고 데이터를 삽입합니다.
- 3 1번에서 수행한 오프라인 백업 이미지를 이용하여 복구를 수행 합니다.
- 4 백업 받은 시점으로 데이터베이스를 복구합니다. T1 테이블은 존재하지 않게 됩니다.

시나리오 2: 온라인 백업 및 롤포워드 복구



절차:

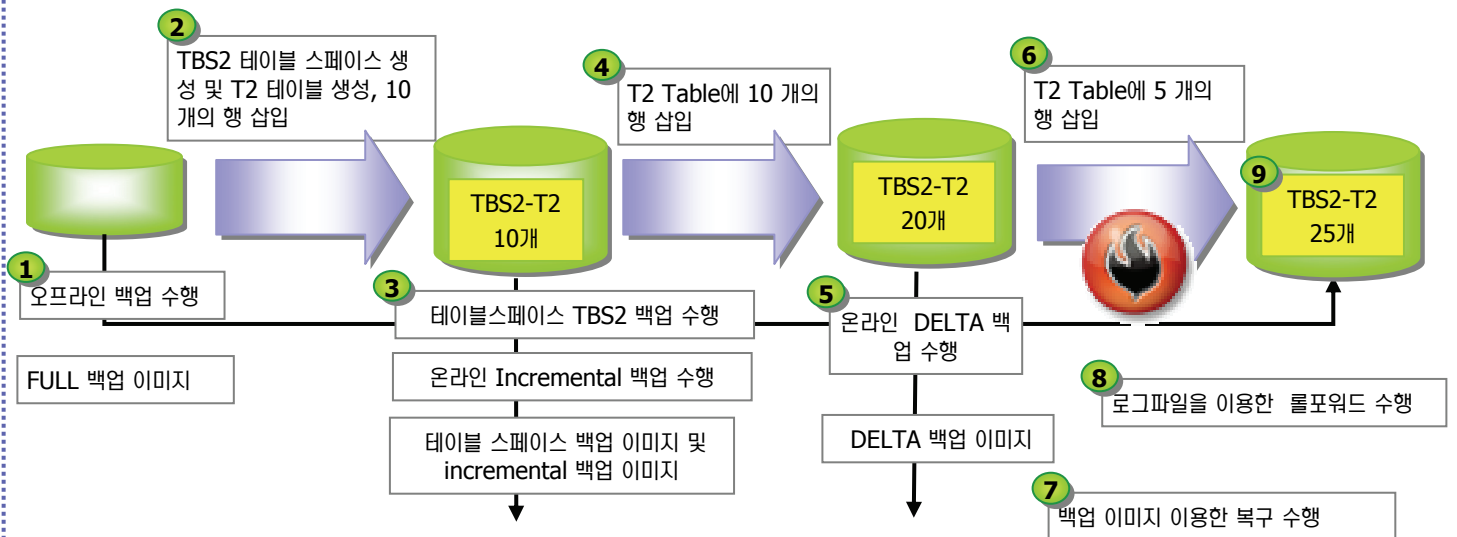
- 1 SAMPLE 데이터베이스 온라인 백업 수행과 동시에 다른 클라이언트에서 T1 테이블 생성 및 데이터를 삽입합니다.
- 2 아카이브 로그 파일을 저장합니다.
- 3 온라인 백업 이미지를 이용하여 SAMPLE 데이터베이스를 복구합니다.
- 4 아카이브 로그 파일을 이용하여 장애 발생시점 전으로 롤포워드 복구합니다.
- 5 데이터베이스 장애 발생시점 전까지 데이터베이스를 복구합니다. T1 테이블이 존재합니다.

Point 다음은 다음장에서 실습할 백업 및 복구에 대한 시나리오입니다. 총 5가지 시나리오를 가지고 백업 및 복구를 실습합니다.

전제 조건

- db2sampl 명령어를 이용하여 SAMPLE 데이터베이스를 생성합니다.
- 앞에 로깅 모드를 변경하는 Lab 을 수행하여 아카이브 로깅 모드와 아카이브 로그 디렉토리를 설정합니다.

시나리오 3: Incremental 및 Delta 백업 및 복구



절차 :

- 1 SAMPLE 데이터베이스의 오프라인 백업을 수행합니다.
- 2 TBS2 테이블 스페이스를 생성하고 TBS2 테이블 스페이스에 T2 테이블을 생성하고 10개의 데이터를 삽입합니다.
- 3 온라인 Incremental 백업을 수행합니다. TBS2 에 대한 백업 이미지가 없으므로 incremental 백업이 실패합니다. TBS2 백업을 수행한후에 온라인 incremental 백업을 수행합니다.
- 4 T2 테이블에 10개의 데이터를 더 삽입합니다.
- 5 온라인 DELTA 백업을 수행합니다.
- 6 T2 테이블에 5개의 행을 삽입합니다.
- 7 백업 이미지를 이용하여 데이터베이스를 복구합니다. 이때 자동 복구와 수동 복구 모드가 있습니다. 자동 복구의 경우 RESTORE 시 "AUTOMATIC" 옵션을 두어 복구하고 수동 복구시 "db2ckrst" 명령어를 이용하여 수동으로 RESTORE 시 수행해야할 순서 및 명령어를 보여줍니다. 이를 바탕으로 수동 복구를 수행합니다.
- 8 ROLLFORWARD 명령어를 이용하여 로그를 적용합니다.
- 9 SAMPLE 데이터베이스에 접속하여 T2 테이블에 25개의 데이터가 입력되었는지 확인합니다.

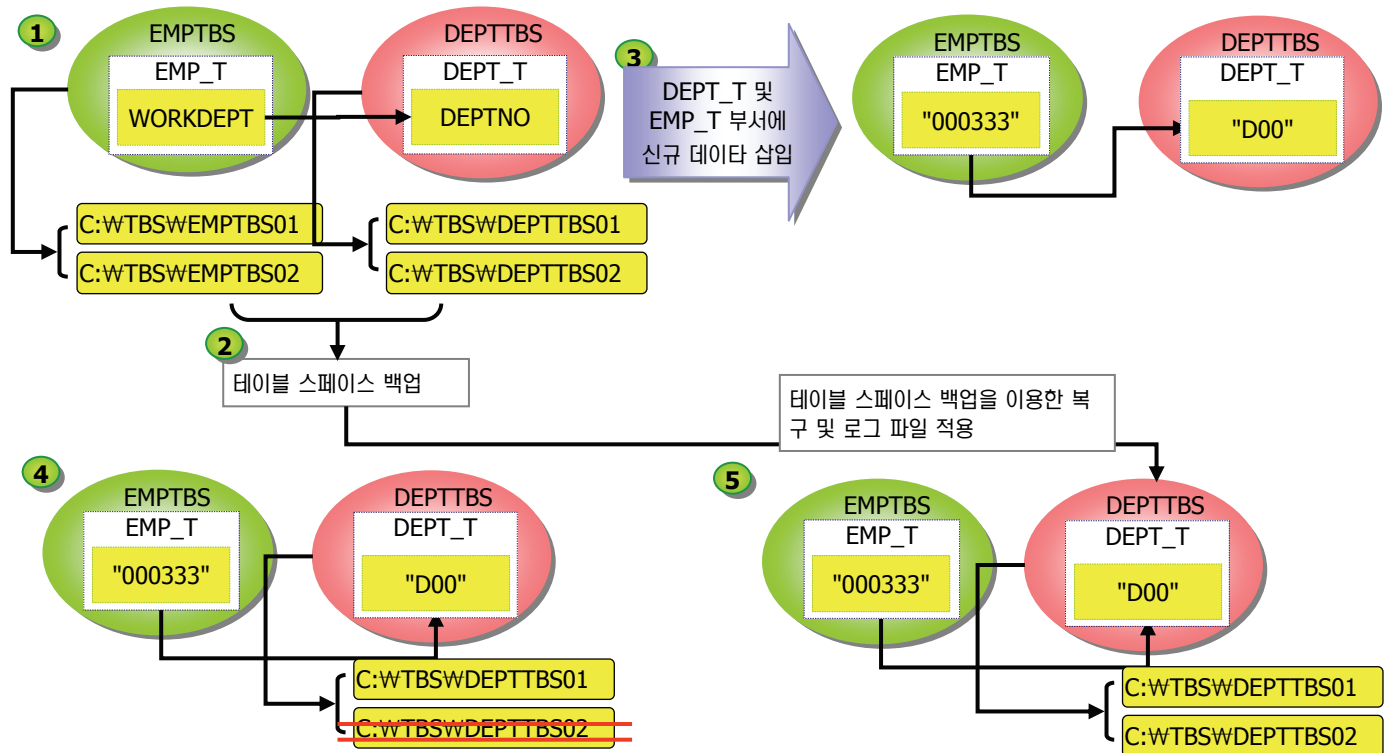
25 백업 및 복구 시나리오

Point 다음은 다음장에서 실습할 백업 및 복구에 대한 시나리오입니다. 총 5가지 시나리오를 가지고 백업 및 복구를 실습합니다.

전제 조건

- db2sampl 명령어를 이용하여 SAMPLE 데이터베이스를 생성합니다.
- 앞에 로깅 모드를 변경하는 Lab 을 수행하여 아카이브 로깅 모드와 아카이브 로그 디렉토리를 설정합니다.

시나리오 4: 테이블 스페이스 백업 및 복구



절차:

- 1** EMBTBS 와 DEPTTBS 테이블 스페이스를 생성하고 EMP_T, DEPT_T 테이블을 생성합니다. 또한 DEPT_T 테이블의 DEPTNO 를 Primary Key 로 지정하고 EMP_T 테이블의 WORKDEPT 를 Foreign Key 로 지정합니다.
- 2** EMBTBS 와 DEPTTBS 테이블 스페이스를 백업합니다.
- 3** EMP_T 테이블과 DEPT_T 테이블에 데이터를 삽입합니다.
- 4** DEPTTBS의 컨테이너 중 한 파일을 삭제합니다.
- 5** 테이블 스페이스 백업 이미지와 로그 파일을 이용하여 복구합니다.
- 6** 신규 데이터인 "000333" 및 "D00" 데이터를 모두 조회할 수 있습니다.

참고 사항:

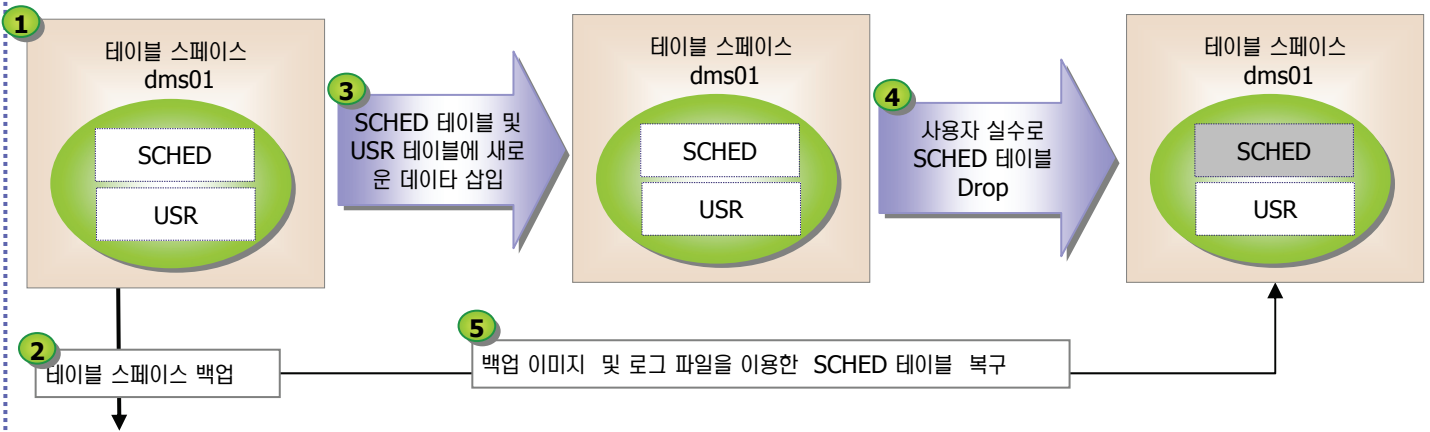
DEPTTBS 의 테이블 스페이스 백업 이미지를 이용한 후 로그 파일을 적용하지 않은채 복구할 경우 EMP_T 테이블과 DEPT_T 테이블은 Consistency 가 이루어지지 않습니다. 즉 EMP_T 테이블의 "000333" 직원이 속한 "D00" 부서가 DEPT_T 에는 존재하지 않기 때문에 EMP_T 테이블은 Set Integrity 상태가 됩니다. 이럴 경우 SET INTEGRITY 명령어를 이용하여 Consistency 가 이루어지지 않은 레코드를 EMP_T 에서 제거하여 Consistency 상태를 맞추도록 합니다.

Point 다음은 다음장에서 실습할 백업 및 복구에 대한 시나리오입니다. 총 5가지 시나리오를 가지고 백업 및 복구를 실습합니다.

전제 조건

- db2sampl 명령어를 이용하여 SAMPLE 데이터베이스를 생성합니다.
- 앞에 로깅 모드를 변경하는 Lab 을 수행하여 아카이브 로깅 모드와 아카이브 로그 디렉토리를 설정합니다.


시나리오 5: 삭제된 테이블 복구



절차:

- 1 dms01 테이블 스페이스를 생성한 후 USR, SCHED 를 dms01 테이블 스페이스에 생성하고 데이터를 입력합니다. 테이블 스페이스를 백업하기 전에 데이터베이스를 조회하여 기록해 놓습니다.
- 2 테이블 스페이스를 백업합니다.
- 3 USR Table 과 SCHED Table에 새로운 데이터를 삽입합니다.
- 4 SCHED 테이블을 Drop 합니다.
- 5 테이블이 Drop 되기 전에 백업받았던 데이터베이스 또는 Table space level backup image 를 restore 한 후 rollforward 명령어를 수행하여 삭제된 테이블 뿐만 아니라 가장 최근의 로그까지 적용하도록 합니다. 롤포워드 수행시 지정된 디렉토리에 data 파일이 생성됩니다. 이를 이용하여 SCHED 테이블을 복구합니다.
- 6 SCHED 테이블이 Drop 되기 전까지의 데이터를 조회할 수 있습니다.

27 시나리오 1: 오프라인 백업 및 버전 복구

Point  명령어를 이용하여 DB2 백업 및 복구를 수행합니다. 여기서는 로깅 모드가 아카이브 모드라고 가정하고 오프라인 백업 및 버전 복구를 수행합니다.

1. DB CFG의 파라미터를 확인하여 DB가 현재 Circular 인지 Archive 모드인지 확인합니다.

```
c:\W> db2 get db cfg for sample | find "LOGARCHMETH1"
```

```
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) =
DISK:c:\Warch_logs
```

2. 현재 SAMPLE DB에 접속하여 접속한 애플리케이션의 연결을 모두 끊고 현재 연결도 종료합니다.

```
c:\W> db2 force applications all
c:\W>db2 terminate
c:\W>db2 deactivate db sample
```

3. 두개의 CMD 창을 띄워 한쪽 창에서는 백업 명령어를 이용하여 오프라인 백업을 수행하고 한쪽 창에서는 "list utilities" 명령어를 수행하여 백업 진행 상황을 체크합니다.

CMD 1

```
c:\W>db2 backup db sample
```

```
백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은 20080417
012842입니다.
```

CMD 2

```
C:\Wbackup_test>db2 list utilities show detail
```

```
ID                = 82
유형              = BACKUP
데이터베이스 이름 = SAMPLE
파티션 번호      = 0
설명             = offline db
시작 시간        = 2008-04-17 01:28:49.552295
상태             = 실행 중
호출 유형       = 사용자
조절 기능:
우선순위        = 조절되지 않음
진행 모니터링:
  예상된 백분율 완료 = 92
  전체 작업         = 77289216 bytes
  완료 작업        = 71211240 bytes
  시작 시간        = 2008-04-17 01:28:49.552314
```

4. db2ckbckp 명령어를 이용하여 백업 이미지 정보 확인 및 이미지 파일에 손상이 없는지 체크합니다.

```
C:\Wbackup>db2ckbckp -a
SAMPLE.0.DB2.NODE0000.CATN0000.200804170128
42.001 | more
```

```
=====
MEDIA HEADER REACHED:
=====
Server Database Name      -- SAMPLE
Server Database Alias    -- SAMPLE
Client Database Alias    -- SAMPLE
Timestamp                 -- 20080417012842
Database Partition Number -- 0
Instance                  -- DB2
Sequence Number          -- 1
Release ID                -- C00
Database Seed             -- 48471AC6
DB Comment's Codepage (Volume) -- 0
DB Comment (Volume)      --
DB Comment's Codepage (System) -- 0
DB Comment (System)      --
Authentication Value     -- 255
Backup Mode              -- 0
Includes Logs            -- 0
Compression              -- 0
Backup Type              -- 0
Backup Gran.            -- 0
Status Flags             -- 21
System Cats inc          -- 1
Catalog Partition Number -- 0
DB Codeset               -- UTF-8
DB Territory             --
LogID                    -- 1208353537
LogPath                  -- c:\Wsample_logs\WNODE0000W
Backup Buffer Size       -- 6033408
Number of Sessions      -- 1
Platform                 -- 5
-- more --
```



여기서는 로깅 모드가 아카이브 모드라고 가정하여 오프라인 백업 및 버전 복구를 수행합니다.

5. 다음과 같이 T1 Table 을 생성하고 데이터를 입력합니다.

```
c:\W>db2 create table t1 ( c1 int)
c:\W>db2 insert into t1 values (1),(2),(3),(4),(5),(6);
```

6. SAMPLE 데이터베이스에 접속한 모든 애플리케이션 접속을 끊고 SAMPLE 데이터베이스를 비활성화 합니다.

```
c:\W> db2 force applications all
c:\W>db2 terminate
c:\W>db2 deactivate db sample
```

7. 로깅 모드가 아카이브일 경우 버전 복구 및 rollforward 복구 모두 가능합니다. 버전 복구, 즉 백업을 받은 시점으로 데이터베이스를 만들기 위해서는 복구시 현재 활성 로그를 적용하지 않도록 **without rolling forward** 옵션을 주어 복구 하도록 합니다. 이 옵션을 생략할 경우 데이터베이스는 "rollforward pending" 상태가 되므로 버전 복구시에는 이 옵션을 사용토록 합니다. 또한 백업 이미지가 여러 개일 경우 **TAKEN AT** 옵션을 두어 어느 시점의 백업 이미지로 복구할 것인지 지정합니다.

```
c:\W>db2 restore db sample taken at 20080417012842
without rolling forward
```

```
SQL2539W 경고! 백업 이미지 데이터베이스와 동일한 기존 데이터베이스로 리스토어
중입니다. 데이터베이스 파일이 삭제됩니다.
계속하시겠습니까? (y/n) y
DB20000I RESTORE DATABASE 명령이 완료되었습니다.
```

8. 백업을 받은 후에 생성한 T1 테이블이 존재하는지 체크합니다.

```
c:\W>db2 connect to sample
c:\W>db2 "SELECT * T1"
```

```
SQL0204N "KR010389.T1"은(는) 정의되지 않은 이름입니다.
```

29 시나리오 2: 온라인 백업 및 롤포워드 복구

백업 및 복구 비교

Point 온라인 백업 및 롤포워드 복구를 수행하기 위해서는 로깅 모드가 아카이브 모드이어야 합니다.

1. DB CFG 의 파라미터를 확인하여 DB가 현재 Circular 인지 Archive 모 드인지 확인한 후 로그 파일 크기와 로그 버퍼의 크기를 늘리도록 합니다. 그리고 SAMPLE 데이터베이스를 Deactivate 한 후 다시 ACTIVATE 하 도록 합니다.

```
c:\W> db2 get db cfg for sample | find "LOGARCHMETH1"
```

```
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) =
DISK:c:\Warch_logs
```

```
c:\W> db2 update db cfg for sample using LOGBUFSZ 100
```

```
DB20000I UPDATE DATABASE CONFIGURATION 명령이 완
료되었습니다.
```

```
c:\W> db2 update db cfg for sample using LOGFILSIZ 10000
```

```
DB20000I UPDATE DATABASE CONFIGURATION 명령이 완
료되었습니다.
```

```
c:\W> db2 force applications all
```

```
c:\W>db2 terminate
```

```
c:\W>db2 deactivate db sample
```

2. SAMPLE DB에 다시 접속하여 T1 테이블을 생성하고 데이터를 삽입합니 다.

```
c:\W> db2 drop table t1
```

```
c:\W> db2 create table t1 ( c1 int)
```

```
c:\W>db2 insert into t1 values (1),(2),(3),(4),(5),(6),(7)
```

3. CMD 창을 두개 더 띄운 후에 CMD 1에서는 t1 테이블에 데이터 삽입 및 T1 테이블의 데이터 개수를 조회하는 SELECT 문을 수행하고 CMD 2에 서는 온라인 백업을 수행하고 CMD3 창에서는 백업 상황을 모니터링할 수 있도록 "list utilities" 명령어를 동시에 수행합니다. CMD1에서 수행하는 ins.db2에서는 insert 문을 20개정도 수행하도록 하며 5번째마다 Commit을 수행하도록 합니다.

CMD 1

```
c:\W>notepad ins.db2
insert into t1 select * from t1;
insert into t1 select * from t1;
.....
commit;
insert into t1 select * from t1;
insert into t1 select * from t1;
c:\W>db2 -tvf ins.db2
```

CMD 2

```
c:\W>db2 backup db sample online
```

```
백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은
20080419131216 입니다.
```

CMD 3

```
C:\Wbackup_test>db2 list utilities show detail
```

```
ID = 5
유형 = BACKUP
데이터베이스 이름 = SAMPLE
파티션 번호 = 0
설명 = online db
시작 시간 = 2008-04-19 13:12:20.907648
상태 = 실행 중
호출 유형 = 사용자
조정 기능:
우선순위 = 조절되지 않음
진행 모니터링:
예상된 백분율 완료 = 90
전체 작업 = 81919632 bytes
완료 작업 = 73845880 bytes
시작 시간 = 2008-04-19 13:12:20.928511
```

4. "list history" 명령어를 이용하여 복구 기록 파일을 확인합니다.

```
C:\Wbackup>db2 list history backup since 200804191312 for
sample
```

sample에 대한 실행기록 파일 목록

```
일치하는 파일 항목의 수 = 1
```

```
Op Obj 시간소인+순서 유형 Dev 최초 로그 현재 로그 백
업 ID
```

```
-----
B D 20080419131216001 N D S0000001.LOG
S0000005.LOG
-----
```

```
3 테이블 스페이스를 포함함:
```

```
00001 SYSCATSPACE
00002 USERSPACE1
00003 IBMDB2SAMPLEREL
```

```
-----
주석: DB2 BACKUP SAMPLE ONLINE
```

```
시작 시간: 20080419131216
```

```
종료 시간: 20080419131239
```

```
상태: A
```

```
-----
EID: 20 위치: C:\Wbackup
```

30 시나리오 2: 온라인 백업 및 롤포워드 복구

백업 및 복구 비교

Point 명령어를 이용하여 DB2 백업 및 복구를 수행합니다. 온라인 백업 및 롤포워드 복구를 수행하기 위해서는 로깅 모드가 아카이브 모드이어야 합니다.

5. c:\wtest_logs 디렉토리를 생성합니다.

```
C:\Wbackup>mkdir c:\wtest_logs
```

6. 앞에서 수행한 SAMPLE 온라인 백업이미지를 이용하여 TEST 라는 새로운 데이터베이스로 복구합니다. 이때 위에서 생성한 디렉토리를 로그 파일의 디렉토리로 지정합니다.

```
C:\Wbackup>db2 terminate
C:\Wbackup>db2 restore db sample taken at 200804191312
into test newlogpath c:\wtest_logs
```

DB20000I RESTORE DATABASE 명령이 완료되었습니다.

7. TEST 데이터베이스에 연결합니다. 로그 파일이 적용되지 않았으므로 rollforward pending 상태가 됩니다.

```
C:\Wbackup>db2 connect to test
```

SQL1117N ROLL-FORWARD PENDING 때문에 데이터베이스 "TEST"에 연결하거나 활성화할 수 없습니다. SQLSTATE=57019

8. TEST 데이터베이스의 CFG 파라미터중 아카이브 디렉토리를 확인합니다.

```
C:\Wbackup>db2 get db cfg for test | find "LOG"
```

```
카탈로그 캐시 크기(4KB)(CATALOGCACHE_SZ) =
(MAXAPPLS*4)
로그 버퍼 크기(4KB)(LOGBUFSZ) = 8
로그 파일 크기(4KB)(LOGFILSIZ) = 1000
1차 로그 파일 수(LOGPRIMARY) = 3
2차 로그 파일 수(LOGSECOND) = 2
로그 파일에 대한 변경된 경로(NEWLOGPATH) =
오버플로우 로그 경로(OVERFLOWLOGPATH) =
미러 로그 경로(MIRRORLOGPATH) =
처음에 사용되는 로그 파일 = S0000001.LOG
디스크가 가득 차면 로그 블록화(BLK_LOG_DSK_FUL) = NO
트랜잭션에서 사용 중인 최대 1차 로그 스페이스 백분율(MAX_LOG)
= 0
1개의 활성 UOW에 대해 사용 중인 로그 파일 수
(NUM_LOG_SPAN) = 0
복구를 위해 로그 유지 작동(LOGRETAIN) = OFF
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) =
DISK:c:\Warch_logs\w
```

9. SAMPLE 데이터베이스의 로그 아카이브 디렉토리와 로그 디렉토리에서 아카이브 로그 파일과 활성 로그 파일을 TEST 데이터베이스의 아카이브 디렉토리와 Restore 시 NEWLOGPATH 로 지정된 디렉토리에 각각 복사합니다.

```
C:\Wbackup>copy c:\wdb2_logs\w* c:\wtest_logs
C:\Wbackup>copy c:\Warch_logs\WDB2\WSAMPLE\w*
c:\Warch_logs\WDB2\WTEST\w
```

10. rollforward 명령어를 수행하여 로그 파일을 적용합니다.

```
C:\Wbackup>db2 rollforward db test to end of logs and
complete
```

롤 포워드 상태

입력 데이터베이스 별명	= test
상태를 리턴한 노드 수	= 1
노드 번호	= 0
롤 포워드 상태	= 보류하지 않음
다음에 읽을 로그 파일	=
처리된 로그 파일	= S0000001.LOG - S0000022.LOG
최종 커미트된 트랜잭션	= 2008-04-19-23.40.54.000000 UTC

DB20000I ROLLFORWARD 명령이 완료되었습니다.

11. TEST 데이터베이스에 접속하여 T1 테이블의 총 개수와 SAMPLE 데이터베이스에 있는 T1 테이블의 총 개수가 동일한지 확인합니다.

```
C:\Wbackup>db2 connect to test
c:\Wbackup>db2 "selet count(*) from t1"
```

```
1_____
3670016
1 레코드가 선택되었습니다.
```

```
C:\Wbackup>db2 connect to sample
c:\Wbackup>db2 "selet count(*) from t1"
```

```
1_____
3670016
1 레코드가 선택되었습니다.
```


31 시나리오 3: Incremental 백업 및 복구

백업 및 복구 비교

Point 명령어를 이용하여 DB2 백업 및 복구를 수행합니다. incremental 백업 및 복구를 수행하기 위해서는 로깅 모드가 아카이브 모드이어야 합니다.

1. DB CFG 의 파라미터를 확인하여 DB가 현재 Circular 인지 Archive 모 드인지 확인합니다.

```
c:\W> db2 get db cfg for sample | find "LOGARCHMETH1"
```

```
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) =
DISK:c:\Warch_logs
```

2. DB 구성 파라미터인 TRACKMOD 를 "ON" 으로 설정합니다.

```
c:\W> db2 update db cfg for sample using TRACKMOD ON
```

3. 접속한 Application 의 연결을 모두 끊고 백업을 수행합니다.

```
c:\W>db2 force applications all
c:\W>db2 terminate
c:\W>db2 deactivate db sample
c:\W>db2 backup db sample
```

```
백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은
20080422084437 입니다.
```

4. SAMPLE 데이터베이스에 접속한 후 테이블 스페이스 TBS2 를 생성하고 T2 테이블을 TBS2에 생성합니다. 그리고 T2 테이블에 10개의 데이터를 삽입합니다.

```
c:\W>db2 connect to sample
c:\W>db2 create tablespace tbs2 managed by database
using (file 'c:\wtbs\wtbs021' 10M, file 'c:\wtbs\wtbs022'
10M)
c:\W>db2 create table t2 ( c1 int) in tbs2
c:\W>db2 insert into t2 values
(1),(2),(3),(4),(5),(6),(7),(8),(9),(10)
```

5. SAMPLE 데이터베이스를 online incremental 합니다.

```
c:\Wbackup>db2 backup db sample online incremental
```

```
SQL2426N 데이터베이스가 증분 백업 조작을 허용하도록 구성되어
있지 않습니다.
이유 코드 = "2".
```

6. SQL2426N 의 에러를 확인하여 조치사항에 대해 알아봅니다.

```
c:\Wbackup>db2 ? SQL2426N
```

설명: 데이터베이스에 대해 수정 추적기 활성화되고 테이블 스페이스에 비증분 백업이 수행된 후에야 테이블 스페이스에 대해 증분 백업이 사용 가능합니다.

가능한 이유 코드는 다음과 같습니다.

1. 구성 매개변수 TRACKMOD가 데이터베이스에 대해 설정되어 있지 않습니다.
2. TRACKMOD 구성 매개변수가 설정되었지만, TRACKMOD 매개변수가 설정된 이후에 적어도 하나의 테이블 스페이스에 비증분 백업이 수행되지 않았습니다.

사용자 응답:

조치는 다음과 같이 이유 코드에 따라 다릅니다.

1. TRACKMOD 데이터베이스 매개변수를 ON으로 설정하여 데이터베이스의 수정 추적을 활성화한 다음, 데이터베이스의 전체 백업을 수행하십시오.
2. db2diag.log 파일을 참조하여 테이블 스페이스 이름을 판별한 후 해당 테이블 스페이스의 전체 백업을 수행하십시오.

7. Reason 코드가 2이므로 db2diag.log 파일을 참조하도록 합니다. db2diag 명령어를 이용하여 out 이라는 파일에 db2diag.log 파일의 내용을 기록하도록 합니다. 일반 Editor 프로그램을 이용하여 out 이라는 파일을 열어 확인합니다. online 백업을 수행한 시간을 찾아 원인과 조치사항을 찾아봅니다.

```
c:\Wbackup>db2diag -output out
```

```
c:\Wbackup>notepad out
```

```
2008-04-22-01.44.23.268000+540 E3007375H576
LEVEL: Warning
PID : 3584 TID : 6176 PROC :
db2syscs.exe
INSTANCE: DB2 NODE : 000 DB :
SAMPLE
APPHDL : 0-586 APPID:
*LOCAL.DB2.080421164421
AUTHID : KR010389
EDUID : 6176 EDUNAME: db2agent (SAMPLE)
0
FUNCTION: DB2 UDB, database utilities,
sqlubBuildAppTbsp, probe:1562
MESSAGE : ADM8005W Incremental backup was not
enabled for table space "TBS2"
(ID "6"). A non-incremental backup of this table
space is required.
```

32 시나리오 3: Incremental 백업 및 복구

백업 및 복구 비교

Point 명령어를 이용하여 DB2 백업 및 복구를 수행합니다. incremental 백업 및 복구를 수행하기 위해서는 로깅 모드가 아카이브 모드이어야 합니다.

8. 모든 응용 프로그램의 접속을 끊고 SAMPLE 데이터베이스를 DEACTIVATE 한 후에 TBS2에 대해 테이블 스페이스 백업을 수행합니다.

```
c:\Wbackup> db2 force applications all
c:\Wbackup> db2 terminate
c:\Wbackup> db2 backup db sample tablespace(tbs2)
```

백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은 20080422084659 입니다.

9. ONLINE Incremental 백업을 수행합니다.

```
C:\Wbackup>db2 backup db sample online incremental
```

백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은 20080422084736 입니다.

10. t2 테이블에 10개의 데이터를 더 삽입하고 총 개수를 확인합니다.

```
c:\Wbackup> db2 connect to sample
c:\Wbackup> db2 insert into t2 values
(11),(12),(13),(14),(15),(16),(17),(18),(19),(20)
c:\W>db2 select count(*) from t2
```

```
1
-----
20
```

11. ONLINE Incremental delta 백업을 수행합니다.

```
C:\Wbackup>db2 backup db sample online incremental delta
```

백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은 20080422084854 입니다.

12. t2 테이블에 5개의 데이터를 더 입력합니다.

```
C:\Wbackup>db2 insert into t2 values
(21),(22),(23),(24),(25);
c:\Wbackup> db2 select count(*) from t2
```

```
1
-----
25
```

13. Restore 명령어에 "AUTOMATIC" 옵션을 두어 가장 최근 시점의 백업 이미지를 취하여 자동 복구를 수행함과 동시에 CLP 창을 하나 더 띄우고 "list utilities" 명령어를 이용하여 복구 현황을 체크합니다.

CMD 1

```
C:\Wbackup>db2 force applications all
C:\Wbackup>db2 terminate
C:\Wbackup>db2 restore db sample incremental automatic
taken at 20080422084854
```

SQL2539W 경고! 백업 이미지 데이터베이스와 동일한 기존 데이터베이스로 리스토어 중입니다. 데이터베이스 파일이 삭제됩니다. 계속하시겠습니까? (y/n) y


CMD 2

```
C:\Wbackup_test>db2 list utilities show detail
```

```
ID = 41
유형 = RESTORE
데이터베이스 이름 = SAMPLE
파티션 번호 = 0
설명 = automatic incremental db
시작 시간 = 2008-04-22 08:52:08.150942
상태 = 실행 중
호출 유형 = 사용자
진행 모니터링:
  단계 번호 = 1
  전체 작업 = 7376896 bytes
  완료 작업 = 7376896 bytes
  시작 시간 = 2008-04-22 08:52:08.150951
  단계 번호 = 2
  설명 = 20080422084437
  전체 작업 = 98967552 bytes
  완료 작업 = 98967552 bytes
  시작 시간 = 2008-04-22 08:52:11.313640
  단계 번호 = 3
  설명 = 20080422084659
  전체 작업 = 18378752 bytes
  완료 작업 = 18378752 bytes
  시작 시간 = 2008-04-22 08:52:16.067129
  단계 번호 [현재] = 4
  설명 = 20080422084736
  완료 작업 = 22073344 bytes
  시작 시간 = 2008-04-22 08:52:16.356168
  단계 번호 = 5
```


33 시나리오 3: Incremental 백업 및 복구

백업 및 복구 비교

Point  명령어를 이용하여 DB2 백업 및 복구를 수행합니다. incremental 백업 및 복구를 수행하기 위해서는 로깅 모드가 아카이브 모드이어야 합니다.

14. SAMPLE 데이터베이스에 접속합니다. 로그 파일을 적용하지 않았으므로 ROLLFORWARD PENDING 상태가 됩니다.

```
C:\Wbackup>db2 connect to sample
```

```
SQL1117N  ROLL-FORWARD PENDING 때문에 데이터베이스 "SAMPLE"에 연결하거나 활성화할 수 없습니다.
SQLSTATE=57019
```

15. 롤포워드 복구를 로그 끝까지 수행합니다.

```
C:\Wbackup>db2 rollforward db sample to end of logs and complete
```

롤 포워드 상태

입력 데이터베이스 별명 = sample

상태를 리턴한 노드 수 = 1

노드 번호 = 0

롤 포워드 상태 = 보류하지 않음

다음에 읽을 로그 파일 =

처리된 로그 파일 = S0000047.LOG - S0000048.LOG

최종 커밋된 트랜잭션 = 2008-04-21-23.49.56.000000 UTC

16. SAMPLE 데이터베이스에 접속한 후 t2 테이블의 데이터 개수를 확인합니다.

```
c:\Wbackup> db2 connect to sample
```

```
c:\Wbackup>db2 select count(*) from t2
```

```
1
-----
25
```

17. db2ckrst 명령어를 이용하여 수동으로 incremental 복구시 수행해야 할 명령어 및 순서를 알려줍니다. t 옵션을 이용하여 복구할 마지막 백업 이미지를 지정합니다. 이때 시간소인은 초단위까지 정확히 입력해야 합니다.

```
C:\Wbackup>db2ckrst -d sample -t 20080422084854
```

Suggested restore order of images using timestamp 20080422084854 for database sample.

```
=====
restore db sample incremental taken at 20080422084854
restore db sample incremental taken at 20080422084437
restore db sample incremental taken at 20080422084659
restore db sample incremental taken at 20080422084736
restore db sample incremental taken at 20080422084854
=====
```

18. 위의 restore 명령어를 Editor 프로그램을 열어 복사한 후 저장합니다.

```
C:\Wbackup>notepad restore.db2
```

```
restore db sample incremental taken at 20080422084854;
restore db sample incremental taken at 20080422084437;
restore db sample incremental taken at 20080422084659;
restore db sample incremental taken at 20080422084736;
restore db sample incremental taken at 20080422084854;
```

34 시나리오 3: Incremental 백업 및 복구

백업 및 복구 비교

Point 명령어를 이용하여 DB2 백업 및 복구를 수행합니다. incremental 백업 및 복구를 수행하기 위해서는 로깅 모드가 아카이브 모드이어야 합니다.

19. 앞에서 저장한 restore.db2 스크립트를 수행합니다.

```
C:\Wbackup>db2 -tvf restore.db2
```

```
restore db sample incremental taken at 20080422084854
SQL2539W 경고! 백업 이미지 데이터베이스와 동일한 기존 데이터베이스로 리스토어 중입니다. 데이터베이스 파일이 삭제됩니다.
계속하시겠습니까? (y/n) y
DB20000I RESTORE DATABASE 명령이 완료되었습니다.
```

```
restore db sample incremental taken at 20080422084437
DB20000I RESTORE DATABASE 명령이 완료되었습니다.
```

```
restore db sample incremental taken at 20080422084659
DB20000I RESTORE DATABASE 명령이 완료되었습니다.
```

```
restore db sample incremental taken at 20080422084736
DB20000I RESTORE DATABASE 명령이 완료되었습니다.
```

```
restore db sample incremental taken at 20080422084854
DB20000I RESTORE DATABASE 명령이 완료되었습니다.
```

20. SAMPLE 데이터베이스에 접속합니다. 로그 파일을 적용하지 않았으므로 ROLLFORWARD PENDING 상태가 됩니다.

```
C:\Wbackup>db2 connect to sample
```

```
SQL1117N ROLL-FORWARD PENDING 때문에 데이터베이스 "SAMPLE"에 연결하거나 활성화할 수 없습니다.
SQLSTATE=57019
```

21. 롤포워드 복구를 로그 끝까지 수행합니다. 앞에서 자동 복구로 수행했을 때 롤포워드 결과 화면과 동일한지 확인합니다.

```
C:\Wbackup>db2 rollforward db sample to end of logs and complete
```

```
롤 포워드 상태

입력 데이터베이스 별명      = sample
상태를 리턴한 노드 수      = 1

노드 번호                    = 0
롤 포워드 상태              = 보류하지 않음
다음에 읽을 로그 파일      =
처리된 로그 파일            = S0000047.LOG - S0000048.LOG
최종 커밋된 트랜잭션        = 2008-04-21-23.49.56.000000 UTC
```

22. SAMPLE 데이터베이스에 접속한 후 t2 테이블의 데이터 개수를 확인합니다.


```
c:\W> db2 connect to sample
```

```
c:\W>db2 select count(*) from t2
```

```
1
-----
25
```

35 시나리오 4 : 테이블 스페이스 백업 및 복구

백업 및 복구 비교

Point  명령어를 이용하여 DB2 백업 및 복구를 수행합니다. 테이블 스페이스 백업 및 rollforward 복구를 수행하기 위해서는 로깅 모드가 아카이브 모드이어야 합니다.

1. DB CFG의 파라미터를 확인하여 DB가 현재 Circular 인지 Archive 모드인지 확인합니다.

```
c:\W> db2 get db cfg for sample | find "LOGARCHMETH1"
```

```
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) =
DISK:c:\Warch_logs
```

2. SAMPLE DB에 접속하여 다음과 같이 EMPTBS, DEPTBS 테이블 스페이스와 EMP_T 테이블 및 DEPT_T 테이블을 생성합니다. 또한 EMP_T 테이블과 DEPT_T 테이블에 Referential 관계를 생성한 후에 데이터를 입력합니다.

```
C:\WLABWUNIT11> notepad s4_crtbs.db2
```

```
CONNECT TO SAMPLE;
DROP TABLESPACE EMPTBS;
DROP TABLESPACE DEPTBS;
```

```
CREATE TABLESPACE EMPTBS MANAGED BY DATABASE
USING (FILE 'C:\WTBSWEMPTBS01' 10M,
FILE 'C:\WTBSWEMPTBS02' 10M);
CREATE TABLESPACE DEPTBS MANAGED BY DATABASE
USING (FILE 'C:\WTBSWDEPTBS01' 10M,
FILE 'C:\WTBSWDEPTBS02' 10M);
CREATE TABLE EMP_T LIKE EMPLOYEE IN EMPTBS;
CREATE TABLE DEPT_T LIKE DEPARTMENT IN DEPTBS;
ALTER TABLE DEPT_T ADD PRIMARY KEY ( DEPTNO);
ALTER TABLE EMP_T ADD CONSTRAINT F_WORKDEPT
FOREIGN KEY(WORKDEPT)REFERENCES DEPT_T;
INSERT INTO DEPT_T SELECT * FROM DEPARTMENT;
INSERT INTO EMP_T SELECT * FROM EMPLOYEE;
```

```
C:\WLABWUNIT11> db2 -tvf s4_crtbs.db2
```

```
INSERT INTO DEPT_T SELECT * FROM DEPARTMENT
DB20000I SQL 명령이 완료되었습니다.
```

```
INSERT INTO EMP_T SELECT * FROM EMPLOYEE
DB20000I SQL 명령이 완료되었습니다.
```

3. EMPTBS와 DEPTBS 테이블 스페이스를 각각 백업한 후에 시간 소인을 확인합니다.

```
C:\WLABWUNIT11> notepad s4_backup.db2
```

```
BACKUP DB SAMPLE TABLESPACE(EMPTBS) ONLINE;
BACKUP DB SAMPLE TABLESPACE(DEPTBS) ONLINE;
```

```
C:\WLABWUNIT11> db2 -tvf s4_backup.db2
```

```
BACKUP DB SAMPLE TABLESPACE(EMPTBS)
백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은
20080428172338입니다.
```

```
BACKUP DB SAMPLE TABLESPACE(DEPTBS)
백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은
20080428172342입니다.
```

4. DEPT_T 테이블에 'D00' 데이터를 입력하고 EMP_T 테이블에 'D00' 부서에 속하는 직원 정보를 입력하고 DEPT_T와 EMP_T 테이블의 데이터 개수를 확인합니다.

```
C:\WLABWUNIT11> notepad s4_ins.db2
```

```
CONNECT TO SAMPLE;
INSERT INTO DEPT_T VALUES
('D00','TECHLINE','000010','A00','SEOUL ');
INSERT INTO EMP_T VALUES
('010333','Myungjoo','Ahn','D00','6756','2001-02-
16','FIELDREP',16,'F','1984-11-08',31840.00,500.00,
1907.00);
SELECT COUNT(*) FROM DEPT_T;
SELECT COUNT(*) FROM EMP_T;
```

```
C:\WLABWUNIT11> db2 -tvf s4_ins.db2
```

5. DEPTBS 테이블 스페이스의 컨테이너 중 한 파일을 다른 이름으로 변경합니다. 만약 아래와 같이 rename 할 때 오류가 발생할 경우 SAMPLE 데이터베이스에 연결된 모든 세션을 종료한 후에 다시 한번 수행합니다.

```
C:\WLABWUNIT11> rename c:\WTBSWDEPTBS01
TEMPBS01
```


```
C:\WLABWUNIT11> rename c:\WTBSWDEPTBS01
TEMPBS01 다른 프로세스가 파일을 사용 중이기 때문에 프로세스가
액세스 할 수 없습니다.
```

```
C:\WLABWUNIT11> db2 force applications all
```

```
C:\WLABWUNIT11> rename c:\WTBSWDEPTBS01 TEMPBS01
```

36 시나리오 4 : 테이블 스페이스 백업 및 복구

백업 및 복구 비교

Point  명령어를 이용하여 DB2 백업 및 복구를 수행합니다. 테이블 스페이스 백업 및 rollforward 복구를 수행하기 위해서는 로깅 모드가 아카이브 모드이어야 합니다.

6. SAMPLE 에 다시 접속하여 EMP_T 와 DEPT_T 테이블을 조회합니다.

```
C:\WLABWUNIT11> db2 connect to sample
C:\WLABWUNIT11> db2 select count(*) from emp_t
```

```
1
-----
      43
1 레코드가 선택되었습니다.
```

```
C:\WLABWUNIT11> "select count(*) from dept_t
```

```
1
-----
SQL0290N 테이블 스페이스 액세스가 허용되지 않습니다.
SQLSTATE=55039
```

7. "db2cmd" 명령어를 이용하여 또하나의 DB2 CLP 창을 띄운후 SAMPLE 데이터베이스에 접속합니다. DEPTTBS 테이블 스페이스 백업 이미지를 이용하여 복구를 수행합니다. 다른 백업 이미지가 존재하므로 앞에서 수행한 DEPTTBS의 백업 수행시간을 TAKEN AT 옵션을 두어 지정합니다.

```
C:\WLABWUNIT11> db2 restore db sample
tablespace(depttbs) online taken at
20080428172342
```

```
DB20000I RESTORE DATABASE 명령이 완료되었습니다.
```

8. 원래의 창에서 다시 EMP_T 테이블과 DEPT_T 테이블을 조회합니다.

```
C:\WLABWUNIT11> db2 connect to sample
C:\WLABWUNIT11> db2 select count(*) from emp_t
```

```
1
-----
      43
1 레코드가 선택되었습니다.
```

```
C:\WLABWUNIT11> db2 "select count(*) from dept_t
```

```
1
-----
SQL0290N 테이블 스페이스 액세스가 허용되지 않습니다.
SQLSTATE=55039
```

9. Restore를 수행했던 창에서 다시 rollforward를 수행합니다. 이때 rollforward는 로그를 적용하지 않은채 완료합니다.

```
C:\WLABWUNIT11> db2 rollforward db sample to end of
logs tablespace(depttbs) online
```

```
롤 포워드 상태

입력 데이터베이스 별명 = sample
상태를 리턴한 노드 수 = 1
노드 번호 = 0
롤 포워드 상태 = 보류하지 않음
다음에 읽을 로그 파일 =
처리된 로그 파일 = -
최종 커밋된 트랜잭션 = 2008-04-22-19.44.25.000000
Local
```

```
DB20000I ROLLFORWARD 명령이 완료되었습니다.
```

10. 원래의 창에서 DEPT_T 테이블을 조회합니다. "D00" 부서 번호가 있는지 확인합니다.

```
C:\WLABWUNIT11> db2 "select count(*) from dept_t
```

```
1
-----
      15
1 레코드가 선택되었습니다.
```


11. 이번에는 일부러 Consistency 가 깨지도록 유도합니다. 즉 DEPT_T 테이블에는 부서가 존재하지 않으나 EMP_T 테이블에는 그 부서에 속한 직원이 있도록 합니다. EMPTBS 와 DEPTTBS 테이블 스페이스를 DROP 한후 다시 2번부터 7번까지 restore 를 모두 수행한 후 다음을 수행합니다. Restore를 수행했던 창에서 다시 rollforward를 수행합니다. 이 때 rollforward는 로그를 적용하지 않은채 완료합니다.

```
C:\WLABWUNIT11> db2 rollforward db sample complete
tablespace(depttbs) online
```

```
SQL4907W "sample" 데이터베이스가 복구되었지만 롤 포워드 조작
을 위해 포함된 테이블 스페이스 목록에서 하나 이상의 테이블이 무결성
설정 보류 상태에 있습니다.
```

37 시나리오 4 : 테이블 스페이스 백업 및 복구

백업 및 복구 비교

Point  명령어를 이용하여 DB2 백업 및 복구를 수행합니다. 테이블 스페이스 백업 및 rollforward 복구를 수행하기 위해서는 로깅 모드가 아카이브 모드여야 합니다.

11. 로그가 적용되지 않은채 테이블 스페이스가 복구될 경우 DEPT_T 테이블에는 존재하지 않는 "D00" 부서가 EMP_T 테이블에는 그 부서에 속한 직원이 존재하므로 Consistency가 이루어 지지 않습니다 다음과 같이 EMP_T 테이블은 무결성 설정 보류 상태가 되며 이를 위해서는 EMP_T 테이블을 "SET INTEGRITY " 명령어를 이용하여 보류 상태에서 풀어주어야 합니다.

```
C:\WLABWUNIT11> db2 "select * from emp_t
```

```
SQL0668N 이유 코드 "1"(으)로 인해 테이블 "KR010389.EMP_T"에서 조작이 허용되지 않습니다.
SQLSTATE=57016
```

```
C:\WLABWUNIT11> db2 "select * from dept_t
```

DEPTNO	DEPTNAME	MGRNO
ADM	ADM DEPT	
A00	SPIFFY COMPUTER SERVICE DIV.	000010 A00

12. SET INTEGRITY 명령어를 이용하여 EMP_T 테이블을 무결성 설정 보류 상태에서 빠져 나오도록 합니다. 일단 SET INTEGRITY 명령어를 수행하기에 앞서 Consistency가 이루어지지 않는 레코드를 저장할 EXCEPTION 테이블을 생성한 후에 SET INTEGRITY 명령어를 수행합니다.

```
C:\WLABWUNIT11> notepad s4_setigt.db2
```

```
CREATE TABLE EMPT_EXP LIKE EMP_T;
ALTER TABLE EMPT_EXP ADD COLUMN ts TIMESTAMP;
ALTER TABLE EMPT_EXP ADD COLUMN msg
CLOB(32K);
SET INTEGRITY FOR EMP_T IMMEDIATE CHECKED
FOR EXCEPTION IN EMP_T USE EMPT_EXP;
```

```
C:\WLABWUNIT11> db2 -tvf s4_setigt.db2
```


```
SET INTEGRITY FOR EMP_T IMMEDIATE CHECKED FOR
EXCEPTION IN EMP_T USE EMPT_EXP
SQL3602W 데이터 점검 처리시 제한조건 위반이 발견되어 예외테이
블로 이동시켰습니다. SQLSTATE=01603
```

13. EMPT_EXP 테이블을 조회하여 어떤 레코드가 삽입되었는지 확인합니다.

```
C:\WLABWUNIT11> db2 select empno from empt_exp
```

```
EMPNO
-----
010333
```


38 시나리오 5: 삭제된 테이블 복구

Point  테이블 스페이스 백업 이미지를 이용하여 삭제된 테이블을 복구할 수 있습니다.

1. DB CFG 의 파라미터를 확인하여 DB가 현재 Circular 인지 Archive 모 드인지 확인합니다.

```
c:\W> db2 get db cfg for sample | find "LOGARCHMETH1"
```

```
첫 번째 로그 아카이브 메소드(LOGARCHMETH1) =
DISK:c:\warch_logs
```

2. 현재 SAMPLE DB에 접속하여 dms01 테이블 스페이스와 USR,SCHED 테이블을 생성 및 데이터를 입력합니다. 테이블 스페이스 생성시 Dropped table Recovery 옵션을 "ON" 으로 설정하여 삭제된 테이블 도 복구가 가능하도록 설정합니다.

```
c:\W> notepad crtbs5.db2
```

```
CONNECT TO SAMPLE;
DROP TABLESPACE DMS01;
CREATE TABLESPACE DMS01 MANAGED BY AUTOMATIC
STORAGE DROPPED TABLE RECOVERY ON;
CREATE TABLE USR LIKE ADEFUSR in DMS01;
CREATE TABLE SCHED LIKE CL_SCHED in DMS01;
INSERT INTO USR SELECT * FROM ADEFUSR;
INSERT INTO SCHED SELECT * FROM CL_SCHED;
```

```
c:\W> db2 -tvf crtbs5.db2
```

3. USR 테이블과 SCHED 테이블을 조회하여 결과값을 저장합니다.

```
c:\W>db2 SELECT * FROM USR
```

```
WORKDEPT  NO_OF_EMPLOYEES---
A00          5
B01          1
C01          4
D11         11
D21          7
E01          1
E11          7
E21          6
```

8 레코드가 선택되었습니다.

```
c:\W>db2 SELECT * FROM SCHED
```

```
CLASS_CODE DAY_  STARTING_  ENDING_
042:BF      4   12:10:00  14:00:00
553:MJA     1   10:30:00  11:00:00
543:CWM     3   09:10:00  10:30:00
778:RES     2   12:10:00  14:00:00
044:HD      3   17:12:30  18:00:00
```

5 레코드가 선택되었습니다.

4. 테이블 스페이스 dms01 을 백업합니다.

```
c:\W>db2 backup db sample tablespace dms01 to
c:\wbackup
```

백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은 20080422133351 입니다.

5. USR 테이블과 SCHED 테이블에 새로운 데이터를 입력하고 데이터를 조 회하여 총 행의 개수를 확인합니다.

```
c:\W>notepad insert5.db2
```

```
insert into USR values ('F00',10);
insert into USR values ('G00',20);
insert into USR values ('H00',30);
insert into USR values ('I00',40);
insert into USR values ('J00',50);
insert into USR values ('K00',60);
insert into SCHED values ('000',5, '17:58:42','18:58:42');
insert into SCHED values ('001',5, '17:59:18','18:59:18');
insert into SCHED values ('002',10,'17:59:18','18:59:18');
insert into SCHED values ('003',15,'17:59:18','18:59:18');
insert into SCHED values ('004',20,'17:59:18','18:59:18');
```

```
c:\W>db2 -tvf insert5.db2
```

```
c:\W>db2 select count(*) from usr
```

```
1
-----
14
```


```
c:\W>db2 select count(*) from sched
```

```
1
-----
10
```

6. 테이블 SCHED 를 Drop 합니다.

```
c:\W> db2 drop table sched
```

39 시나리오 5: 삭제된 테이블 복구

Point  테이블 스페이스 백업 이미지를 이용하여 삭제된 테이블을 복구할 수 있습니다.

7. list history 명령어를 이용하여 Drop된 테이블 및 테이블이 위치한 테이블 스페이스 정보를 조회합니다.

```
c:\W>db2 list history dropped table all for sample
```

```
Op Obj 시간소인+순서 유형 Dev 최초 로그 현재 로그 백업 ID
-----
D T 20080422135257
0000000000008d1700070005

"KR010389"."SCHED"이(가) 1 테이블 스페이스에 있음:

00001 DMS01

주석: DROP TABLE
시작 시간: 20080422135257
종료 시간: 20080422135257
상태: A

EID: 162

DDL: CREATE TABLE "KR010389"."SCHED"
("CLASS_CODE" CHAR(7), "DAY" SMALLINT,
"STARTING" TIME, "ENDING" TIME) IN "DMS01";
```

8. 테이블이 Drop 되기전에 백업받았던 테이블 스페이스 백업 이미지를 이용하여 복구합니다.

```
c:\W>db2 force applications all
c:\W>db2 restore db sample tablespace (dms01) taken at
20080422133351
```

```
DB20000I RESTORE DATABASE 명령이 완료되었습니다.
```

9. DB 에 접속하여 dms01의 테이블 스페이스 상태를 확인합니다.

```
c:\W> db2 connect to sample
c:\W> db2 list tablespaces
```

```
이름 = DMS01
유형 = 데이터베이스 관리 스페이스
내용 = 모든 영구 데이터. 일반 테이블 스페이스
상태 = 0x0080
세부사항 설명: 롤 포워드 보류
```

10. 삭제된 테이블의 데이터 파일이 생성될 디렉토리를 생성합니다. 여기에서는 c:\WtempWexport 를 생성하도록 합니다.

```
c:\W>mkdir c:\WbackupWexport
```

11. 7번의 화면에 보이는 백업 ID 를 이용하여 rollforward 명령어를 수행하여 삭제된 테이블을 복구합니다.

```
c:\W> db2 force applications all
c:\W> db2 rollforward db sample to end of logs tablespace
(dms01) recover dropped table
0000000000008d1700070005 to c:\WbackupWexport
```

```
롤 포워드 상태

입력 데이터베이스 별명      = sample
상태를 리턴한 노드 수      = 1
노드 번호                    = 0
롤 포워드 상태              = 보류하지 않음
다음에 읽을 로그 파일      =
처리된 로그 파일            = -
최종 커밋된 트랜잭션        = 2008-04-21-23.49.56.000000 UTC

DB20000I ROLLFORWARD 명령이 완료되었습니다.
```

12. rollforward가 완료되면 위에서 생성했던 c:\WbackupWexport 에 Node0000 라는 디렉토리가 생성되며 그 디렉토리안에 data 라는 파일이 생성됩니다. 일반 Editor 프로그램을 이용하여 data 라는 파일을 열어보면 테이블이 삭제되기 전까지의 데이터를 확인하실 수 있습니다.

```
c:\W>cd c:\WbackupWexport\WNODE*
```


13. 위의 복구 프로세스의 1번에서 수행했던 list history dropped table 결과화면에 보면 삭제된 테이블의 DDL 명령어를 볼 수 있습니다. 이 DDL 명령어를 이용하여 삭제된 테이블을 재생성합니다.

```
c:\W>notepad c:\tab5.db2
```

```
CONNECT TO SAMPLE;
CREATE TABLE SCHED (
CLASS_CODE      CHAR(7) ,
DAY              SMALLINT ,
STARTING        TIME ,
ENDING TIME ) IN DMS01;
```

```
c:\W>db2 -tvf c:\tab5.db2
```

40 시나리오 5: 삭제된 테이블 복구

Point  테이블 스페이스 백업 이미지를 이용하여 삭제된 테이블을 복구할 수 있습니다.

14. LOAD 명령어를 통해 export 받은 데이터 파일을 새로 만든 테이블에 Load 합니다.

```
c:\W> db2 load from c:\Wbackup\Wexport\WNODE0000\data
of del insert into sched
```

SQL3109N 유틸리티가 파일 "c:\Wbackup\Wexport\WNODE0000\data"에서 데이터를 로드하기 시작합니다.

SQL3500W 유틸리티가 "2008-04-22 14:54:10.955933"에서 "LOAD" 단계를 시작중입니다.

SQL3519W 일관성 지점 로드 시작. 입력 레코드 계수 = "0".

SQL3520W 일관성 지점 로드에 성공했습니다.

SQL3110N 유틸리티가 처리를 완료했습니다. 입력 파일에서 "10"개의 행을 읽었습니다.

SQL3519W 일관성 지점 로드 시작. 입력 레코드 계수 = "10".

SQL3520W 일관성 지점 로드에 성공했습니다.

SQL3515W 유틸리티가 "2008-04-22 14:54:11.215503"에서 "LOAD" 단계를 완료했습니다.

읽은 행 수	= 10
건너뛴 행 수	= 0
로드된 행 수	= 10
거부된 행 수	= 0
삭제된 행 수	= 0
커밋된 행 수	= 10

15. USR 테이블과 SCHED 테이블을 조회하여 Drop되기 전까지 데이터가 존재하는지 확인합니다. .

```
c:\W>db2 SELECT * FROM USR
```

WORKDEPT	NO_OF_EMPLOYEES
A00	5
B01	1
C01	4
D11	11
D21	7
E01	1
E11	7
E21	6
F00	10
G00	20
H00	30
I00	40
J00	50
K00	60

14 레코드가 선택되었습니다.

```
c:\W>db2 SELECT * FROM SCHED
```

CLASS_CODE	DAY	STARTING	ENDING
042:BF	4	12:10:00	14:00:00
553:MJA	1	10:30:00	11:00:00
543:CWM	3	09:10:00	10:30:00
778:RES	2	12:10:00	14:00:00
044:HD	3	17:12:30	18:00:00
000	5	17:58:42	18:58:42
001	5	17:59:18	18:59:18
002	10	17:59:18	18:59:18
003	15	17:59:18	18:59:18
004	20	17:59:18	18:59:18

10 레코드가 선택되었습니다.

백업 및 복구 관련 Q/A

- 오라클은 백업 및 복구시 OS Copy 명령어 또는 RMAN 에서 COPY 명령어를 이용하여 이미지 복사로 백업 및 복구를 수행할 수 있는데 DB2는 이와 같은 기능이 없는지요? 만약 OS COPY 명령어를 이용하여 백업 및 복구 수행시 문제가 발생하나요?

DB2는 DB2에서 제공하는 BACKUP 및 RESTORE 명령어를 이용하여 백업 및 복구를 수행하는 것을 권고합니다. 물론 OS 명령어를 이용하여 백업 및 복구를 수행할 수 있으나 DBA의 개입없이 백업 및 복구를 수행하기에는 불편한 점이 많으며 복잡합니다. 오라클 역시 이미지 복사로 이용할 경우 복잡하며 DBA 개입없이 거의 불가능합니다.

- 오라클은 RMAN 을 이용하여 데이터 파일을 다른 위치로 변경할 수 있는데 DB2도 가능한지요?

DB2도 경로를 재지정하여 복구하는 것이 가능합니다. 복구 후에 "SET TABLESPACE CONTAINERS FOR <TABLESPACE ID> " 명령어를 이용하여 경로를 재지정합니다.

- 오라클은 RMAN 을 이용한 백업 및 복구시 "Challe"을 Allocate 하여 백업 및 복구할 디바이스 타입 및 parallelism 을 지정할 수 있습니다. DB2는 백업 및 복구를 수행할 디바이스 및 Parallelism 을 어떻게 지정하는지요?

DB2는 디바이스 타입 및 parallelism 을 Backup 또는 Restore 명령어에서 옵션을 이용하여 지정할 수 있습니다.

- 오라클은 9i에서부터 FLASHBACK 기능을 제공하여 특정 시점으로 데이터를 돌려 놓을 수 있습니다. DB2도 가능한지요?

DB2는 Recovery Expert 라는 툴을 이용하여 Flashback 기능과 유사한 기능을 제공합니다. 그러나 Recovery Expert 가 Flashback 기능을 완전히 대체하지는 못합니다. 오라클의 FLASHBACK 기능을 사용하기 위해서는 자동 UNDO 관리시스템을 사용해야 하며 UNDO_RETENTION 파라미터를 지정하여 어느 시점까지 UNDO 정보를 보유할 것인지를 설정해야 합니다. 이는 공간 및 관리적인 비용이 많이 소요되는 작업입니다.

- 오라클은 자동 백업 및 Incremental 또는 Delta 복구를 자동으로 수행하는 기능이 있는데 DB2도 제공하는지요?

DB2는 자동 백업 및 자동 복구 기능을 제공하여 DBA 개입없이도 편리하게 백업 및 복구 작업을 수행할 수 있도록 합니다. 자동 백업을 위해서는 데이터베이스 구성 파일의 파라미터인 AUTO_MAINT 와 AUTO_DB_BACKUP 이 모두 "ON" 으로 되어 있어야 하며 AUTOMAINT_SET_POLICY 스토어드 프로시저를 통해서 자동 백업 환경을 구성하면 됩니다. 자동 복구를 위해서는 RESTORE 명령어에서 "AUTOMATIC" 옵션을 사용하록 합니다. 자세한 사항은 DB2 9.5 도움말을 참고하시기 바랍니다.



Unit 12 : 모니터링 비교

오라클 사용자를 위한 DB2 가이드



진단 및 Trace 비교

모니터링 방법 비교

DB2 모니터링 소개


DB2 스냅샷 모니터링 소개

DB2 이벤트 모니터링 소개

DB2의 db2pd 툴 소개

DB2 활동 모니터 소개

01 진단 및 Trace 파일 비교

Point  오라클과 DB2는 DBMS Engine에 오류나 경고가 발생할 때 진단 파일에 기록하여 DBA에게 알려줍니다. 다음은 오라클과 DB2에서 제공하는 진단 파일의 종류 및 역할을 비교합니다.

진단 파일 비교

	오라클의 Alert Log 파일	DB2의 db2diag.log 파일
설명	<ul style="list-style-type: none"> • 메시지와 오류가 시간 순으로 기록됩니다. • alert log file에 기록되는 정보는 다음과 같습니다. <ul style="list-style-type: none"> • 내부 오류(ORA-600) 및 블락 훼손 오류 (ORA-1578) • 데이터베이스 구조, 파라미터 및 CREATE DATABASE, STARTUP, SHUTDOWN, ARCHIVE LOG, RECOVER문 등에 영향을 주는 작업. • Instance 시작시 기본 값으로 설정되어 있지 않은 모든 초기화 파라미터 값. • 검사 후에는 파일을 정기적으로 제거 또는 정리합니다. 	<ul style="list-style-type: none"> • 데이터베이스 시스템에 오류가 발생하면 db2diag.log에 기록됩니다. • DIAGLEVEL 인스턴스 구성 변수 설정에 의해 기록할 오류의 수준이 결정됩니다. • db2diag 툴을 이용하여 db2diag.log 파일을 해석할 수 있으며 특정 정보만을 추출하여 볼 수도 있습니다. • DB2 지원센터에서는 이러한 Trace File을 사용하여 문제를 진단하고 해결합니다. • 검사 후에는 파일을 정기적으로 제거 또는 정리합니다.
파일 위치	<ul style="list-style-type: none"> • BACKGROUND_DUMP_DEST 파라미터에 의해 위치가 지정됩니다. 	<ul style="list-style-type: none"> • DIAGPATH 인스턴스 구성 변수에 의해 위치가 지정되며 DEFAULT 위치는 <인스턴스 홈 디렉토리>/sql/lib/db2dump 입니다.

Trace 파일 비교


	오라클의 Trace File	DB2의 Trace File
설명	<ul style="list-style-type: none"> • 백그라운드 Process Trace File 와 User Trace 파일이 존재합니다. • 백그라운드 프로세스 Trace File <ul style="list-style-type: none"> • 백그라운드 프로세스가 감지한 오류에 대한 정보를 Trace File에 덤프합니다. • User Trace File <ul style="list-style-type: none"> • 서버 프로세스 단위로 생성됩니다. • 인스턴스 레벨 또는 세션 레벨에서 설정 또는 해제할 수 있습니다. 	<ul style="list-style-type: none"> • 오라클의 User Trace File과 유사하며 오라클의 백그라운드 프로세스 Trace File 정보와 User Trace File 정보의 일부는 db2diag.log 파일에 기록됩니다. • DB2 instance 성능 저하의 원인이 되므로 DB2 지원센터의 요청이 있거나 오류가 반복될 때 Trace 를 시작합니다.
Trace 설정 방법	<ul style="list-style-type: none"> • 인스턴스 레벨 <ul style="list-style-type: none"> • SQL_TRACE 초기화 파라미터에 의해 설정 및 해제 • 세션 레벨 <ul style="list-style-type: none"> • ALTER SESSION SET SQL_TRACE = TRUE 	<ul style="list-style-type: none"> • db2trc 명령어를 이용하여 on/off 합니다. <ul style="list-style-type: none"> • db2trc on -l 4m



오라클과 DB2는 모두 진단 파일을 통해 오류의 원인 및 문제점을 파악합니다. Trace 파일은 진단파일의 정보보다 더 자세한 정보를 알고 싶을 경우 이용합니다.

오라클 진단 파일과 DB2의 진단 파일의 정보는 아키텍처가 다르므로 보여주는 정보가 다를 수 있으므로 이 점 유의하시기 바랍니다.

02 성능 모니터링 뷰 및 유틸리티 비교


Point  다음은 모니터링 툴로 데이터베이스 및 애플리케이션의 성능을 향상시키고 문제점을 파악하는데 사용됩니다. 아래 표는 오라클과 DB2 시스템을 모니터링할 수 있는 툴을 비교한 것입니다. 그러나 오라클과 DB2의 모니터링 툴은 유사할 뿐 동일하지는 않습니다.

성능 모니터링 뷰 및 유틸리티

오라클	DB2	설명
<ul style="list-style-type: none"> Oracle Enterprise Manager 	<ul style="list-style-type: none"> DB2 Control Center 	<ul style="list-style-type: none"> 데이터베이스 관리를 지원하는 다양한 GUI 툴이 있습니다.
<ul style="list-style-type: none"> Diagnostics 및 Tuning Pack 	<ul style="list-style-type: none"> DB2 Health Center 이벤트 분석기 활동 모니터 	<ul style="list-style-type: none"> 그래픽 모니터링, 분석, 자동 튜닝 등 여러가지 성능 관리 분야를 다루는 다양한 관리 툴이 포함되어 있습니다.
<ul style="list-style-type: none"> Statspack 	<ul style="list-style-type: none"> 스냅샷 모니터 및 이벤트 모니터 	<ul style="list-style-type: none"> 시스템 모니터링 정보를 수집하는데 사용됩니다.
<ul style="list-style-type: none"> v\$XXX 동적 문제 해결 및 성능 뷰 	<ul style="list-style-type: none"> SYSIBMADM.XXX 의 테이블 정보 	<ul style="list-style-type: none"> 오라클의 Dynamic View 에서 성능 및 모니터링 정보를 보여주듯이 DB2에서도 SYSIBMADM 에 속하는 여러 테이블이 성능 및 모니터링 정보를 보여줍니다.
<ul style="list-style-type: none"> dba_XXX 디셔너리 뷰 	<ul style="list-style-type: none"> SYSIBM.XXX 및 SYSSTAT.XXX의 테이블 정보 	<ul style="list-style-type: none"> 데이터 저장 영역 통계 자료 및 인덱스 저장영역, 컬럼 저장 정보등이 포함되어 있습니다.
<ul style="list-style-type: none"> 오라클 대기 이벤트 	<ul style="list-style-type: none"> 스냅샷 모니터 및 이벤트 모니터 	<ul style="list-style-type: none"> 자원을 획득하기 위해 발생하는 대기현상을 기록하고 관찰하는 일련의 기능을 오라클 대기 이벤트라 합니다. DB2에서는 대기 이벤트라는 개념은 존재하지 않지만 자원을 획득하기 위해 발생하는 대기현상을 스냅샷 모니터 및 이벤트 모니터를 통해 기록 및 관찰하고 해결합니다.


부연설명

오라클과 DB2를 비교할 때 가장 어려운 부분이 모니터링 부분입니다. 앞에서 설명한 아키텍처 및 오브젝트 관리, 데이터 관리등은 기본 RDBMS 개념이 유사하므로 비교가 가능하나 모니터링 부분은 RDBMS의 기본 개념을 바탕으로 만들어진 부분이 아니라 각 제품의 성능 및 문제점을 파악하기 위해 만들어진 기능이므로 각기 다른 특성과 장점을 가지고 있어 일대일로 기능을 비교하기 어렵습니다. 이 점 참고하시기 바랍니다.

Point  DB2에서는 다음과 같은 목적을 위해 모니터링을 수행합니다. 아래 목적 및 단계를 확인하여 모니터링이 필요할 경우 아래의 순서에 따라 수행하도록 합니다.

DB2 시스템 모니터 목적

- 모니터링의 목적 :
 - 데이터베이스 및 애플리케이션의 성능을 향상시킵니다.
 - Configuration 파라미터를 튜닝합니다.
 - 문제점을 파악합니다.
 - 사용자 및 애플리케이션의 활동에 대해 살펴보고 튜닝할 수 있습니다.

데이터베이스 모니터링 순서

1. 모니터를 하기 위한 목적을 정의합니다.
2. 분석할 정보를 결정합니다.
3. 사용할 모니터의 타입을 결정합니다.
4. 모니터 데이터를 확보합니다.
5. 모니터 정보를 분석합니다.
6. 모니터 정보를 분석하여 성능 또는 문제해결을 위해 변경되어야 할 사항이 무엇인지 확인합니다.
7. 변경사항을 적용합니다.

DB2 모니터링 툴

- 스냅샷 모니터 (Snapshot Monitor)
 - 현재 시점의 인스턴스 및 데이터베이스, 응용 프로그램등의 상태에 대한 모니터링 정보를 제공합니다.
- 이벤트 모니터 (Event Monitor) 와 이벤트 분석기 (Event Analyzer)
 - 일정 기간의 특정 이벤트에 대한 모니터링 정보를 제공하며 이벤트 분석기를 통해 결과값을 분석합니다.
- Health Monitor 와 Health Center
 - Health Indicator를 통해 인스턴스의 건강 지수를 정의하고 상태를 체크하여 Health Center를 통해 건강 상태를 보여줍니다. 또한 DBA에게 잠재적인 문제점등에 대해 E-Mail로 통보합니다.
- 인다우트 트랜잭션 관리 프로그램 (Indoubt Transaction Manager)
 - 인다우트 트랜잭션 (예를 들면 2 Phase Commit에서 한쪽 회선이 끊어져 Commit 또는 Rollback을 못할 경우) 이 필요한 자원 (예를 들면 테이블 및 인덱스에 대한 잠금, 로그 스페이스 및 트랜잭션에 사용되는 스토리지 등)을 독점할 경우 인다우트 트랜잭션을 조희하여 조치할 수 있습니다.
- 활동 모니터 (Activity Monitor)
 - 활동 모니터를 이용하여 응용프로그램 성능 및 동시성, 자원 소비 및 데이터베이스 또는 데이터베이스 파티션의 SQL문 사용을 모니터링 합니다. 활동 모니터는 모니터 데이터의 특정 서브세트를 기본으로 사전 정의된 보고서 세트를 제공합니다.
- Memory Visualizer
 - Memory Visualizer를 사용하여 인스턴스 및 이에 해당하는 모든 데이터베이스의 메모리 관련 성능을 모니터링합니다.
- db2pd 도구
 - db2pd 도구는 DB2 메모리 세트의 빠르고 즉각적인 정보를 포함하는 문제점 판별 도구입니다. 이는 래치를 확보하거나 엔진 자원을 사용하지 않고 정보를 수집합니다.



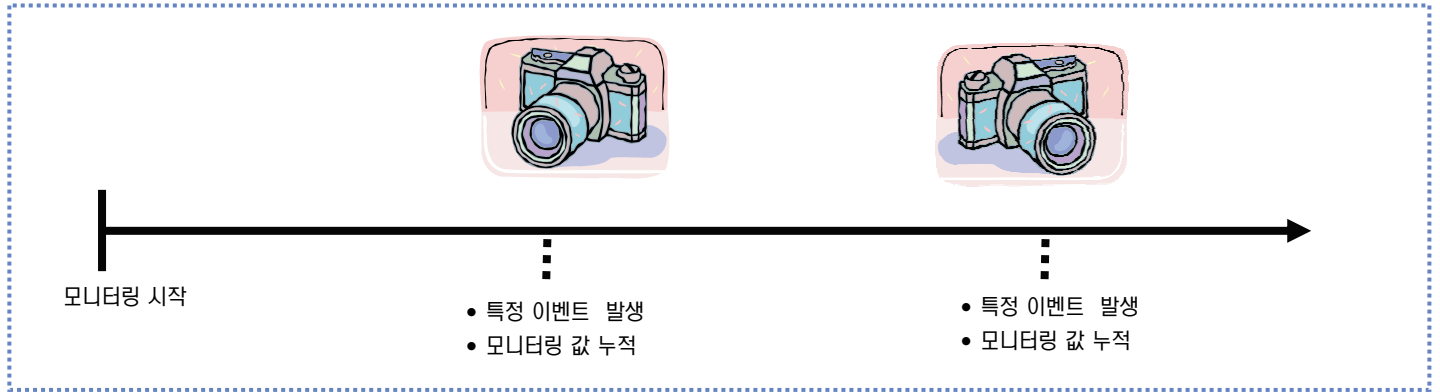
DB2에서는 다양한 모니터링 툴을 제공합니다. 목적 및 시스템 부하를 고려하여 모니터링 툴을 선택하도록 합니다.

모니터링 비교

	스냅샷 모니터	이벤트 모니터	Health monitor
설명	<ul style="list-style-type: none"> 실시간 모니터링 툴입니다. 현재 시점의 데이터베이스 상태를 제공합니다. 데이터베이스 상태 체크, 잠재적인 문제점 파악 등에 사용됩니다. 일정 간격으로 모니터링할 경우 데이터베이스의 활동에 대한 추이를 알 수 있습니다. 	<ul style="list-style-type: none"> 실시간, 트리거 기반의 모니터링 툴입니다. 특정 타입의 이벤트가 일어날 때마다 일정 시간 동안 데이터베이스의 활동을 분석할 수 있습니다. 문제점 파악 및 정확한 문제점을 찾기 위한 자세한 데이터를 제공합니다. 	<ul style="list-style-type: none"> 예외 기반의 모니터링 툴입니다. 데이터베이스의 비정상적 상황이나 잠재적 문제점에 대해 알려줍니다. 데이터베이스의 상태에 대한 정보를 제공합니다.
수집 데이터 수준	<ul style="list-style-type: none"> 데이터베이스 매니저 (인스턴스) 데이터베이스 애플리케이션 (구문 수준의 정보 포함) 버퍼풀 테이블 스페이스 테이블 Lock 및 Lock Wait Dynamic SQL DCS 애플리케이션 및 데이터베이스 	<ul style="list-style-type: none"> 데이터베이스 Connection 버퍼풀 테이블 스페이스 테이블 Deadlock 트랜잭션 Statement 	<ul style="list-style-type: none"> 데이터베이스 매니저 (인스턴스) 데이터베이스 테이블 스페이스 테이블 스페이스 컨테이너
활성화 방법	<ul style="list-style-type: none"> 특정 모니터 스위치를 'ON' 합니다. TIMESTAMP는 디폴트로 ON 으로 설정되어 있습니다. 스위치는 인스턴스 또는 애플리케이션 수준에서 활성화 설정이 가능합니다. 	<ul style="list-style-type: none"> AUTOSTART 옵션을 이용하여 이벤트 모니터를 생성합니다. 이벤트 모니터를 State 1로 설정합니다. 	<ul style="list-style-type: none"> 기본적으로 활성화 되어 있습니다. 비활성화하기 위해 health_mon 이라는 인스턴스 구성 파라미터를 OFF로 설정합니다.
데이터 수집 시점	<ul style="list-style-type: none"> 사용자가 스냅샷 테이블 함수, SNAPSHOT 관리 뷰 또는 CLP에서 "GET SNAPSHOT" 명령어 수행시 수집됩니다. 스냅샷 API, SNAP_WRITE_FILE 이라는 Stored 프로시저 호출시에도 수집됩니다. 	지정된 이벤트가 일어날때 데이터가 수집됩니다.	미리 지정된 간격마다 기본적으로 수집됩니다.
데이터 확인 방법	<ul style="list-style-type: none"> 스냅샷 테이블 함수, 스냅샷 관리 뷰 또는 SNAPSHOT 명령어를 통해 결과값을 확인합니다. 활동 모니터를 이용하여 확인할 수 있습니다. 	<ul style="list-style-type: none"> 이벤트 분석기 툴 (Event Analyzer)를 이용하여 데이터를 분석합니다. 테이블 이벤트 모니터의 경우(즉 이벤트 모니터의 결과값을 테이블에 저장할 경우) SQL문을 이용하여 이벤트 테이블을 조회합니다. named pipe 또는 파일 이벤트 모니터의 경우 db2evmon 유틸리티를 이용할 수도 있습니다. 	<ul style="list-style-type: none"> e-mail 또는 Pager로 경고 메시지를 받습니다. SQL 테이블 함수, snapshot 명령어, health snapshot API 를 이용하여 health data를 읽습니다. Health Center를 이용하여 현재 경고 메시지를 확인합니다. 경고를 분석하여 문제가 될 수 있는 것을 파악하여 여러 Advisor 툴을 이용해 권장값을 얻어 해결합니다.
오버헤드	활성화된 모니터 스위치의 수 및 인스턴스내에서 운영되는 워크로드 타입에 따라 달라지나 대략 3 ~ 10% 정도 시스템 워크로드가 증가합니다.	모니터되는 데이터의 타입과 이벤트 모니터의 조건에 따라 달라집니다.	Health 모니터는 최소한의 오버헤드가 필요하며 그래픽 툴이 시작될때 추가적인 오버헤드가 필요합니다.

05 DB2 스냅샷 모니터

Point 실시간 모니터링 톨로 스냅샷 함수, 스냅샷 API, 또는 스냅샷 명령어를 이용하여 데이터베이스의 활동 상황을 모니터링 할 수 있습니다. 이 곳에서는 스냅샷 모니터를 활성화 시키는 방법 및 데이터 수집등에 대해 알아보도록 합니다.



스냅샷 모니터링 종류


	Instance Level 모니터링	응용 프로그램 Level 모니터링
설명	<ul style="list-style-type: none"> 인스턴스가 시작되면 모니터 스위치가 'ON'된 대상에 대해 항상 모니터링합니다. 오버헤드가 많이 발생하므로 되도록이면 응용 프로그램 Level 에서 모니터링을 활성화 시키도록 합니다. 	<ul style="list-style-type: none"> 현재 세션의 모니터 스위치의 값을 설정하면 인스턴스 구성 변수에 설정된 시스템 모니터 스위치의 값보다 우선적으로 적용됩니다.

스냅샷 모니터 스위치

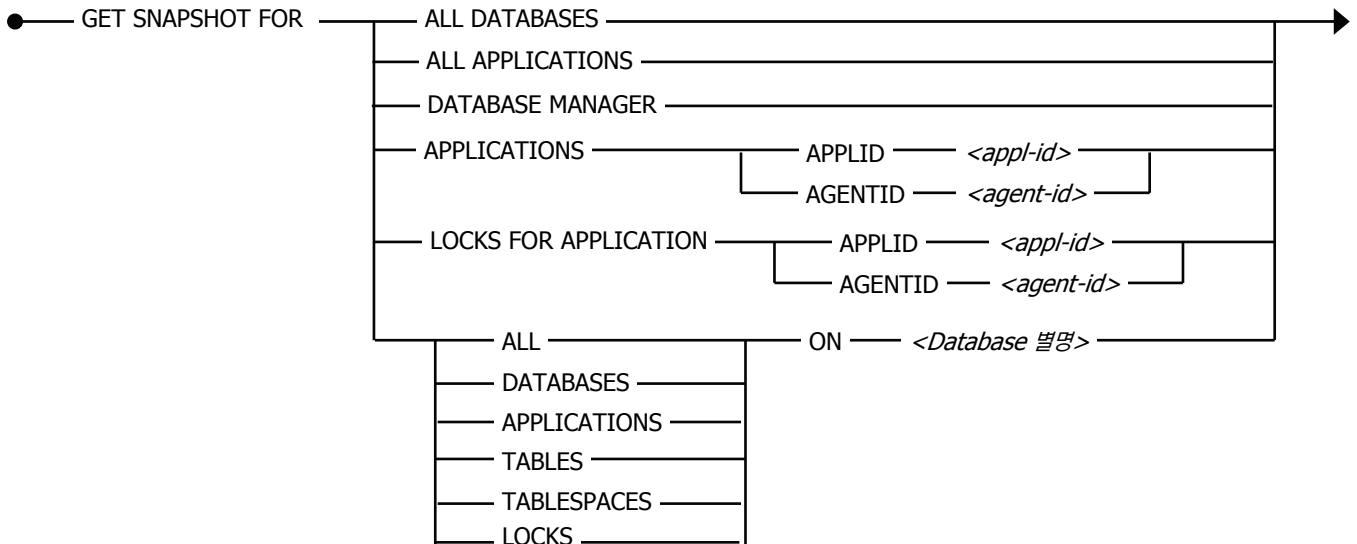
모니터 스위치	DBM 파라미터	설명
BUFFERPOOL	DFT_MON_BUFPOOL	• 읽기 및 쓰기 의 횟수 및 취해진 시간
LOCK	DFT_MON_LOCK	• Lock Wait 시간 및 Deadlocks
SORT	DFT_MON_SORT	• 사용된 Heap 의 수 및 Sort 성능
STATEMENT	DFT_MON_STMT	• 시작 및 종료 시간 과 Statement ID
TABLE	DFT_MON_TABLE	• 활동 내역 (읽고 쓴 행의 수)
UOW	DFT_MON_UOW	• 시작 및 종료 시간 및 완료 상황
TIMESTAMP	DFT_MON_TIMESTAMP	• TIMESTAMP - 디폴트로 'ON' 되어 있습니다.

모니터 스위치 관련 명령어

명령어	설명
db2=> GET MONITOR SWITCHES	• 모니터 스위치의 설정값을 확인합니다.
db2 => UPDATE MONITOR SWITCHES USING <모니터 스위치명> [ON OFF]	• 모니터 스위치의 값을 설정합니다.
db2=> RESET MONITOR FOR [ALL DB <DB 명>	모니터의 누적된 값을 0 으로 설정합니다.

Point  모니터 스위치를 켜면 현재 시점의 데이터베이스의 활동 상황을 모니터링할 수 있습니다. 모니터링의 대상은 인스턴스, 데이터베이스, 애플리케이션이 될 수 있습니다. 모니터 스위치 설정값에 따라 표시되는 텍스트 정보가 달라집니다.

GET SNAPSHOT 구문



예문

db2=> get snapshot for application agentid 29

```

Application Snapshot
Application handle           = 29
Application status          = Lock-wait
Status change time         = Not Collected
Application code page       = 819
Application country/region code = 1
DUOW correlation token     = *LOCAL.jwr.070222182152
Application name            = db2bp
Application ID              = *LOCAL.jwr.070222182152
Sequence number            = 00001 TP Monitor client user ID =
TP Monitor client workstation name =
TP Monitor client application name =
TP Monitor client accounting string =
Connection request start timestamp = 02/22/2007 13:21:52.587168
Connect request completion timestamp = 02/22/2007 13:21:53.291779
Application idle time      =
CONNECT Authorization ID   = JWR
Client login ID            = jwr
Configuration NNAME of client = gilera
Client database manager product ID = SQL09050
Process ID of client application = 843852
Platform of client application = AIX 64BIT
Communication protocol of client = Local Client
Inbound communication address = *LOCAL.jwr
Database name              = SAMPLE
  
```

Point GET SNAPSHOT 명령어를 이용하여 아래와 같은 모니터링 정보를 확인할 수 있습니다.

스냅샷 명령어

Monitor level	CLP 명령어	설명
연결 목록	list applications [show detail]	현재 인스턴스가 관리하는 데이터베이스에 접속한 애플리케이션의 ID 정보를 제공합니다.
	list applications for database dbname [show detail]	현재 특정 데이터베이스에 접속한 애플리케이션의 ID 정보를 제공합니다.
인스턴스	get snapshot for dbm	인스턴스 레벨의 모니터 스위치 설정값을 포함한 정보를 제공합니다.
데이터베이스	get snapshot for database on dbname	적어도 하나의 애플리케이션이 특정 데이터베이스에 접속되어 있을 경우 그 데이터베이스에 대한 정보 및 Counter 정보를 제공합니다.
	get snapshot for all databases	적어도 하나의 애플리케이션이 데이터베이스에 접속되어 있을 경우 파티션에서 Active 한 데이터베이스의 정보 및 Counter 정보를 제공합니다.
	list active databases	인스턴스 안에서 현재 애플리케이션이 접속되어 있는 데이터베이스와 ACTIVATE DATABASE 명령어로 인해 Active 한 데이터베이스와 그 수를 알려줍니다.
애플리케이션	get snapshot for application applid appl-id get snapshot for application agentid appl-handle	누적된 값 및 상태 정보, 그리고 Statement 모니터 스위치가 "ON"되어 있을 경우 가장 최근에 실행된 SQL문을 제공합니다.
	get snapshot for applications [on dbname]	파티션의 해당 데이터베이스에 접속된 각 애플리케이션의 정보를 제공합니다. 누적된 값 및 상태 정보, 그리고 Statement 모니터 스위치가 "ON" 되어 있을 경우 가장 최근에 실행된 SQL문을 제공합니다.
테이블	get snapshot for tables on dbname	데이터베이스에 접속된 각 애플리케이션이 접근한 테이블 활동에 대한 정보를 제공합니다. 테이블 모니터 스위치가 설정되어야 합니다.
Lock	get snapshot for locks for application applid appl-id get snapshot for locks for application agentid appl-handle	특정 애플리케이션에서 획득한 Lock 목록을 보여줍니다. Lock Wait 정보는 Lock 모니터 스위치가 필요합니다.
	get snapshot for locks on dbname	데이터베이스에 접속한 애플리케이션에서 획득한 Lock 정보를 보여줍니다. Lock Wait 정보는 Lock 모니터 스위치가 필요합니다.
테이블 스페이스	get snapshot for tablespaces on dbname	데이터베이스의 테이블 스페이스 활동에 대한 정보를 제공합니다. 이럴 경우 버퍼풀 모니터 스위치가 필요합니다. 또한 컨테이너 정보도 제공합니다.
버퍼풀	get snapshot for all bufferpools [on dbname]	버퍼 풀 활동 및 counter 정보를 제공합니다. 이럴 경우 버퍼풀 모니터 스위치가 필요합니다.
Dynamic SQL	get snapshot for dynamic sql on dbname	데이터베이스의 SQL Statement 캐쉬에 있는 특정 시점의 Statement 정보를 보여줍니다.

참고: 이외에도 다양한 구문이 제공되므로 도움말을 참고하시기 바랍니다.

08 DB2 스냅샷 테이블 뷰 와 테이블 함수

Point GET SNAPSHOT 명령어의 실행 결과와 동일한 스냅샷 정보를 테이블의 형태로 저장하는 테이블 뷰와 테이블 함수입니다. 테이블 뷰의 경우 SYSIBMADM 스키마에 속합니다.

스냅샷 테이블 함수 및 뷰

Monitor level	테이블 함수	뷰	설명
애플리케이션	SNAP_GET_AGENT	SNAPAGENT	데이터베이스에 연결된 애플리케이션과 결합된 Agent 정보를 제공합니다.
	SNAP_GET_AGENT_MEMORY_POOL	SNAP_AGENT_MEMORY_POOL	Agent 레벨의 메모리 사용에 대한 정보를 제공합니다
	SNAP_GET_APPL	SNAPAPPL	데이터베이스에 연결된 각 애플리케이션에 대한 정보를 제공합니다. 이는 누적된 counter, 상태 정보 및 가장 최근에 실행된 SQL문 (Statement 모니터 스위치가 "ON" 일 경우) 에 대한 정보가 포함되어 있습니다.
	SNAP_GET_APPL_INFO	SNAPAPPL_INFO	데이터베이스에 연결된 각 애플리케이션에 대한 일반적인 정보를 제공합니다.
	SNAP_GET_STMT	SNAPSTMT	데이터베이스에 연결된 애플리케이션의 statement 에 대한 정보를 제공합니다. 이는 가장 최근에 실행된 SQL 문도 포함되어 있습니다. (STATEMENT 모니터 스위치가 "ON"일 경우)
	SNAP_GET_LOCKWAIT	SNAPLOCKWAIT	데이터베이스에 연결된 애플리케이션의 Lock wait 관련 정보를 제공합니다.
버퍼풀	SNAP_GET_BP	SNAPBP	특정 데이터베이스에 대한 버퍼풀의 활동에 대한 Counter를 제공합니다. BUFFER POOL 모니터 스위치가 "ON"으로 설정되어 있어야 합니다.
	SNAP_GET_BP_PART	SNAPBP_PART	파티션당 계산된 버퍼풀 크기, 사용현황에 대한 정보를 제공합니다.
LOCK	SNAP_GET_LOCK	SNAPLOCK	데이터베이스에 연결된 애플리케이션 레벨 및 데이터베이스 레벨의 Lock 정보를 제공합니다. Lock 모니터 스위치가 "ON"으로 설정되어야 합니다.
테이블	SNAP_GET_TAB	SNAPTAB	데이터베이스에 연결된 애플리케이션이 접근하는 테이블에 대한 활동 정보를 제공합니다. 반드시 TABLE 모니터 스위치는 "ON"으로 설정되어야 합니다.
	SNAP_GET_TAB_REORG	SNAPTAB_REORG	테이블 재구성시 테이블 재구성에 대한 정보를 제공합니다.
Dynamic SQL	SNAP_GET_DYN_SQL	SNAPDYN_SQL	데이터베이스의 SQL Statement 캐쉬에 있는 Statement 정보를 제공합니다.
데이터베이스	SNAP_GET_DB	SNAPDB	데이터베이스 레벨의 정보를 제공합니다. 그러나 데이터베이스에 적어도 하나의 애플리케이션이 접속했을 때만 결과값이 나옵니다.
	SNAP_GET_DB_MEMORY_POOL	SNAPDB_MEMORY_POOL	메모리에 대한 데이터베이스 레벨의 정보를 제공합니다. 다만 Unix Platform 의 경우에만 제공합니다.
	SNAP_GET_DETAILLOG	SNAPDETAILLOG	로그 파일에 대한 데이터베이스 레벨의 정보를 제공합니다.
	SNAP_GET_STORAGE_PATHS	SNAPSTORAGE_PATHS	각 저장 공간에 대한 파일 시스템 정보를 포함하여 데이터베이스의 자동 저장 Path 의 리스트를 제공합니다.

09 DB2 스냅샷 테이블 뷰 와 테이블 함수

Point GET SNAPSHOT 명령어의 실행 결과와 동일한 스냅샷 정보를 테이블의 형태로 저장하는 테이블 뷰와 테이블 함수입니다.

스냅샷 테이블 함수 및 뷰

Monitor level	테이블 함수	뷰	설명
테이블 스페이스	SNAP_GET_TBSP	SNAPTbsp	테이블 스페이스의 활동에 대한 정보를 제공하며 데이터베이스에 연결된 각 애플리케이션이 접근하는 테이블 스페이스에 대한 정보를 제공합니다. BUFFER POOL 모니터 스위치가 "ON"으로 설정되어 있어야 합니다.
	SNAP_GET_TBSP_PART	SNAPTbsp_PART	테이블 스페이스 구성에 대한 정보를 제공합니다.
	SNAP_GET_CONTAINER	SNAPCONTAINER	테이블 스페이스 레벨에서 테이블 스페이스의 컨테이너의 구성에 대한 정보를 제공합니다.
인스턴스	SNAP_GET_DBM	SNAPDBM	인스턴스 레벨의 정보를 제공합니다.
	SNAP_GET_SWITCHES	SNAPSWITCHES	인스턴스의 모니터 스위치 설정값을 제공합니다.
	SNAP_GET_DBM_MEMORY_POOL	SNAPDBM_MEMORY_POOL	메모리 사용에 대한 인스턴스 레벨의 정보를 제공합니다.

참고: 이외에도 다양한 테이블 함수 및 뷰가 제공되므로 도움말을 참고하시기 바랍니다.

조회 방법

```
db2=> SELECT * FROM TABLE ( SNAP_GET_APPL_INFO ('<DB 명>', -1 ) ) APPLINFO
```

테이블 함수 호출

테이블 함수 이름

DB 명과 파티션 번호 (모든 Partition 일 경우 -1 명시)

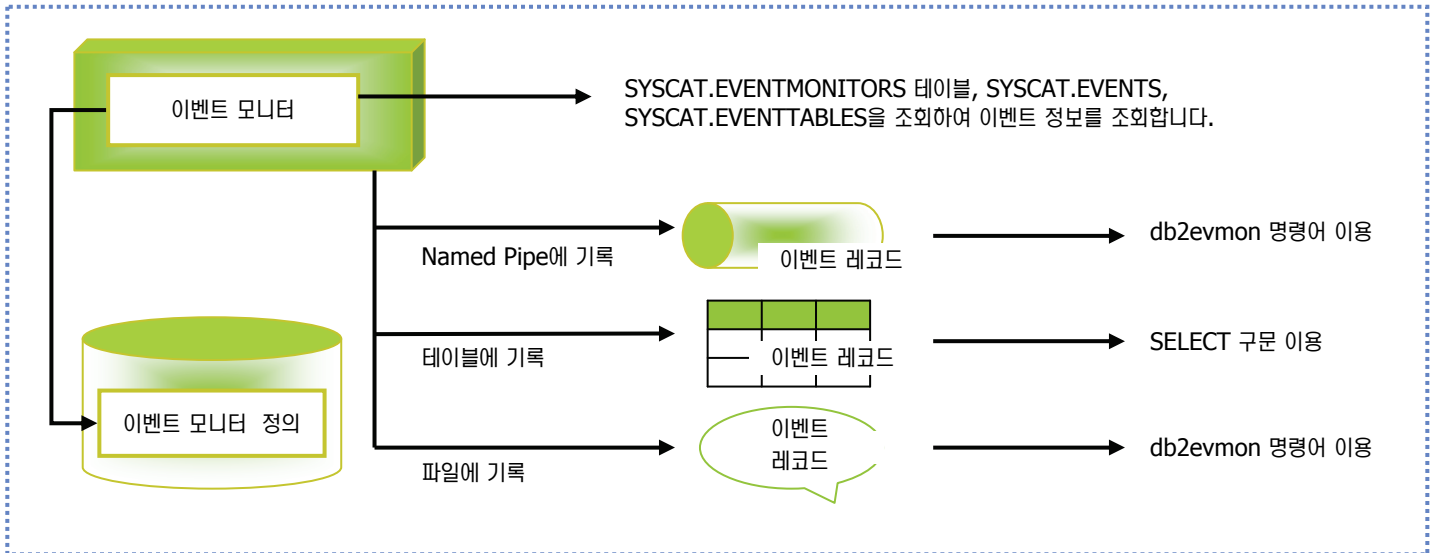
테이블 함수의 별명

```
db2=> SELECT * FROM SYSIBMADM.SNAPAPPL_INFO
```



DB2에서는 다양한 방법을 통해 스냅샷 모니터링 정보를 확인할 수 있습니다. 스냅샷 명령어, 스냅샷 테이블 함수, 스냅샷 뷰를 통해 모니터링 정보를 확인할 수 있으며 또한 활동 모니터(ACTIVITY MONITOR)를 통해 그래픽하게 스냅샷 모니터링 정보를 확인할 수 있습니다.

Point DB2 이벤트 모니터는 일정 기간동안 데이터베이스의 활동 내역에 대한 모니터링 정보를 수집하여 파일 또는 테이블에 저장하여 분석합니다.



이벤트 모니터

이벤트 유형	설명
데이터베이스	마지막 애플리케이션이 접속을 종료할 때, 데이터베이스에 관련된 모든 모니터링 항목을 기록합니다.
테이블	마지막 애플리케이션이 접속을 종료할 때, 변경이 발생했던 테이블에 대해 읽혀진 행의 수와 기록된 행의 수등을 기록합니다.
DEADLOCKS	교착 상태가 발생할 때마다 교착 상태를 유발한 잠금 모드와 관련된 애플리케이션의 정보를 기록합니다. WITH DETAILS 옵션으로 상세한 정보를 수집할 수 있습니다.
TABLESPACES	마지막 애플리케이션이 접속을 종료할 때, 각각의 테이블 스페이스에 대해 버퍼풀 모니터링 항목과 프리페처, 페이지 클리너, Direct I/O 횟수 등의 정보를 기록합니다.
BUFFERPOOLS	마지막 애플리케이션이 접속을 종료할 때, 각각의 버퍼풀에 대한 기본 모니터링 항목과 프리페처, 페이지 클리너, Direct I/O 횟수등의 정보를 기록합니다.
CONNECTIONS	애플리케이션이 접속을 종료할 때, 애플리케이션과 관련된 모든 모니터링 항목을 기록합니다.
TRANSACTIONS	COMMIT 또는 ROLLBACK 으로 트랜잭션이 종료될 때마다 UOW의 시작과 종료 시간, 이전 UOW의 종료 시간등의 정보를 기록합니다.
STATEMENTS	SQL문이 실행을 종료할 때마다 SQL문의 시작과 종료 시간, 사용한 CPU, Dynamic SQL문의 텍스트 등을 정보를 기록합니다.

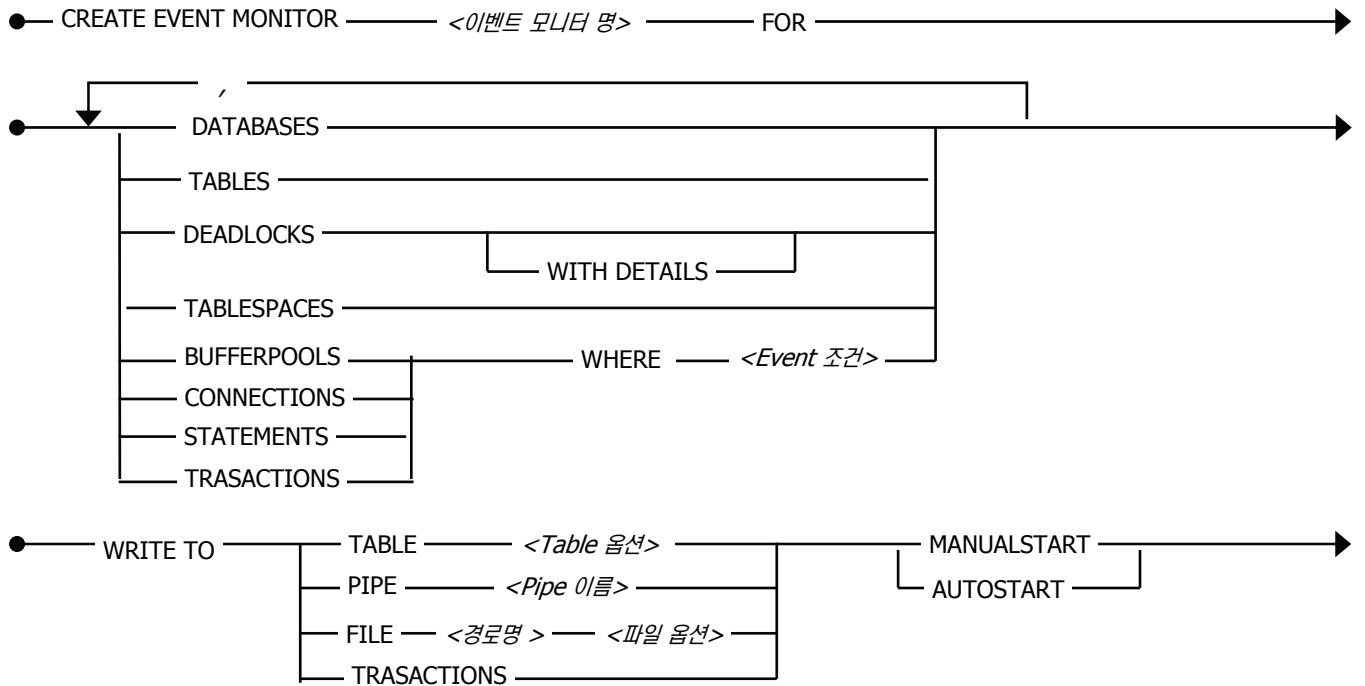
이벤트 모니터 명령어

명령어	설명
db2=> CREATE EVENT MONITOR	이벤트 모니터를 생성합니다.
db2=> DROP EVENT MONITOR	이벤트 모니터를 제거합니다. 그러나 상태값이 ON일 경우 제거할 수 없습니다.
db2=> SET EVENT MONITOR	원하는 시점에 이벤트 모니터의 상태값을 "ON/OFF" 로 설정하여 이벤트 모니터의 기간을 조절할 수 있습니다. 상태값이 "ON"인 경우 이벤트 모니터는 활성화되어 이벤트의 발생 정보를 수집합니다.
db2=> FLUSH EVENT MONITOR	이벤트 모니터의 값을 메모리에서 출력 I/O 로 강제적으로 기록합니다.

11 DB2 이벤트 모니터 구문

Point 모니터 스위치를 켜면 현재 시점의 데이터베이스의 활동 상황을 모니터링할 수 있습니다. 모니터링의 대상은 인스턴스, 데이터베이스, 애플리케이션이 되며 모니터 스위치 설정값에 따라 정보가 달라집니다.

CREATE EVENT MONITOR 구문



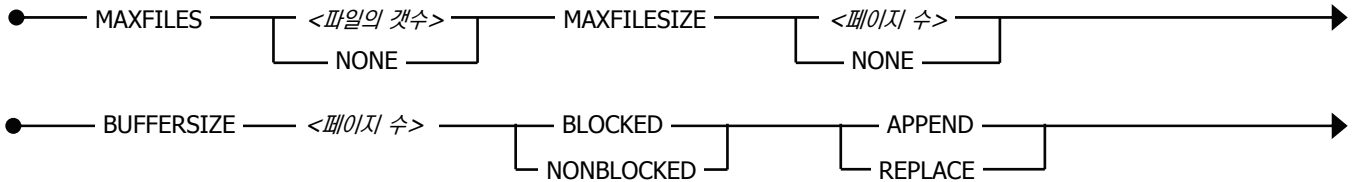
참고 : 이외에도 다양한 옵션이 제공되므로 도움말을 참고하시기 바랍니다.

옵션 설명 :

모드	설명
이벤트 모니터 명	임의의 고유한 값으로 지정합니다.
WHERE	CONNECTIONS, STATEMENTS, TRANSACTIONS 유형은 수집되는 정보의 양이 너무 많게 되므로 모니터링의 범위를 줄일 수 있는 조건을 지정합니다. APPL_ID, AUTH_ID, APPL_NAME 을 이용하여 조건식을 표현합니다.
WRITE TO TABLE	수집된 정보를 테이블에 저장합니다.
WRITE TO FILE	수집된 정보를 파일에 저장합니다.
WRITE TO PIPE	수집된 정보를 파이프에 저장합니다.
MANUALSTART	데이터베이스가 활성화되어도 자동적으로 시작되지 않습니다. SET EVENT MONITOR STATE문으로 조절합니다.
AUTOSTART	데이터베이스가 활성화되면 자동적으로 시작됩니다.

Point 이벤트 모니터가 수집한 정보를 파일에 저장합니다. 파일의 경우 생성할 파일 위치, 크기, 개수 버퍼의 크기등의 옵션을 추가적으로 지정할 수 있습니다. 생성된 파일은 db2evmon 명령어로 변환하여야 합니다.

파일 옵션



참고 : 이외에도 다양한 옵션이 제공되므로 도움말을 참고하시기 바랍니다.

옵션 설명 :

모드	설명
MAXFILES	이벤트 모니터 파일의 수를 지정합니다. NONE 일 경우 파일의 개수에 제한이 없습니다.
MAXFILESIZE	이벤트 모니터 파일의 크기를 지정합니다.
BUFFERSIZE	이벤트 모니터 버퍼의 크기를 지정합니다.
BLOCKED NONBLOCKED	BLOCKED 의 경우 이벤트 모니터가 디스크에 쓰여지는 것을 기다려 주며 NONBLOCKED 의 경우 기다려 주지 않기 때문에 데이터 유실이 발생할 수 있습니다.

파일 이벤트 모니터 절차

- 이벤트 모니터가 수집하는 정보를 저장할 경로를 생성합니다. 이벤트 모니터의 수집된 정보는 향후 지정된 경로에 .evt 파일에 저장됩니다.

```
c:\W>mkdir c:\WEM_REP
```

- CREATE EVENT MONITOR 명령어로 파일 이벤트 모니터를 생성합니다. 만약 출력 파일의 개수가 이 곳에서 지정한 MAXFILES 수를 초과하거나 또는 데이터베이스가 비활성화되면 이벤트 모니터는 자동적으로 중지됩니다.

```
db2=> CREATE EVENT MONITOR <이벤트 모니터 명> FOR <이벤트 유형명> WRITE TO FILE
<경로명> MAXFILES <최대 파일 개수> MANUALSTART
```

- 이벤트 모니터를 활성화시킵니다.

```
db2=> SET EVENT MONITOR <이벤트 모니터 명> STATE=1
```

- 일정한 시간동안 이벤트 정보를 수집한 후에 이벤트 모니터를 비활성화시킵니다.

```
db2=> SET EVENT MONITOR <이벤트 모니터 명> STATE=0
```

- db2evmon 명령어를 이용하여 파일로 수집된 이벤트 정보를 텍스트 형태로 변환합니다.

```
c:\WEM_REP>db2evmon -db <DB 명> -evm <이벤트 모니터 명> <출력 파일 명>
c:\WEM_REP>del *.evt
```

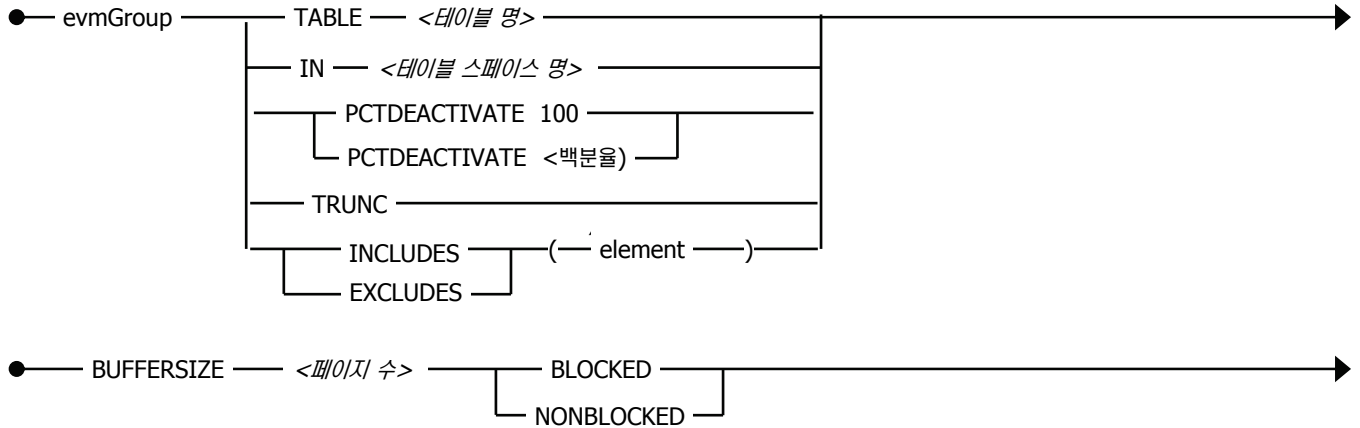
- 불필요한 이벤트 모니터는 제거합니다.

```
db2=> DROP EVENT MONITOR <이벤트 모니터 명>
```

13 DB2 테이블 이벤트 모니터

Point 이벤트 모니터가 수집한 정보를 테이블에 저장합니다. 수집된 데이터 스트림을 한 개 이상의 논리적인 그룹으로 분할하여 해당 테이블에 INSERT 합니다. SELECT 구문을 이용하여 분석합니다.

테이블 옵션



참고 : 이외에도 다양한 옵션이 지정되어 있으므로 도움말을 참고하시기 바랍니다.

옵션 설명 :

옵션	설명
evmGroup	evmGroup은 이벤트 유형에 따라 생성되는 데이터의 논리적인 그룹입니다. 하나의 이벤트 유형에서 발생하는 데이터는 여러 개의 논리적인 그룹으로 나뉘어 집니다. 예를 들면 이벤트 유형을 TABLESPACE로 할 경우 evmGroup은 TABLESPACE 와 CONTROL 두개의 논리적인 그룹으로 데이터가 나뉘어 집니다. 만약 TABLESPACE 에 대한 정보만을 필요로 할 경우 evmGroup에 TABLESPACE라는 논리적인 그룹을 지정하고 그 데이터를 저장할 테이블을 지정하면 됩니다. 아래 evmGroup표는 각 이벤트 유형에서 발생하는 논리적 데이터 그룹입니다.
TABLE <테이블 명>	논리적 그룹의 데이터가 저장될 테이블을 지정합니다.
PCTDEACTIVATE 정수값	테이블이 저장되는 테이블 스페이스가 DMS일 경우 테이블 스페이스가 용량이 얼마인지 체크하여 이 곳에서 지정한 백분율만큼 테이블 스페이스가 사용될 경우 이벤트 모니터는 자동으로 비활성화됩니다. 이 옵션은 SMS 테이블 스페이스일 경우에는 사용될 수 없습니다.
INCLUDE EXCLUDE	INCLUDE는 테이블에 기록할 Element를 지정합니다. EXCLUDE는 테이블에 기록하지 않을 Element를 지정합니다.

evmGroup :

이벤트 유형	설명
데이터베이스	DB, CONTROL
테이블	TABLE, CONTROL
DEADLOCKS	CONNHEADER, DEADLOCK, DLCONN, CONTROL
TABLESPACES	TABLESPACE, CONTROL
BUFFERPOOLS	BUFFERPOOL, CONTROL
CONNECTIONS	CONNHEADER, CONN, CONTROL
TRANSACTIONS	CONNHEADER, XACT, CONTROL
STATEMENTS	CONNHEADER, STMT, SUBSECTION, CONTROL

Point 이벤트 모니터가 수집한 정보를 테이블에 저장합니다. 수집된 데이터 스트림을 한 개 이상의 논리적인 그룹으로 분할하여 해당 테이블에 INSERT 합니다. SELECT 구문을 이용하여 분석합니다.

테이블 이벤트 모니터 절차

- 1 이벤트 모니터의 수집 데이터가 저장될 테이블들을 위한 테이블 스페이스를 생성합니다.

```
db2=> create tablespace EVTBS managed by automatic storage
```

- 2 이벤트 모니터를 생성합니다. 이때 evmGroup이 지정되지 않을 경우 이벤트 유형에 해당하는 모든 논리적인 그룹에 대한 테이블들이 자동으로 생성됩니다. 이 때 디폴트로 생성되는 테이블 이름은 evmGroup_<이벤트 모니터 이름> 이며 스키마는 접속한 사용자의 이름이 사용됩니다. 만약 특정 evmGroup 에 대해서만 기록할 경우 evmGroup 을 지정하고 테이블 명을 지정할 수 있습니다.

```
db2=> CREATE EVENT MONITOR T_EVMON FOR TABLES WRITE TO TABLE
      TABLE ( TABLE evmon.TABLE_T_EVMON, IN EVTBS, PCTDEACTIVATE 90 )
```

- 3 테이블 이벤트 모니터 생성 명령어에 대해 생소할 경우 db2evtbl 명령어를 이용하여 생성 구문에 대해 다음과 같이 도움을 받을 수 있습니다. 보여주는 구문을 편집하여 이벤트 모니터를 생성하도록 합니다.

```
C:\W>db2evtbl -schema evmon -evm amj tables > crevmon.ddl
C:\W>db2 -tvf crevmon.ddl
```

crevmon.ddl 결과 화면 :

```
CREATE EVENT MONITOR amj FOR TABLES WRITE TO TABLE
      TABLE (TABLE evmon.TABLE_amj
              INCLUDES (DATA_OBJECT_PAGES,
                        DATA_PARTITION_ID,
                        EVENT_TIME,
                        EVMON_ACTIVATES,
                        .....
                        XDA_OBJECT_PAGES ) ),
      CONTROL (TABLE evmon.CONTROL_amj
              INCLUDES (EVENT_MONITOR_NAME,
                        MESSAGE,
                        MESSAGE_TIME ) );
```

- 4 이벤트 모니터를 활성화시킵니다. 일정한 시간 동안 이벤트 정보를 수집한 후에 이벤트 모니터를 비활성화 시킵니다.

```
db2=> SET EVENT MONITOR T_EVMON STATE=1
db2=> SET EVENT MONITOR T_EVMON STATE=0
```

- 5 이벤트 모니터가 종료되면 불필요한 이벤트 모니터와 테이블을 제거합니다 .

```
db2=> DROP EVENT MONTIOR T_EVMON
db2=> DROP TABLE evmon.TABLE_T_EVMON
```

15 db2pd Tool

Point db2pd 도구는 DB2 메모리 세트의 빠르고 즉각적인 정보를 포함하는 문제점 판별 도구입니다. 이는 래치를 확보하거나 엔진 자원을 사용하지 않고 정보를 수집합니다.

db2pd Tool 구문


- db2pd [-inst | -help | -version]
- [-db <DB 명> | -alldatabases] [-file <파일 명>] [-everything] [-repeat <초> <횟수>]
- [-applications [<applHandle>]] [-apinfo]
- [-transactions [<transaction handle>]] [-bufferpools [<bpID>]]
- [-logs] [-locks [<transHdl>] | -wlocks]
- [-tablespaces [<테이블 스페이스 ID>]] [-dynamic | -static]
- [-tcbstats [-all | -index]] [-osinfo] [-activestatements]

참고 : 이외에도 다양한 옵션이 제공되오니 도움말을 참고하시기 바랍니다.

옵션 설명 :

옵션	설명
-inst	모든 인스턴스 레벨의 정보를 보여줍니다.
-h -help	온라인 도움말을 보여줍니다.
-v -version	설치된 DB2 제품의 현재 버전과 서비스 레벨을 보여줍니다.
-database -db <DB 명>	지정된 데이터베이스의 메모리 집합에 접근하도록 지정합니다.
-alldatabases -alldbws	모든 데이터베이스의 메모리 집합에 접근하도록 지정합니다.
-everything	모든 데이터베이스의 모든 옵션을 수행합니다.
-file <파일명>	결과값을 지정된 파일에 기록합니다.
-applications	애플리케이션에 대한 정보를 보여줍니다. 애플리케이션 ID가 지정될 경우 그 애플리케이션 정보만 보여줍니다.
-apinfo	현재 UOW의 Dynamic SQL 구문의 실행을 포함하여 애플리케이션에 대한 자세한 정보를 제공합니다.
-transactions	active 트랜잭션에 대한 정보를 보여줍니다.
-bufferpools	버퍼풀에 대한 정보를 보여줍니다. 만약 버퍼풀 ID가 지정되면 지정된 버퍼풀에 대한 정보만 보여줍니다.
-logs	로그 파일에 대한 정보를 보여줍니다.
-locks	LOCK에 대한 정보를 보여줍니다. 만약 showlocks 옵션을 함께 사용할 경우 LOCK이름에 대한 자세한 정보를 보여주며 wait 옵션과 함께 사용할 경우 wait 상태인 LOCK과 그 LOCK의 소유주에 대한 정보를 보여줍니다.
-wlocks	각 LOCK에 대한 소유주와 그 LOCK이 풀리기를 기다리는 자에 대한 정보를 보여줍니다. 결과화면에서 Sts 의 값이 'G'일 경우는 LOCK의 소유주를 표시하며 'W'는 LOCK이 풀리기를 기다리는 자를 의미합니다. 9.5 FP1 에 새로 추가된 기능입니다.
-tablespaces	테이블 스페이스에 대한 정보를 보여줍니다.
-dynamic -static	Dynamic SQL 또는 Static SQL 및 패키지의 실행 정보를 보여줍니다.
-tcbstats	테이블과 인덱스에 대한 정보를 보여줍니다.
-osinfo	OS 정보를 보여줍니다.
-activestatements	active satement 목록에 대한 정보를 보여줍니다.

16 DB2 활동 모니터 (Activity Monitor)

Point  활동 모니터를 사용하여 응용 프로그램 성능 및 동시성, 자원 소비 및 데이터베이스의 SQL문 사용을 모니터링 할 수 있습니다.

활동 모니터 개요

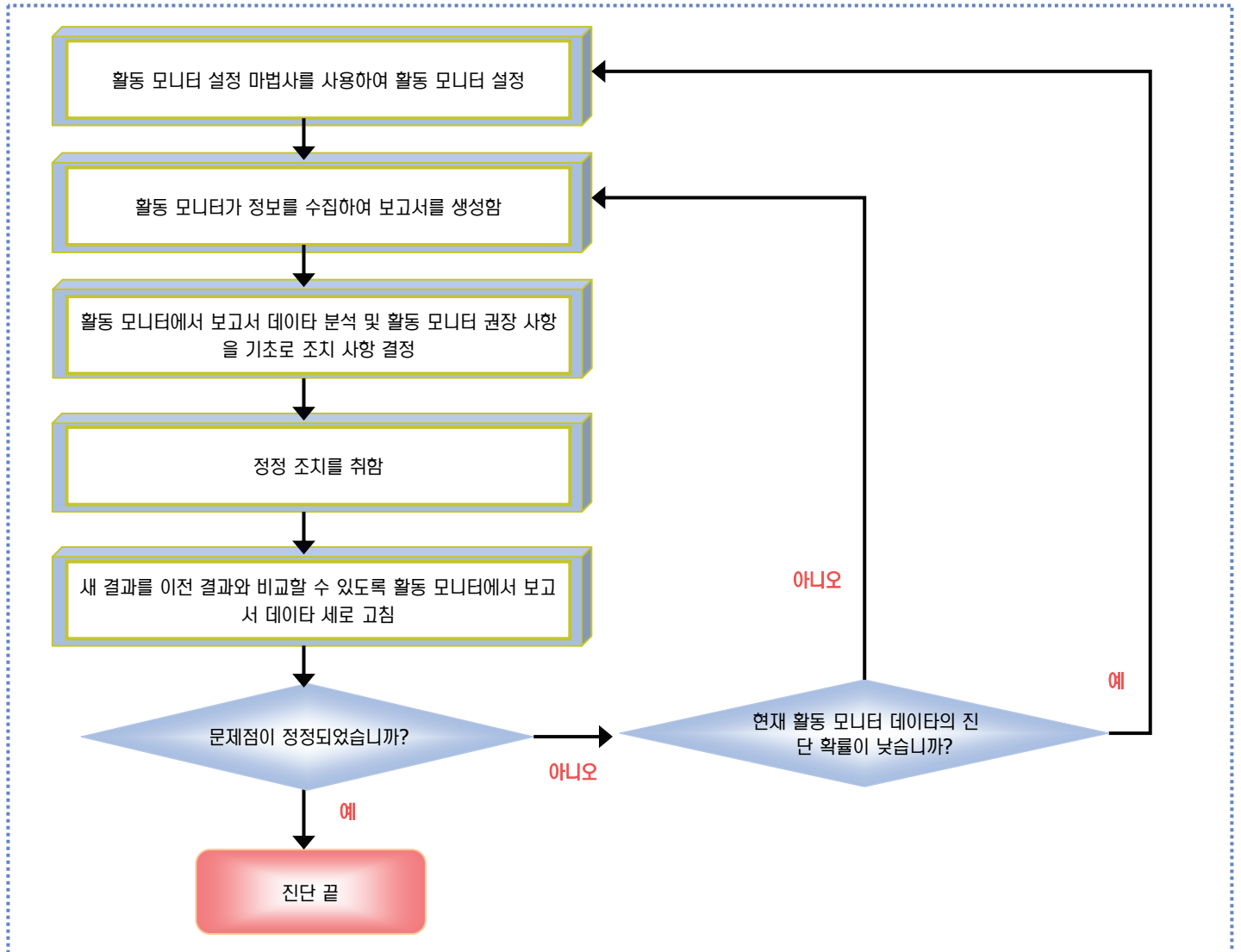
활동 모니터를 사용하여 애플리케이션 성능 및 동시성, 자원 소비 및 데이터베이스 또는 데이터베이스 파티션의 SQL문을 모니터링할 수 있습니다. 활동 모니터는 모니터 데이터의 특정 서브세트를 기본으로 사전 정의된 보고서 세트를 제공합니다. 이 보고서를 통해 응용프로그램 성능, 응용프로그램 동시성, 자원 소비 및 SQL문 사용에 대한 모니터링에 초점을 둘 수 있습니다. 활동 모니터는 보고서를 분석하여 그 보고서를 바탕으로 권장사항도 제공합니다. 이러한 권장사항을 통해 데이터베이스 성능 문제점의 원인을 진단하고 데이터베이스 자원 사용을 최적화하도록 쿼리를 조정할 수 있습니다.

활동 모니터에서 수행할 수 있는 태스크

태스크	태스크 유형	오출
트랜잭션	선택된 응용프로그램에서 실행되는 트랜잭션을 봅니다.	보고서 데이터 분할창에서 하나 이상의 응용프로그램을 선택한 후 오른쪽 단추를 누른 다음 최신 트랜잭션 표시를 선택합니다.
명령문	선택된 응용프로그램에서 실행 중인 SQL문을 봅니다.	보고서 데이터 분할창에서 하나 이상의 응용프로그램을 선택한 후 오른쪽 단추를 누른 다음 최신 명령문 표시를 선택합니다.
	선택된 응용프로그램에서 실행 중인 SQL문의 텍스트를 봅니다.	응용프로그램 명령문 창에서 보고서 데이터 분할창에 있는 명령문을 마우스 오른쪽 단추로 누른 후 명령문 텍스트 표시를 선택합니다.
애플리케이션 잠금 체인	현재 선택된 응용프로그램에 영향을 미치는 잠금 또는 잠금 대기 중 상황을 봅니다.	보고서 데이터 분할창에서 응용프로그램을 선택하십시오. 마우스 오른쪽 단추를 누른 다음 잠금 체인 표시를 선택합니다.
	잠금 정보를 볼 선택된 응용프로그램에 대한 정보를 봅니다.	응용프로그램 잠금 체인 창에서, 응용프로그램을 마우스 오른쪽 단추로 누른 다음 정보를 선택합니다.
	데이터베이스에서 선택된 응용프로그램이 보유하거나 대기 중인 잠금에 대한 정보를 봅니다.	응용프로그램 잠금 체인 창에서, 응용프로그램을 마우스 오른쪽 단추로 누른 다음 잠금 세부사항 표시를 선택합니다.
보고서 데이터 및 권장사항 보기	보고서 데이터를 해석할 수 있는 정보를 봅니다.	활동 모니터 창, 응용프로그램 명령문 창 또는 응용프로그램 트랜잭션 창에서 보고서 화살표를 사용하여 보고서를 선택한 후 보고서 세부사항 누름 단추를 누른 후 세부사항 페이지를 확인합니다.
	활동 모니터가 제공하는 권장사항을 봅니다.	활동 모니터 창, 응용프로그램 명령문 창 또는 응용프로그램 트랜잭션 창에서 보고서 화살표를 사용하여 보고서를 선택한 후 보고서 세부사항 누름 단추를 누른 후 권장 사항 페이지를 확인합니다.

17 DB2 활동 모니터 (Activity Monitor)


Point 다음은 활동 모니터의 개요와 활동 모니터의 화면입니다.



활동 모니터 화면

보고서 데이터

응용프로그램 핸들(에이전트 ID)	응용프로그램 이름	권한 부여 ID	응용프로그램 ID	총 CPU 시간	사용자 C
18	acmerpt.exe	EDWARDL	*LOCAL.DB2.00...	180259	
20	db2cc.exe	DB2ADMIN	*LOCAL.DB2.00...	30042	
22	acmefin.exe	FREDS	*LOCAL.DB2.00...	20028	
21	db2evm.exe	DB2ADMIN	*LOCAL.DB2.00...	20028	
27	acmeacct.exe	ALICET	*LOCAL.DB2.00...	10015	

Point  다음은 스냅샷 모니터를 이용하여 시스템 모니터를 하는 예제입니다.

1. SAMPLE 데이터베이스에 접속합니다.

```
c:\WLAB\WUNIT13> db2 connect to sample
```

2. UPDATE MONITOR SWITCHES 명령어로 LOCK 스위치를 켜서 잠금 대기에 소비된 시간을 수집합니다.

```
c:\WLAB\WUNIT13>db2 update monitor switches using LOCK on
```

```
DB20000I UPDATE MONITOR SWITCHES 명령이 완료되었습니다.
```

3. 데이터베이스 카탈로그에 잠금을 필요로 하는 명령 또는 명령문을 발생합니다. 이 곳에서는 커서 선언, 열기, 폐지를 수행합니다.

```
c:\WLAB\WUNIT13>db2 -c- declare c1 cursor for
      SELECT * from staff where job='Sales' for update
c:\WLAB\WUNIT13>db2 -c- open c1
c:\WLAB\WUNIT13>db2 -c- fetch c1
```

4. snapshot 명령어를 이용하여 Lock 정보를 얻도록 합니다.

```
c:\WLAB13>db2 get snapshot for locks on sample
```


데이터베이스 잠금 스냅샷

```

데이터베이스 이름      = SAMPLE
데이터베이스 경로      = C:\WDB2\WNODE0000\WSQL00003W
입력 데이터베이스 별명  = SAMPLE
보유된 잠금 수          = 4
현재 연결된 응용프로그램 수 = 1
현재 잠금 대기 중인 에이전트 수 = 0
스냅샷 시간소인        = 2008-05-18 13:06:54.179537
응용프로그램 핸들      = 12
응용프로그램 ID        = *LOCAL.DB2.080518040441
시퀀스 번호              = 00002
응용프로그램 이름      = db2bp.exe
CONNECT 권한 부여 ID    = KR010389
응용프로그램 상태      = UOW 대기 중
상태 변경 시간          = 수집되지 않음
응용프로그램 코드 페이지 = 1363
보유된 잠금 수          = 4
전체 대기 시간(밀리초)  = 0
잠금 목록
잠금 이름              = 0x02000F000500000600000000052
잠금 속성              = 0x00000000
릴리스 플래그         = 0x00000001
잠금 계수              = 1
보유 계수              = 0
잠금 오브젝트 이름     = 100663301
오브젝트 유형          = 행
테이블 스페이스 이름   = USERSPACE1
테이블 스키마          = KR010389
테이블 이름            = STAFF
모드                    = U

잠금 이름              = 0x02000F000000000000000000054
잠금 속성              = 0x00000000
릴리스 플래그         = 0x00000001
잠금 계수              = 1
보유 계수              = 0
잠금 오브젝트 이름     = 15
오브젝트 유형          = 테이블
테이블 스페이스 이름   = USERSPACE1
테이블 스키마          = KR010389
테이블 이름            = STAFF
모드                    = IX

```

Point  다음은 교착 상태 모니터링 시나리오의 예로 이벤트 모니터링의 특성을 예시합니다. 이 시나리오를 수행하려면 3개의 DB2 CLP 창을 열어 수행합니다. 첫번째 CLP 를 모니터 세션으로, 나머지 CLP를 CMD1과 CMD2라 합니다.

1. CLP 창을 하나 띄운 후에 모니터 세션으로 사용합니다. 이 창에서는 SAMPLE 데이터베이스에 연결하여 데이터베이스로 연결하는 사이에 발생하는 교착 상태와 테이블 데이터를 로그하는 이벤트 모니터를 정의합니다.

```
c:\WLABWUNIT13> db2 connect to sample
c:\WLABWUNIT13> mkdir dlmon
c:\WLABWUNIT13> notepad crevent.ddl
```

```
CREATE EVENT MONITOR DLMON
FOR TABLES, DEADLOCKS WITH DETAILS
WRITE TO FILE 'C:\WLABWUNIT13\WDLMON';
SET EVENT MONITOR DLMON STATE 1;
```

```
c:\WLABWUNIT13> db2 -tvf crevent.ddl
```

2. CMD 창을 두개 띄워 교착상태에 빠지도록 합니다. 교착 상태는 각 애플리케이션이 처리를 계속하기 위해 다른 애플리케이션이 필요로 하는 잠금을 보유하고 있는 상황입니다. 이 때 교착상태는 DB2 교착 상태 검출기에 의해 발견되어 트랜잭션 중 하나를 롤백하여 해결합니다. 따라서 애플리케이션 중 하나만이 트랜잭션을 성공적으로 완료하게 됩니다. 아래와 같이 CMD 1은 STAFF 테이블의 한 행에서 exclusive lock을 보유하고 CMD 2는 DEPARTMENT 테이블의 한 행에서 exclusive lock을 보유하도록 합니다. +c 옵션은 자동으로 Commit 하는 것을 해제합니다.

CMD 1

```
c:\WLABWUNIT13> db2 connect to sample
c:\WLABWUNIT13> db2 +c "insert into staff values
(26, 'Simpson', 2, 'Mgr', 13, 35000, 0)"
```

CMD 2

```
c:\WLABWUNIT13> db2 connect to sample
c:\WLABWUNIT13> db2 +c "insert into department values
('7G','Safety',NULL, 'A00', NULL)"
```

3. CMD 1 창에서 DEPARTMENT 테이블을 조회합니다. 이 때 CMD1 은 행이 폐지될 때 DEPARTMENT 테이블의 행이 필요하지만 CMD2가 배타적 잠금을 가지고 있기 때문에 Shared Lock 을 획득할 수 없습니다. CMD1은 잠금이 풀리기를 기다려야 하므로 LOCK WAIT 상태에 들어갑니다.

CMD 1 - 응용 프로그램 1

```
c:\WLABWUNIT13> db2 +c "select deptname from
department"
```

4. 응용 프로그램 2에서 Staff 를 조회합니다. 이때 응용 프로그램 2도 응용 프로그램 1이 Staff의 마지막 행에서 Exclusive Lock을 보유하고 있으므로 Shared Lock 을 획득할 수 없습니다. 응용 프로그램 2도 역시 LOCK WAIT 에 들어가게 되어 교착상태에 빠지게 됩니다.

CMD 1 - 응용 프로그램 1

```
c:\WLABWUNIT13> db2 +c "select deptname from
department"
```

5. 교착 상태 검출기가 교착 상태를 점검하고 롤백할 희생(Victim)을 선택하게 됩니다. 두개의 창 중 하나의 창에서 다음과 같이 검출기에 의해 강제 종료되었다라는 메시지를 확인할 수 있으며 다른 한 창에서는 조회를 성공적으로 수행하는 것을 확인할 수 있습니다.

```
SQL0911N 현재 트랜잭션이 교착 상태 또는 시간종료로 인해 롤백되었습니다. 이유코드 "2". SQLSTATE=40001
```


6. CMD1 과 CMD2 의 연결을 종료합니다.

CMD 1

```
c:\WLABWUNIT13> db2 connect reset
```

CMD 2

```
c:\WLABWUNIT13> db2 connect reset
```

Point  다음은 교착 상태 모니터링 시나리오의 예로 이벤트 모니터링의 특성을 예시합니다. 이 시나리오를 수행하려면 3개의 DB2 CLP 창을 열어 수행합니다. 첫번째 CLP 를 모니터 세션으로, 나머지 CLP를 CMD1 과 CMD2 라 합니다.

7. 이벤트 모니터창으로 돌아가서 현재 이벤트 버퍼의 내용을 파일로 강제 기록합니다. 이벤트 모니터의 경우 우선적으로 버퍼에 기록하였다가 버퍼가 채워지면 파일에 기록하게 됩니다. 현재 수행한 실습은 이벤트 버퍼를 채울 수 있을 만큼의 모니터 데이터가 존재하지 않으므로 "FLUSH EVENT MONITOR" 명령으로 강제 기록합니다.

```
C:\LAB\Unit13>db2 flush event monitor dlmon
```

8. 바이너리 파일로 기록된 이벤트 모니터 파일을 db2evmon 명령어로 format 하여 확인합니다. 앞에서 이벤트 파일 모니터의 위치를 확인하여 다음 명령어를 수행합니다.

```
C:\LAB\Unit13>db2 connect reset
C:\LAB\Unit13>db2evmon -path DLMON > out
```

9. 이벤트 모니터가 교착상태에 대해 Capture 한 것을 확인할 수 있습니다.

```
C:\LAB\Unit13>notepad out
```

EVENT LOG HEADER

```
Event Monitor name: DLMON
Server Product ID: SQL09050
Version of event monitor data: 9
Byte order: LITTLE ENDIAN
Number of nodes in db2 instance: 1
Codepage of database: 1208
Territory code of database: 82
Server instance name: DB2
Database Name: SAMPLE
Database Path: C:\WDB2\WNODE0000\WSQL00003W
First connection timestamp: 2008-05-19 07:23:13.299688
Event Monitor Start time: 2008-05-19 07:51:53.852560
```

계속 :

5) Table Event ...


```
Table schema: <105><KR010389>
Table name: TEMP (00001,00002)
Data partition id: 0
Record is the result of a flush: FALSE
Table type: Temporary
Data object pages: 1
Index object pages: 0
Lob object pages: 0
Long object pages: 0
Rows read: 1
Rows written: 1
Overflow Accesses: 0
Page reorgs: 0
Tablespace id: 1
Table event timestamp: 2008-05-19 07:52:12.528283
```

6) Table Event ...

```
Table schema: <105><KR010389>
Table name: TEMP (00001,00002)
Data partition id: 0

Record is the result of a flush: FALSE
Table type: Temporary
Data object pages: 1
Index object pages: 0
Lob object pages: 0
Long object pages: 0
Rows read: 5
Rows written: 1
Overflow Accesses: 0
Page reorgs: 0
Tablespace id: 1
Table event timestamp: 2008-05-19 07:53:04.974464
```


21 DB2 이벤트 모니터 예제

Point  다음은 교착 상태 모니터링 시나리오의 예로 이벤트 모니터링의 특성을 예시합니다. 이 시나리오를 수행하려면 3개의 DB2 CLP 창을 열어 수행합니다. 첫번째 CLP 를 모니터 세션으로, 나머지 CLP를 CMD1 과 CMD2 라 합니다.

계속 :

```

7) Table Event ...
Table schema: <105><KR010389>
Table name: TEMP (00001,00003)
Data partition id: 0
Record is the result of a flush: FALSE
Table type: Temporary
Data object pages: 1
Index object pages: 0
Lob object pages: 0
Long object pages: 0
Rows read: 8
Rows written: 1
Overflow Accesses: 0
Page reorgs: 0
Tablespace id: 1
Table event timestamp: 2008-05-19 07:53:04.974500

8) Table Event ...
Table schema: <105><KR010389>
Table name: TEMP (00001,00002)
Data partition id: 0

Record is the result of a flush: FALSE
Table type: Temporary
Data object pages: 1
Index object pages: 0
Lob object pages: 0
Long object pages: 0
Rows read: 1
Rows written: 1
Overflow Accesses: 0
Page reorgs: 0
Tablespace id: 1
Table event timestamp: 2008-05-19 07:53:21.403712

9) Deadlock Event ...
Deadlock ID: 2
Number of applications deadlocked: 2
Deadlock detection time: 2008-05-19 07:53:39.325449
Rolled back Appl participant no: 2
Rolled back Appl Id: *LOCAL.DB2.080518225210
Rolled back Appl seq number: : 0005

```

계속 :


```

11) Deadlocked Connection ...
Deadlock ID: 2
Participant no.: 2
Participant no. holding the lock: 1
Appl Id: *LOCAL.DB2.080518225210
Appl Seq number: 00005
Appl Id of connection holding the lock:
*LOCAL.DB2.080518225202
Seq. no. of connection holding the lock: 00001
Lock wait start time: 2008-05-19 07:53:34.167278
Lock Name      : 0x02000F002900000600000000052
Lock Attributes: 0x00000000
Release Flags  : 0x00000001
Lock Count     : 1
Hold Count     : 0
Current Mode   : none
Deadlock detection time: 2008-05-19 07:53:39.325522
Table of lock waited on      : STAFF
Schema of lock waited on     : KR010389
Data partition id for table   : 0
Tablespace of lock waited on : USERSPACE1
Type of lock: Row
Mode of lock: X - Exclusive
Mode application requested on lock: NS - Share (and
Next Key Share)
Node lock occurred on: 0
Lock object name: 100663337
Application Handle: 105
Deadlocked Statement:
Type      : Dynamic
Operation: Fetch
Section   : 201
Creator   : NULLID
Package   : SQLC2G13
Cursor    : SQLCUR201
Cursor was blocking: FALSE
Txt       : select * from staff

List of Locks:
Lock Name      : 0x020000000100000001006B0056
Lock Attributes: 0x00000000
Release Flags  : 0x40000000

```

22 db2pd 사용 예제

Point  db2pd 도구는 DB2 메모리 세트의 빠르고 즉각적인 정보를 포함하는 문제점 판별 도구입니다. 이 곳에서는 db2pd 툴을 사용하여 현재 메모리 정보, 애플리케이션 접속 정보등을 확인하도록 합니다.

1. CLP 창을 하나 띄운 후에 다음과 같이 Sample 데이터베이스에 접속한 애플리케이션 정보와 동적 SQL문의 명령문 고유 ID를 확인할 수 있습니다.

```
c:\WLABWUNIT13> db2pd -db sample -app -dyn
```

```
Applications: Address AppHandl [nod-index] NumAgents
CoorPid Status 0x00000002006D2120 780 [000-00780] 1
10615 UOW-Executing C-AnchID C-StmtUID L-AnchID L-
StmtUID Appid 163 1 110 1
*LOCAL.jmcmahon.050202200412 Dynamic SQL
Statements: Address AnchID StmtUID NumEnv NumVar
NumRef NumExe Text 0x0000000220A02760 163 1 2 2 2 1
CREATE VIEW MYVIEW 0x0000000220A0B460 110 1 2 2 2
1 CREATE VIEW YOURVIEW
```

2. 다음 옵션을 사용할 경우 메모리 사용 현황에 대해 모니터링할 수 있습니다.

```
c:\WLABWUNIT13> db2pd -db sample -memblock
```

All memory blocks in Database SAMPLE set.

Address	PoolID	PoolName	BlkAge	Size(Bytes)	I	LOC	File
0x7E9D0218	5	utilh	3	184	1	223	780686123
0x7E9D02E8	5	utilh	4	228	1	64	1199050825
0x7E9D03E8	5	utilh	5	184	1	223	1780686123
0x7E9D04B8	5	utilh	6	228	1	64	1199050825
0x7E9D05B8	5	utilh	7	184	1	223	1780686123

3. 다음 옵션을 사용할 경우 테이블에서 삽입 횟수를 식별할 수 있습니다.

```
c:\WLABWUNIT13> db2pd -db sample -tcbstats
```

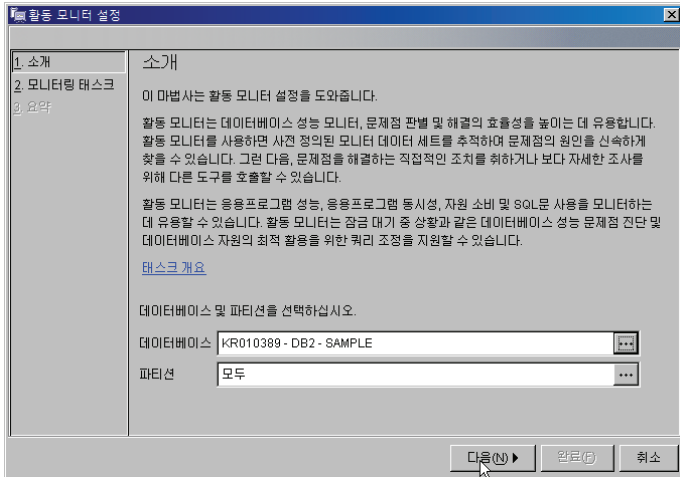
```
Database Partition 0 -- Database SAMPLE -- Active -- Up 0
days 01:07:16

TCB Table Information:
Address  TbspaceID TableID PartID MasterTbs MasterTab
TableName Schem
aNm ObjClass DataSize LfSize LobSize XMLSize
0x7EC2D220 0 1 n/a 0 1 SYSBOOT
SYSIB
M Perm 1 0 0 0
0x7EA17B20 2 -1 n/a 2 -1 INTERNAL
SYSIB
M Perm 1 0 0 0
0x7E9FD620 0 5 n/a 0 5
SYSTABLES SYSIB
M Perm 27 0 416 0
0x7EA102A0 0 6 n/a 0 6
SYSCOLUMNS SYSIB
M Perm 146 0 0 0
0x7EA1EFA0 0 7 n/a 0 7
SYSINDEXES SYSIB
M Perm 17 0 1 0
0x7EA12D20 0 8 n/a 0 8
SYSCOLPROPERTIES SYSIB
M Perm 1 0 0 0
```

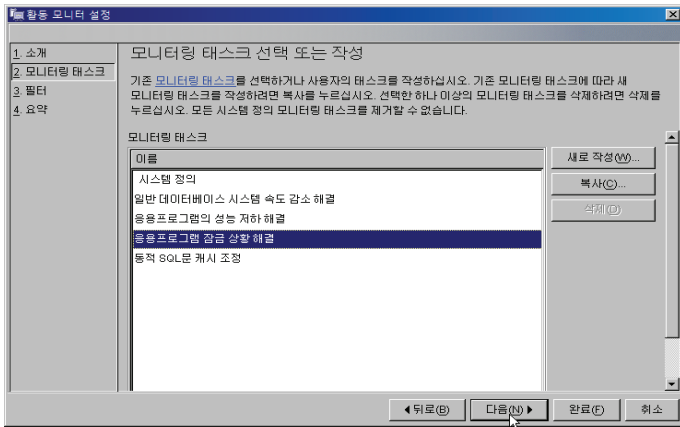
23 DB2 활동 모니터 예제

Point 다음은 활동 모니터를 통해 잠금 정보를 그래픽으로 확인할 수 있는 예제입니다.

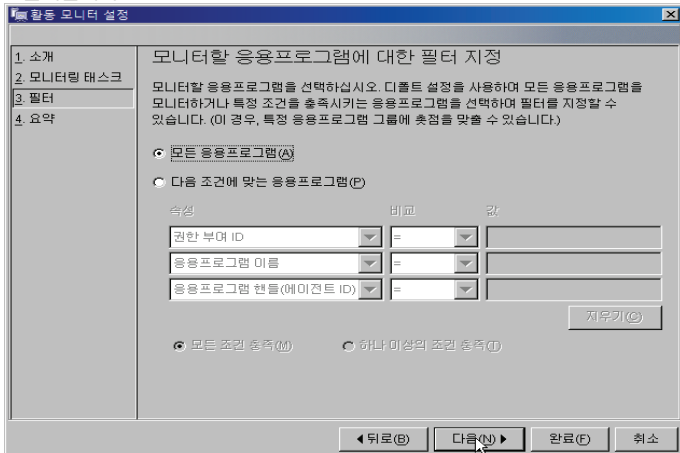
1. 시작 => 모든 프로그램 => IBM DB2 => DB2 COPY 1=> 모니터링 도구 => 활동 모니터를 선택한 후 활동 모니터 창이 뜨면 데이터베이스 선택란에 SAMPLE 데이터베이스를 선택하고 파티션은 모두를 선택합니다.



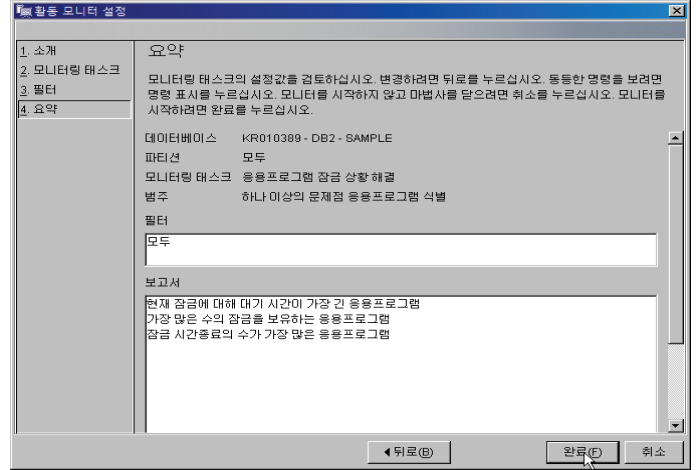
2. 모니터링 태스크 중 미리 정의된 "모니터링 잠금 상황 해결"을 선택하고 다음 버튼을 클릭합니다.



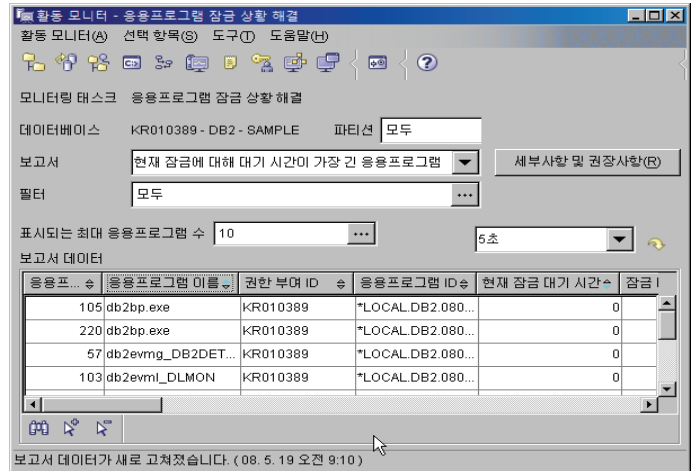
3. 모니터할 응용 프로그램을 "모든 응용 프로그램"으로 선택하고 다음 버튼을 클릭합니다.



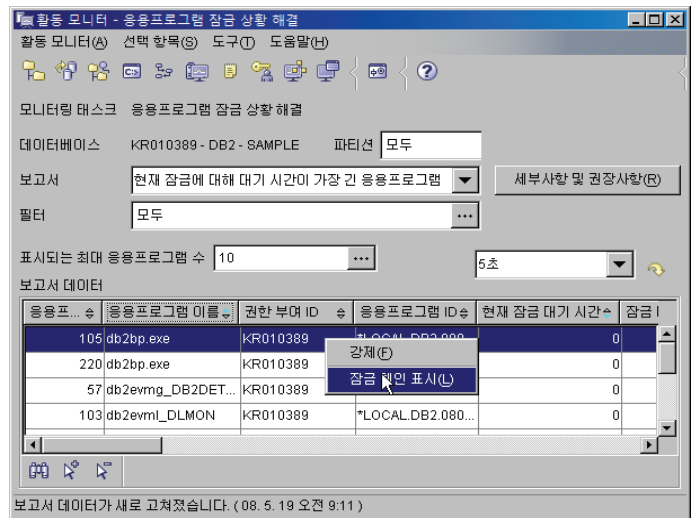
4. 이제까지 선택한 구성값을 확인한 후 완료 버튼을 클릭합니다



5. 보고서와 보고서를 주기적으로 갱신할 시간을 지정합니다.



6. 만약 응용 프로그램에서 Lock 체인이 발생할 경우 해당 응용 프로그램을 선택하고 Lock 체인 정보를 확인할 수도 있습니다.



@ 모니터링 Q/A

- 오라클은 OWI (Oracle Wait Interface)라는 Resource에 대해 경합이 이루어질때 Wait Event 를 발생시켜 관리합니다. DB2는 이러한 개념이 없는지요? 그러면 Resource 경합이 이루어지는 것에 대해 어떻게 알아볼 수 있는지요?

DB2는 OWI 라는 개념은 없습니다 .그러나 모든 RDBMS가 그렇듯이 Resource 에 대한 경합이 이루어지며 이러한 상황에 대한 정보가 DBA는 필요합니다. DB2는 이러한 정보를 오라클처럼 그룹화하여 관리하지는 않으며 스냅샷 모니터와 이벤트 모니터등을 통해 정보를 얻을 수 있습니다. 예를 들면 Lock Wait 의 경우 스냅샷 모니터를 통해 정보를 얻을 수 있습니다.

- DB2의 모니터링의 경우 모니터 스위치를 켜야 시스템에 대한 모니터링 정보를 얻을 수 있는 것 같은데 그러면 항상 모니터링 정보를 얻기 위해서는 항상 모니터 스위치를 켜야되는 건지요? 오라클처럼 메모리에 존재하는 정보를 얻을 수는 없는지요?

DB2의 경우 db2pd 툴을 이용할 경우에 메모리에 존재하는 정보를 얻을 수 있습니다. 또한 보다 자세한 정보를 얻기 위해서는 모니터 스위치를 켜고 스냅샷 모니터를 수행하여 정보를 얻을 수 있으며 이외에도 이벤트 모니터를 통해서 특정 이벤트 발생시 그 이벤트에 대한 정보를 얻을 수 있습니다.