

데이터베이스 개념과 오라클 설치




배재대학교 컴퓨터공학과
김 창 수

Contents

 데이터베이스 등장배경

 데이터베이스 발전과정

 데이터베이스 정의 및 특징

 관계형 데이터 모델

 오라클 설치

데이터베이스의 등장 배경

❖ 데이터베이스의 등장 배경

- 지식 기반 사회에서 대용량 데이터 관리에 대한 필요성
- 파일 시스템의 문제점
 - 응용 프로그램과 데이터의 종속성
 - 다수 사용자들의 정보 공유에 대한 문제
- 1970년대 데이터베이스 관리 시스템 등장
 - 컴퓨터에 저장된 대량의 데이터를 체계적으로 관리하고 사용자의 원하는 정보를 효과적으로 검색하기 위한 소프트웨어
 - 데이터베이스
 - 데이터베이스 관리시스템에 의해 관리되는 데이터의 집합
 - 데이터베이스 언어
 - 데이터베이스에 저장된 데이터를 검색하고 조작하기 위한 언어
 - SQL 언어
 - 현재 가장 널리 사용되는 데이터베이스 언어
 - 미국표준연구소(ANSI)와 국제표준기구(ISO)에서 관계형 데이터베이스 표준 언어로 채택

데이터베이스 발전 과정

❖ 데이터베이스 시스템의 발전 과정

1960년대 초반	최초의 데이터베이스 IDS(Integrated Data Store)제시
1960년대 후반	IBM사에서 IMS(Information Management System)개발
1970년 5월	IBM연구소의 Codd박사에 의해 관계형 데이터 모델 제시
1970년대 초반	System/R, 최초의 관계형 데이터베이스 시스템 모델
1979년	오라클 V1, 최초의 사용 관계형 데이터베이스 시스템
1982년	SQL/DS, IBM의 상용 관계형 데이터베이스 제품
현재	오라클, DB2, Sybase, SQL Server 사용 중

데이터베이스 발전 과정

❖ SQL의 역사

1973	SQUARE(Structured Queries As Relational Express)
1974	System/R용 SEQUEL(Structured English QUery Language)
1976	SEQUEL-2
1980	SQL(Structured Query Language)로 개명
1986	SQL-86, 최초의 SQL 표준안
1988	ANSI, ISO 국제표준 인정
1988	SQL-1(SQL/89) 표준안 제정
1992	SQL-2(SQL/92) 표준안 제정
1999	SQL-3(SQL/99) 표준안 제정

데이터베이스 발전 과정

❖ 현재의 데이터베이스

- 관계형 데이터베이스 시스템
 - 1990년대부터 모든 산업 분야에 널리 보급
 - 1992년 오라클 7 발표 이후, 우리나라에서도 활성화
- 객체지향 및 객체관계형 데이터베이스 시스템 등장
 - PC보급의 일반화에 따른 객체지향 기술과 멀티미디어 기능 요구
 - 관계형 데이터 모델 자체의 제약으로 인한 멀티미디어 기능 지원에 대한 문제점

데이터베이스 발전 과정

❖ 현재의 데이터베이스

- 객체지향 데이터베이스 시스템
 - 객체, 클래스, 메소드, 상속등과 같은 객체 개념을 지원
 - 모델링 능력이 우수하지만 복잡한 개념, 처리 성능의 저하로 인해 일부 특수 분야에서만 사용
 - O2, Versant, Ontos, ObjectStore 등
- 객체관계형 데이터베이스 시스템
 - 기존 관계형 데이터 모델과 객체지향 기술을 접목
 - IBM : DB2 UDB 5.0
 - INFORMIX : 1990년대 중반부터 지원
 - ORACLE :
 - ORACLE 8(1997년 발표) 부터 제한적으로 지원
 - ORACLE 9i부터 SQL-3에서 요구하는 대부분의 표준 기능 지원

데이터베이스의 정의

❖ 통합 데이터(integrated data)

- 여러 군데 분산된 데이터를 통합하여 데이터의 중복을 최소화하기 위한 개념

❖ 공유 데이터(shared data)

- 분산된 파일의 통합 관리를 통해 조직 내의 다수의 사용자나 여러 응용 시스템에서 동일한 데이터를 공유하는 개념

❖ 저장 데이터(stored data)

- 컴퓨터가 접근 가능한 매체에 저장된 디지털 데이터를 의미

❖ 운영 데이터(operational data)

- 특정 조직의 고유 업무를 수행하기 위하여 항상 유지해야 하는 데이터

❖ 의사 결정 데이터(decision-making data)

- 조직에서 장기간 축적된 데이터를 분석하여 새로운 의사 결정을 하는데 활용하기 위한 데이터

데이터베이스의 특징

❖ 실시간 접근(realtime access)

- 데이터베이스를 조회하는 사용자 질의에 대한 처리 시간이 몇 초를 넘지 않아야 한다는 것을 의미

❖ 지속적인 변화(continuous changes)

- 데이터베이스에도 추가, 삭제, 수정과 같은 작업을 통해 현실 세계의 변화를 지속적으로 반영

❖ 동시 공유(concurrent sharing)

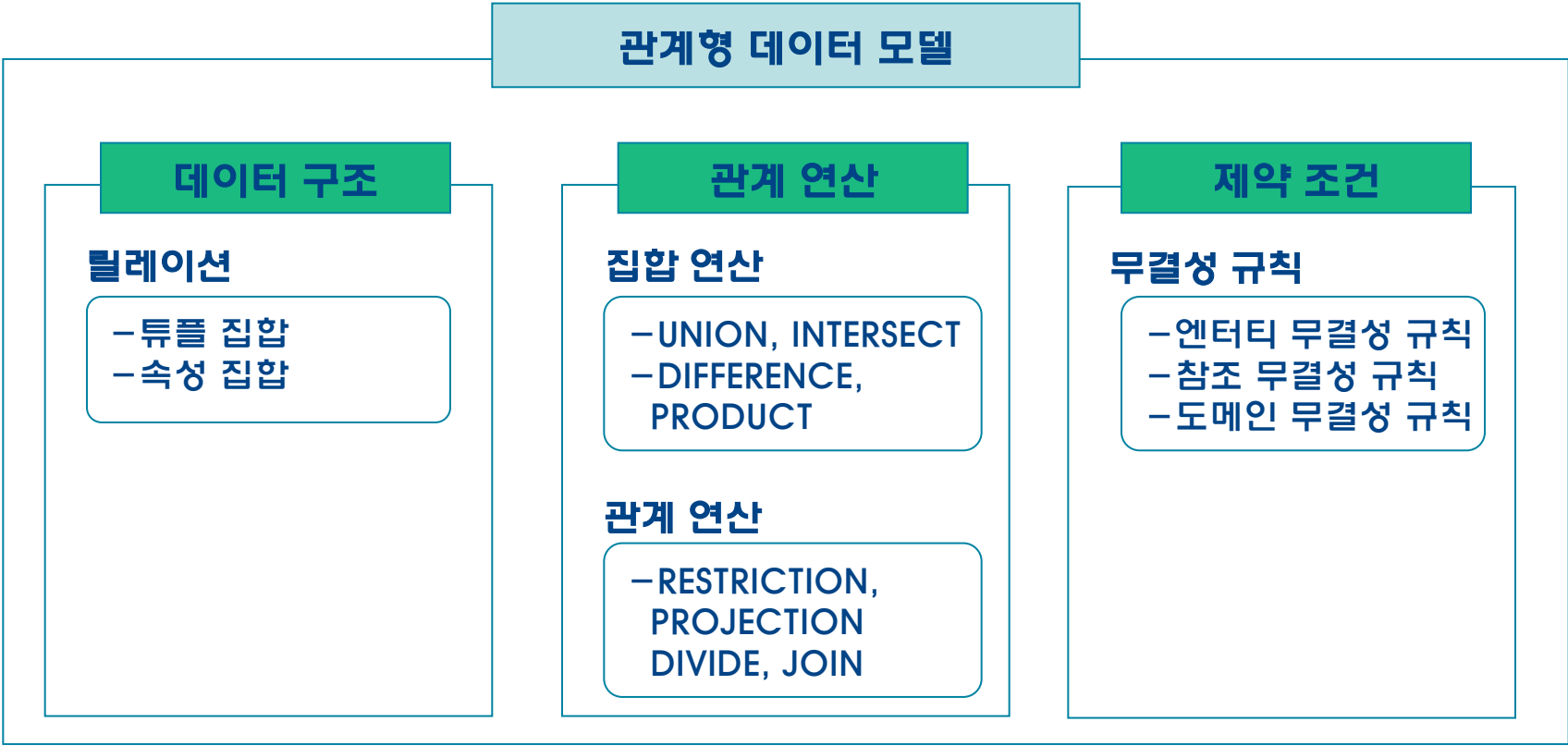
- 서로 다른 응용 업무에서 다수의 사용자가 동일한 데이터를 공유할 수 있는 환경을 제공

❖ 내용에 의한 참조(reference by content)

- 데이터베이스에서 데이터를 검색하는 방식은 데이터의 일부 내용에 의해 참조되는 방식
- 예 : '모든 컴퓨터공학과 학생을 검색하라'

관계형 데이터 모델

❖ 관계 데이터 모델의 구성 요소



관계형 데이터 모델

❖ 릴레이션(relation)개념

- 2차원 테이블 구조
- 튜플(tuple)과 속성(attribute)의 집합으로 구성

기본키 ← STUDENT 릴레이션 속성명 →

STUDNO	NAME	USERID	GRADE	HEIGHT	WEIGHT	DEPTNO	PROFNO
10101	전인하	ijun123	4	176	72	101	9903
20101	이동훈	Dals	1	172	64	201	
.....
10203	윤진옥	Samba7	3	171	70	102	9905
10107	이광훈	huriky	4	175	92	101	9903
.....

← 릴레이션 스키마

튜플 →

속성

인스턴스

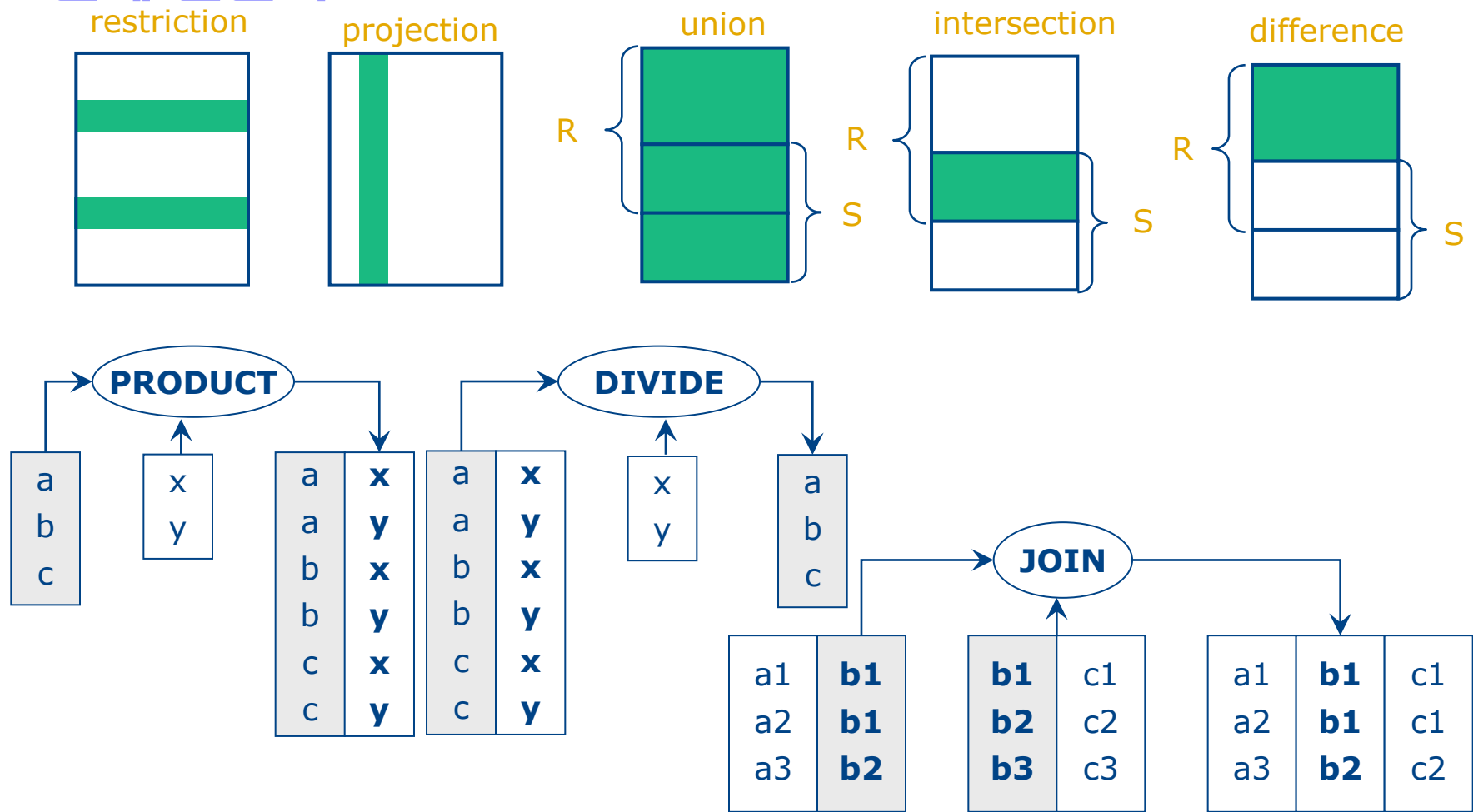
관계형 데이터 모델

❖ 릴레이션 성질

- 릴레이션에 저장된 튜플들은 유일
 - 릴레이션은 튜플들의 집합이기 때문에 하나의 릴레이션에는 동일한 튜플이 저장될 수 없음
- 릴레이션에 저장된 튜플간에는 순서 관계가 없음
 - 하나의 집합내에서 원소간의 순서는 무의미하기 때문에 릴레이션에 저장된 튜플끼리는 순서 관계가 없다
- 릴레이션을 구성하는 속성간에는 순서가 없음
 - 학생1(학번, 이름, 주민번호, 사용자ID)과 학생2(학번, 주민번호, 이름, 사용자ID)는 동일한 릴레이션
- 모든 속성 값은 원자 값(atomic value)
 - 속성 값은 더 이상 나뉘어질 수 없는 논리적인 최소 단위를 의미
 - 예 : 이름

관계형 데이터 모델

❖ 관계 연산자



관계형 데이터 모델

❖ 관계 연산자와 SQL 언어

■ 일반집합 연산자

- UNION : 합집합
 - SQL 예 : SELECT a FROM R UNION SELECT b FROM S;
- INTERSECT : 교집합
 - SQL 예 : SELECT a FROM R INTERSECT SELECT b FROM S;
- DIFFERENCE : 차집합
 - SQL 예 : SELECT a FROM R MINUS SELECT b FROM S;
- PRODUCT : 카티전 곱
 - SQL 예 : SELECT a, b FROM R, S;

관계형 데이터 모델

❖ 관계 연산자와 SQL 언어

■ 순수 관계 연산자

- RESTRICTION : 수평적인 부분 집합
 - SQL 예 : SELECT * FROM R WHERE $r.A=10$;
- PROJECTION : 수직적인 부분 집합
 - SQL 예 : SELECT $r.A_1, r.A_2$ FROM R;
- JOIN : 조건에 만족하는 열 결합
 - SQL 예 : SELECT $r.A, r.B$ FROM R, S WHERE $r.A = s.B$;
- DIVIDE : 두 열 중의 하나의 속성값

관계형 데이터 모델

❖ 제약조건

■ 엔터티 무결성 규칙

- 릴레이션에 저장되는 튜플의 유일성을 보장하기 위한 제약조건
- 기본 키(primary key)
 - 릴레이션에 저장되는 튜플의 유일성을 보장하기 위하여 하나 이상의 속성으로 구성되는 식별자
 - 유일성(uniqueness)과 최소성(minimality)을 만족
- 유일성
 - 기본 키를 구성하는 속성 집합의 값은 릴레이션내에서 유일
 - 예 : 학번
- 최소성
 - 유일성을 해치지 않는 최소의 속성으로 구성
 - 예 : 기본키(학번, 이름) => 학번과 이름을 결합하여 유일성을 보장할 수 있지만, 학번만으로도 유일성을 보장할 수 있으므로 최소성에 위배

관계형 데이터 모델

❖ 제약조건

■ 참조 무결성 규칙

- 릴레이션간의 데이터의 일관성을 보장하기 위한 제약조건
- 하나의 릴레이션에 있는 속성 값이 다른 릴레이션에 있는 속성 값을 참조하기 위해서는 참조되는 속성 값이 반드시 해당 릴레이션에 존재
- 외래 키(foreign key) : 다른 릴레이션의 칼럼 값을 참조하는 칼럼
- 참조 키(reference key) : 다른 릴레이션에서 참조되는 칼럼

■ 도메인 무결성 규칙

- 속성에서 허용 가능한 값의 범위를 지정하기 위한 제약 조건
- 속성의 데이터 타입, 길이, 기본 키, 유일성, NULL 허용, 허용 값의 범위와 같은 제약 조건 지정 가능

관계형 데이터 모델

❖ 관계형 데이터 모델의 성공 요인

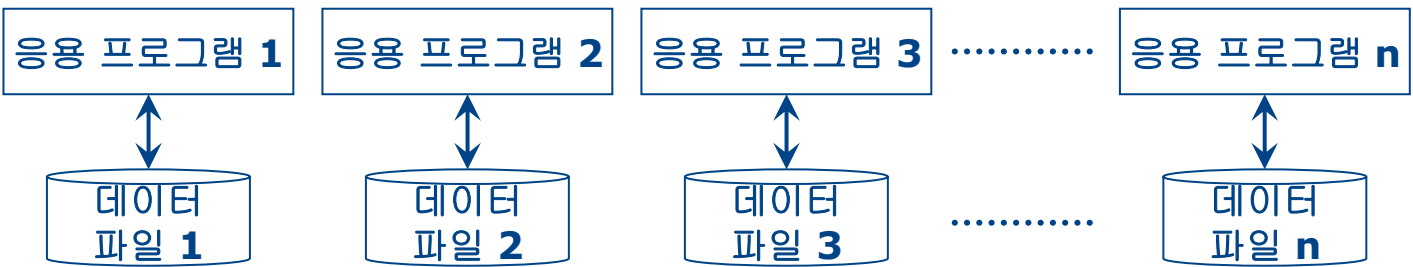
- 단순하고 이해하기 쉬운 모델
 - 테이블의 논리적인 구조가 2차원적인 테이블 형식
- 수학적 이론을 기반으로 한 모델
 - 수학의 집합 이론과 관계 대수를 기반으로 구성
 - 시스템의 기능이나 성능을 예측하거나 검증 가능
 - 질의 최적화나 성능 개선을 위한 기법들은 수학적 기반위에 지원
- 비절차적인 질의어의 존재
 - SQL이라는 비절차적인 언어로 데이터베이스에 접근
- 지속적인 투자와 기술 지원
 - 상용 데이터베이스 회사에서의 지속적인 대규모 투자
 - 새로운 요구에 대한 지속적인 연구 개발

관계형 데이터 모델

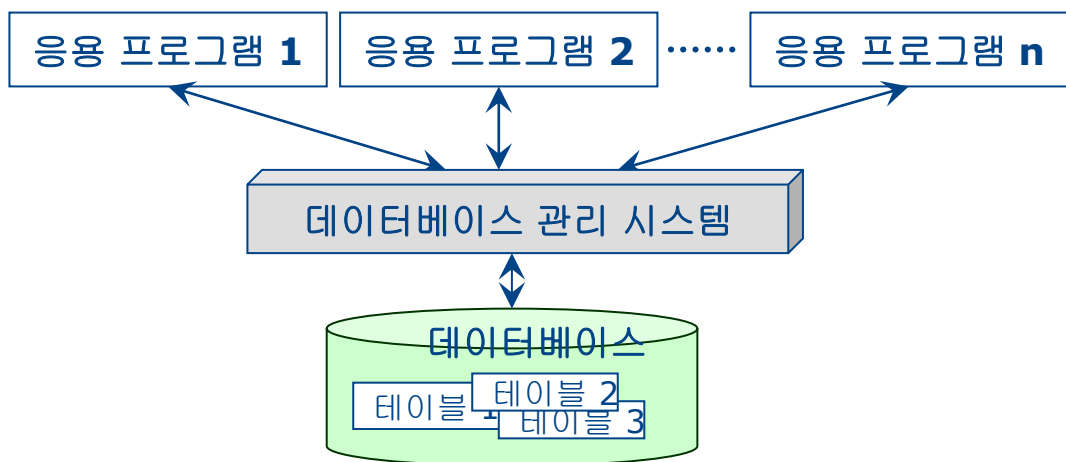
- ❖ 데이터베이스 관리시스템(Database Management System)
 - 사용자 또는 응용 시스템과 데이터베이스간의 인터페이스 역할을 담당하는 시스템 소프트웨어

관계형 데이터 모델

❖ 데이터베이스 관리시스템



a) 파일 시스템을 이용한 응용 프로그램 개발 환경



b) 데이터베이스 관리 시스템을 이용한 응용 프로그램 개발 환경

관계형 데이터 모델

❖ 데이터베이스 관리시스템의 주요 기능

- 데이터 정의 기능
 - 데이터베이스에 스키마를 정의
 - 테이블, 인덱스, 동의어, 시퀀스 등과 같은 스키마 객체(schema object)를 생성, 변경, 삭제
 - CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX등
- 데이터 조작 기능
 - 테이블이나 뷰와 같은 스키마 객체에 저장된 데이터를 조작하는 기능
 - 새로운 데이터를 입력, 수정 또는 삭제하는 기능을
 - INSERT, UPDATE, DELETE문
- 데이터 제어 기능
 - 데이터의 정확성과 안전성을 유지하기 위한 기능
 - 데이터의 정확성 유지를 위한 트랜잭션 관리나 접근 권한 관리 기능
 - COMMIT, ROLLBACK, GRANT, REVOKE등

관계형 데이터 모델

❖ 데이터베이스 관리시스템의 장점

- 데이터 중복의 최소화
 - 데이터의 통합 관리를 통하여 데이터의 중복을 최소한으로 줄일 수 있음
- 데이터의 공유
 - 데이터의 통합 관리를 통하여 여러 사용자가 동일한 데이터를 공유 가능
- 데이터의 일관성 유지
 - 통합된 데이터의 체계적인 관리를 통하여 데이터의 일관성을 유지
- 데이터의 무결성 유지
 - 무결성 규칙에 의해 데이터에 대한 정확성과 일관성을 유지
- 데이터의 보안 보장
 - 데이터에 대한 중앙집중 관리를 통하여 사용자의 접근을 효율적으로 통제
- 전체 데이터에 대한 요구 조정
 - 데이터에 대한 액세스 빈도, 변경 주기, 저장 공간 관리와 사용자 요구 사항을 파악하여 할 전체 데이터에 대한 요구 조건을 조정

관계형 데이터 모델

❖ 데이터베이스 관리시스템의 단점

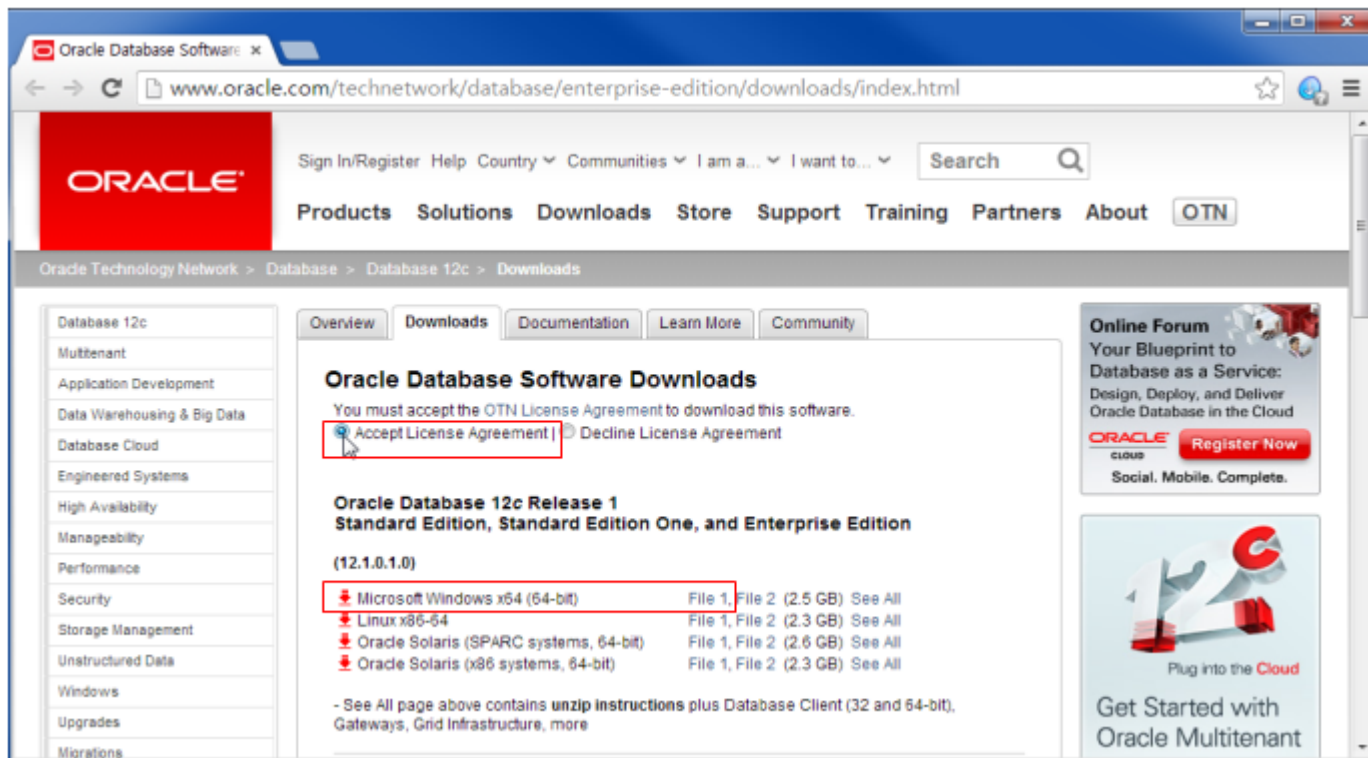
- 운영비 증대
 - 데이터베이스 관리시스템 운영을 위한 하드웨어, 소프트웨어 및 인력 운영 비용이 증가 가능
- 자료 처리의 복잡화
 - 일관성과 보안을 유지하며 데이터를 통합 관리하기 위해 시스템 내부적인 자료 처리가 복잡
- 복잡한 예비와 회복
 - 데이터베이스는 통합된 데이터를 여러 사용자가 동시에 사용하는 관계로 장애 발생시에 복구 과정이 복잡
 - 안정적인 복구를 위해 적절한 백업 작업이 필요
- 시스템의 취약성
 - 데이터에 대한 의존도가 높은 조직이나 업무는 장애 발생시 시스템에 대한 신뢰성과 가용성을 저하

오라클 설치

❖ 오라클 다운로드 및 설치

■ 다운로드

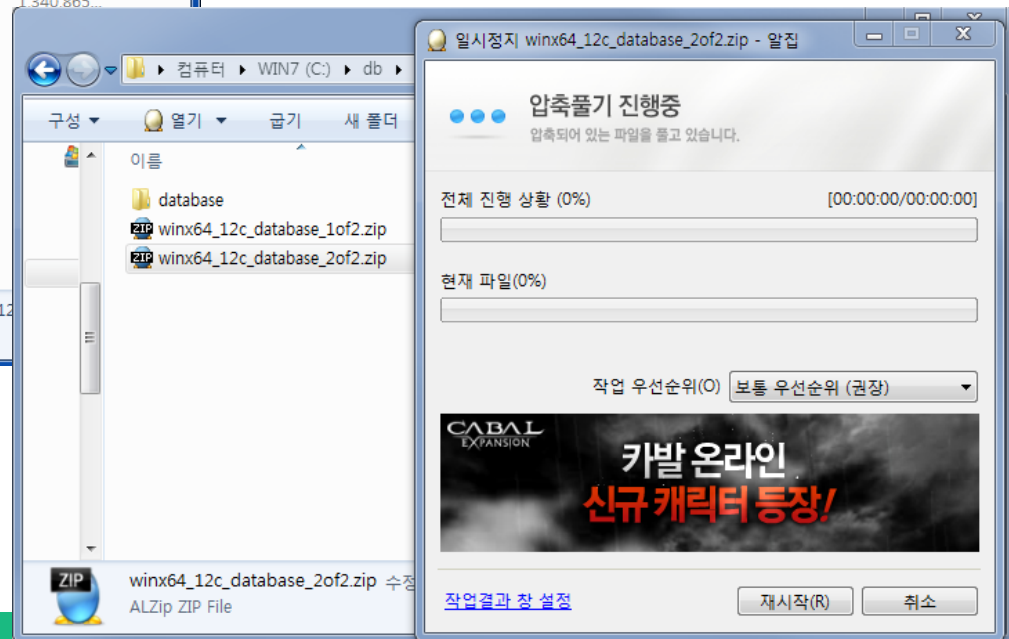
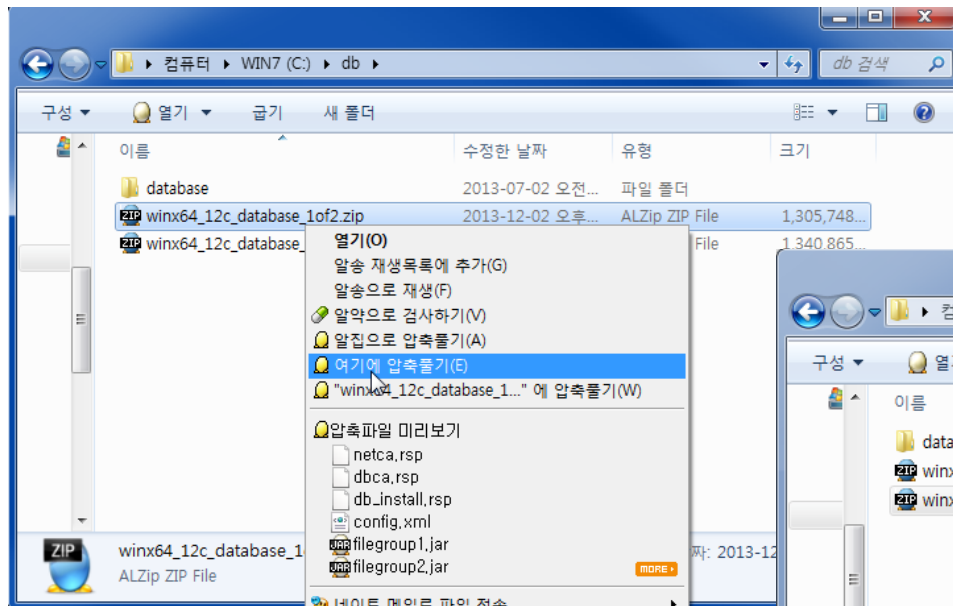
- <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>



오라클 설치

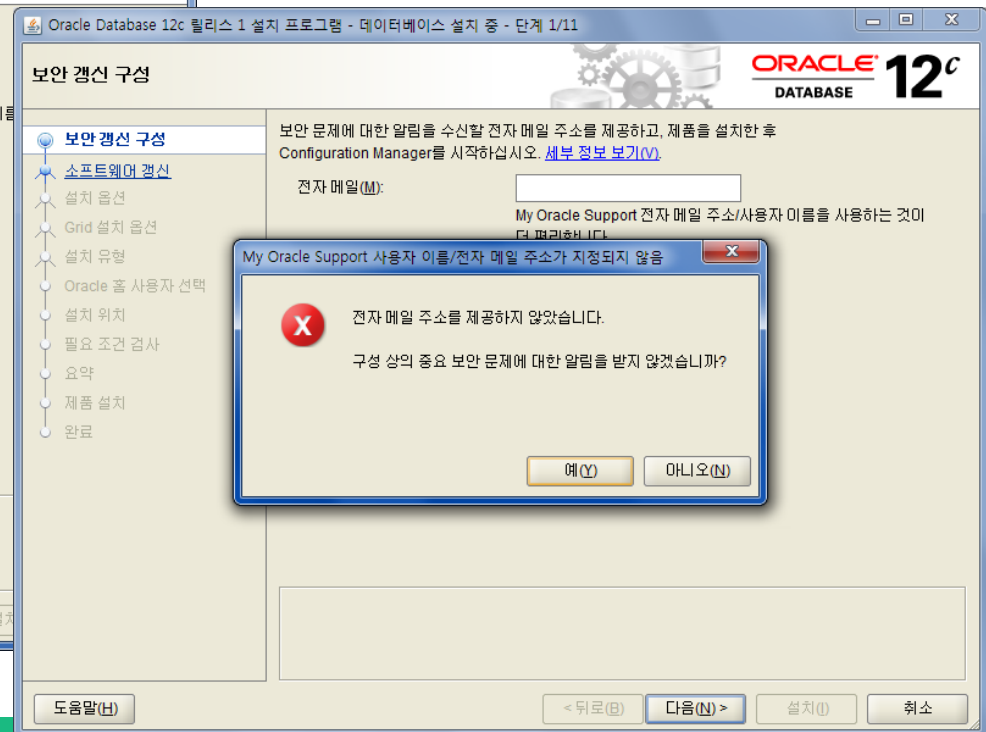
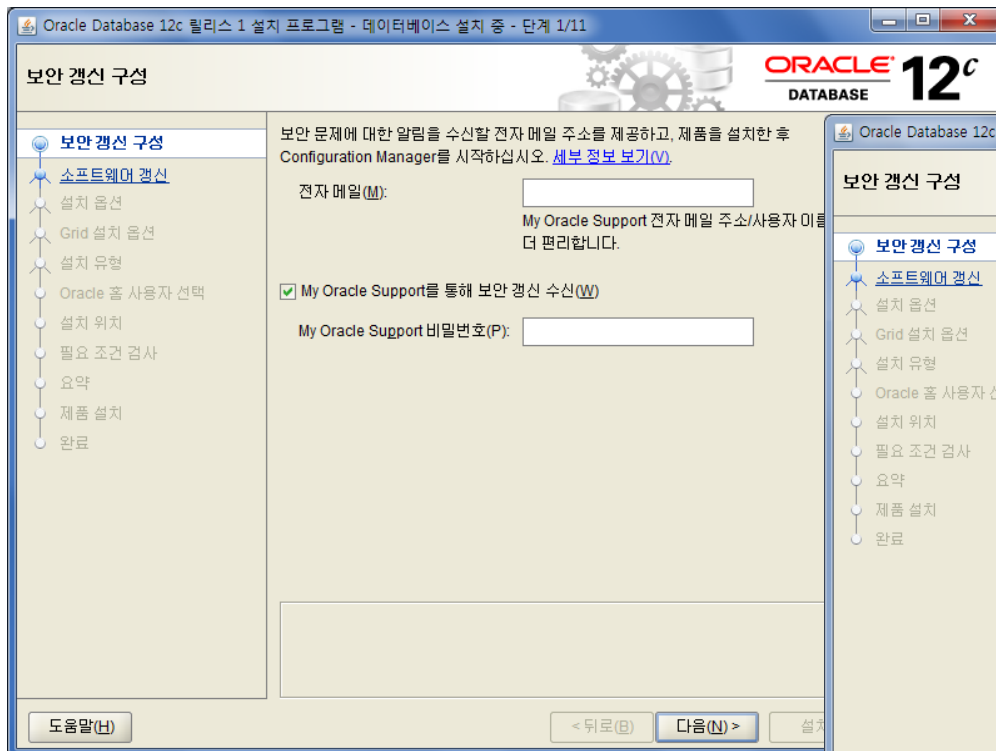
❖ 오라클 설치하기

- 다운로드 받은 File1, File2 파일을 압축 풀면 다음과 같이 database 폴더가 생깁니다.



오라클 설치

- ❖ 압축을 풀어 생긴 database 폴더의 setup.exe 파일을 실행하면 설치 드라이버 로딩 화면이 나타나면서 오라클 설치 시작



오라클 설치

❖ [소프트웨어 갱신]을 위한 단계



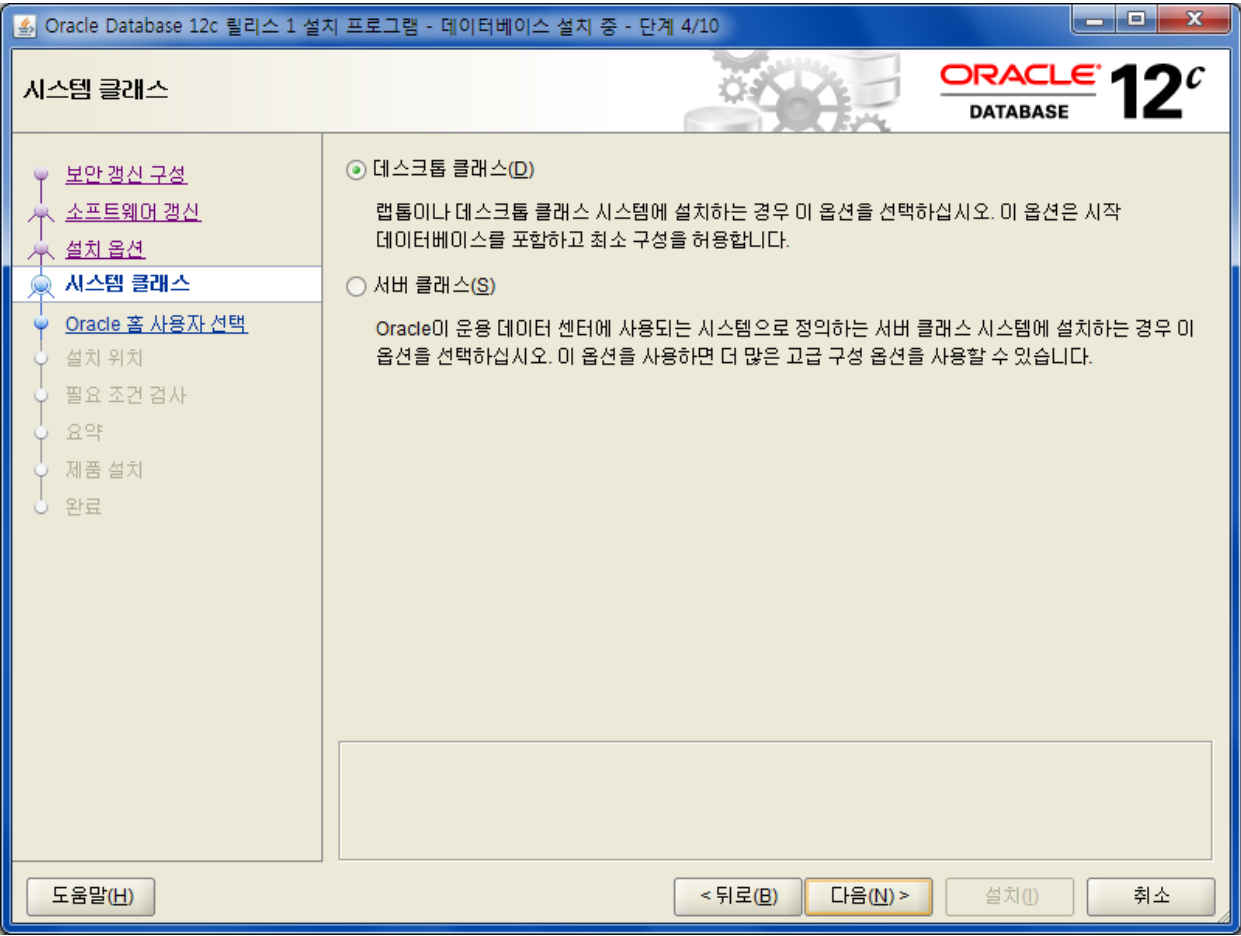
오라클 설치

❖ [설치 옵션 선택]을 위한 단계



오라클 설치

❖ [시스템 클래스]를 위한 단계



오라클 설치

❖ [Oracle 홈 사용자 지정]을 위한 단계

Oracle Database 12c 릴리스 1 설치 프로그램 - 데이터베이스 설치 중 - 단계 5/10

Oracle 홈 사용자 지정

Oracle은 보안 향상을 위해 표준 Windows 사용자 계정(관리자 계정 마님)을 지정하며 Oracle 홈을 설치 및 구성할 것을 권장합니다. 이 계정은 Oracle 홈에 대한 Windows 서비스를 실행하는 데 사용됩니다. 관리 작업을 수행하기 위해 이 계정으로 로그인하지 마십시오.

☐ 기존 Windows 사용자 사용(X)

사용자 이름(U):

비밀번호(P):

☒ 새 Windows 사용자 생성(C)

사용자 이름(S):

비밀번호(A):

비밀번호 확인(Q)?

새로 생성된 사용자의 Windows 로그인 권한이 거부되었습니다.

☐ Windows 내장 계정 사용(L)

도움말(H) < 뒤로(B) 다음(N) > 설치(I) 취소

오라클 설치

❖ [일반설치 구성]을 위한 단계

Oracle Database 12c 릴리스 1 설치 프로그램 - 데이터베이스 설치 중 - 단계 6/10

일반 설치 구성

기본 구성을 사용하여 전체 데이터베이스 설치를 수행합니다.

Oracle Base(S): C:\app\lrcld [찾아보기(R)...]

소프트웨어 위치(L): C:\app\lrcld\product\12.1.0\dbhome_1 [찾아보기(Q)...]

데이터베이스 파일 위치(D): C:\app\lrcld\loradata [찾아보기(W)...]

데이터베이스 버전(E): Enterprise Edition(6.0GB)

문자 집합(I): 기본값 (KO16MSWIN949)

전역 데이터베이스 이름(G): orcl

관리 비밀번호(P):

비밀번호 확인(C): ?

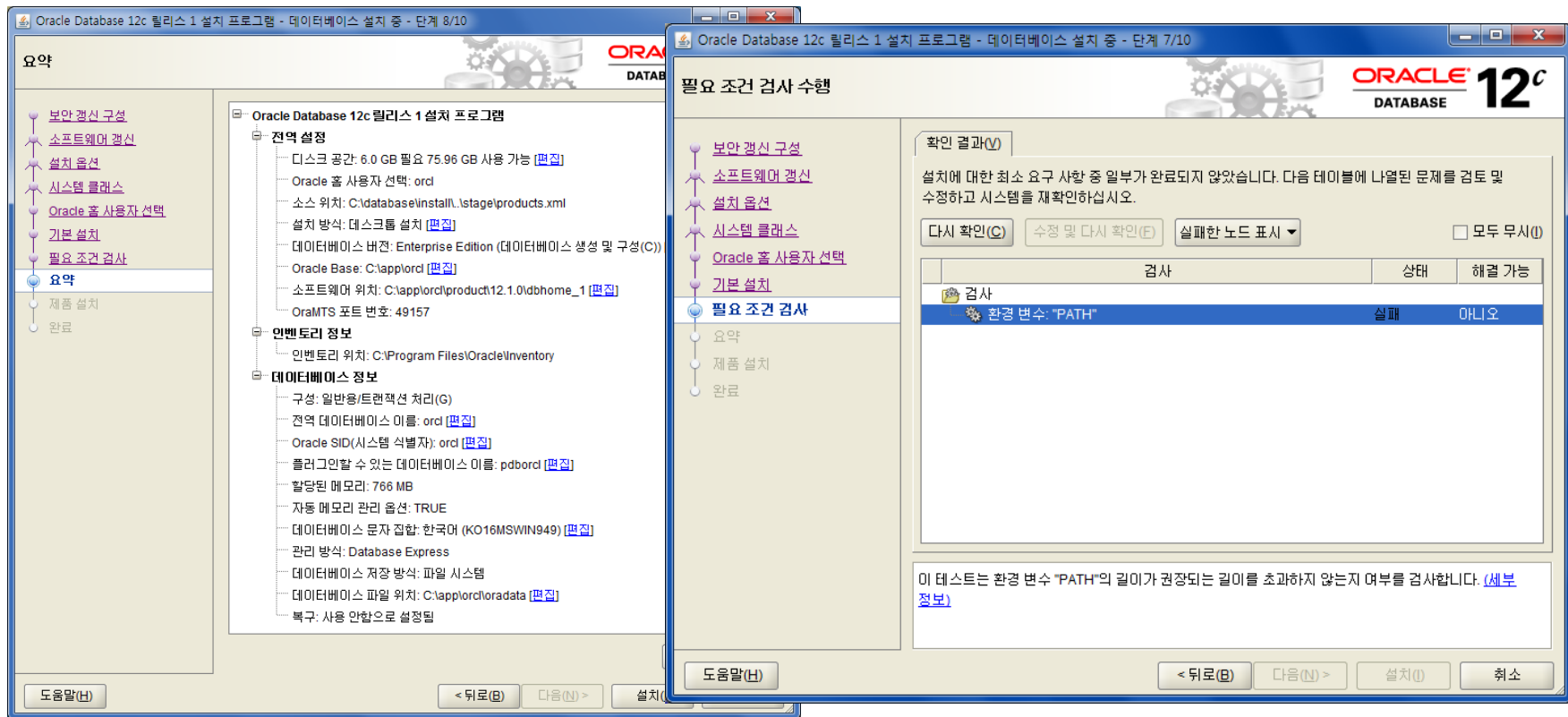
☒ 컨테이너 데이터베이스로 생성(I)

플러그인할 수 있는 데이터베이스 이름(U): pdborcl

도움말(H) < 뒤로(B) 다음(N) > 설치(I) 취소

오라클 설치

- ❖ 필요 조건 검사를 진행
- ❖ 설정한 옵션들을 설치 전 요약하여 보여주는 단계



SQL명령어 종류

유형	명령문	기능
질의어 (DQL:Data Query Language)	SELECT	데이터 검색
데이터 조작어 (DML:Data Manipulation Language)	INSERT UPDATE DELETE	데이터 입력 데이터 수정 데이터 삭제
데이터 정의어 (DDL:Data Definition Language)	CREATE ALTER DROP RENAME TRUNCATE	데이터베이스 객체생성 데이터베이스 객체변경 데이터베이스 객체삭제 데이터베이스 객체 이름 변경 데이터 및 저장 공간 삭제
트랜잭션 처리어 (TCL:Transaction Control Language)	COMMIT ROLLBACK SAVEPOINT	트랜잭션의 정상적인 종료 처리 트랜잭션 취소 트랜잭션내에 임시 저장점 설정
데이터 제어어 (DCL:Data Control Language)	GRANT REVOKE	데이터베이스 객체에 대한 접근 권한 부여 데이터베이스 객체에 대한 접근 권한 취소

오라클 접속하기

❖ 기본 사용자 계정

- 시스템 권한을 가진 사용자인 DBA용 계정(SYS, SYSTEM)과 교육용 계정(HR) 두 가지로 나눕니다.
- sys
 - 오라클 super 사용자 계정, 모든 객체의 소유주
- system
 - 데이터베이스 관리자 권한, 데이터베이스 생성권한 없음
- hr
 - 사용자 실습을 위한 교육용 계정
- Scott
 - 일반 사용자 권한

예제 테이블

❖ 부서(DEPT)테이블

칼럼이름	데이터 타입	Key Type	NN/ Unique	FK table	FK column	설명
DEPTNO	NUMBER(2)	PK	NN,U			부서번호
DNAME	VARCHAR2(14)					부서이름
LOC	VARCHAR2(13)					지역명

예제 테이블

❖ 사원(EMP)테이블

칼럼이름	데이터 타입	Key Type	NN/ Unique	FK table	FK column	설명
EMPNO	NUMBER(4)	PK	NN,U			사원번호
ENAME	VARCHAR2(10)					사원명
JOB	VARCHAR2(9)					업무명
MGR	NUMBER(4)	FK	NN	EMP	EMPNO	상사번호
SAL	NUMBER(7,2)					급여
HIREDATE	DATE					입사일
COMM	NUMBER(7,2)					커미션
DEPTNO	NUMBER(2)	FK	NN	DEPT	DEPTNO	부서번호

예제 테이블

❖ 급여(SALGRADE)테이블

칼럼이름	데이터 타입	Key Type	NN/ Unique	FK table	FK column	설명
GRADE	NUMBER					급여등급
LOSAL	NUMBER					급여 하한 값
HISAL	NUMBER					급여 상한 값

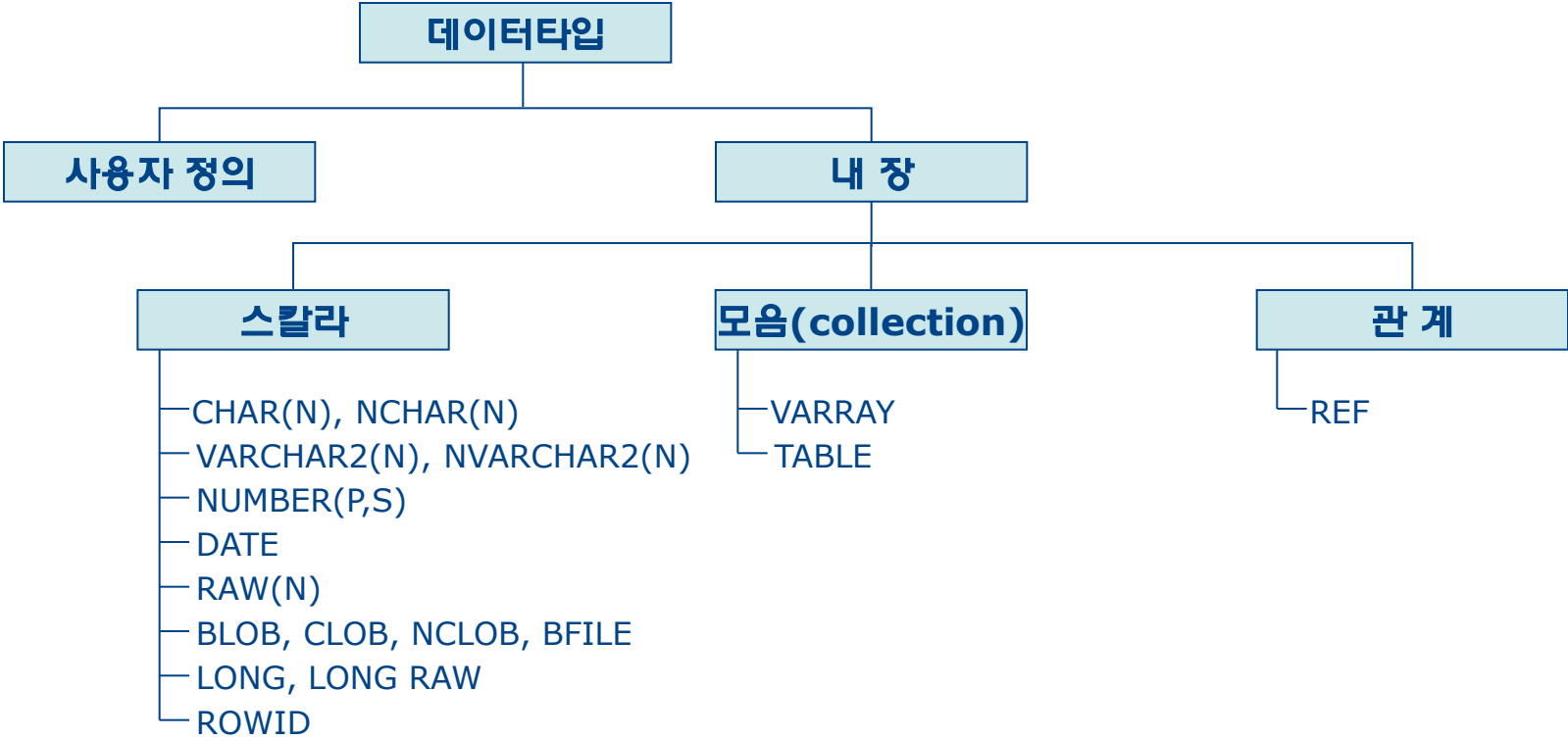
데이터 타입의 종류

❖ 개요

- 데이터베이스에 저장되는 데이터는 문자, 숫자, 날짜, 이미지 등과 같은 다양한 형식으로 표현
- 실무에서 칼럼에 대한 데이터 타입이나 길이의 결정은 데이터베이스 설계 단계에서 신중하게 고려해야하는 중요한 사항
- 데이터베이스 설계 단계에서 잘못 선정된 데이터 타입은 응용 프로그램 개발을 어렵게 하거나 성능을 저하시키는 요인이 됨

데이터 타입의 종류

❖ 오라클에서 지원하는 데이터 타입의 종류



데이터 타입의 종류

데이터 타입의 종류	의미
CHAR(size)	size 크기의 고정 길이 문자 데이터 타입 최대크기 : 2,000 바이트 최소크기 : 1 바이트
VARCHAR2(size)	size 크기의 가변 길이 문자 데이터 타입 최대크기 : 4,000 바이트 최소크기 : 1 바이트
NVARCHAR2(size)	국가별 문자 집합에 따른 size 크기의 문자 또는 바이트의 가변길이 문자 데이터 타입 최대크기 : 4,000 바이트 최소크기 : 1 바이트
NUMBER(p,s)	정밀도(p)와 스케일(s)로 표현되는 숫자 데이터 타입 p : 1 ~ 38 s : -84 ~ 127
DATE	날짜 형식을 저장하기 위한 데이터 타입

데이터 타입의 종류

데이터 타입의 종류	의미
ROWID	테이블내 행의 고유 주소를 가지는 64진수 문자 타입 행당 6 바이트(제한된 ROWID) 또는 10 바이트(확장된 ROWID)
BLOB	대용량의 바이너리 데이터를 저장하기 위한 데이터 타입 최대크기 : 4GB
CLOB	대용량의 텍스트 데이터를 저장하기 위한 데이터 타입 최대크기 : 4GB
BFILE	대용량의 바이너리 데이터를 파일 형태로 저장하기 위한 데이터 타입 최대크기 : 4GB
TIMESTAMP(n)	DATE 데이터 타입의 확장된 형태 n 은 milli second 자리수로 최대 9자리까지 표현 가능
INTERVAL YEAR TO MONTH	년과 월을 이용하여 기간을 저장
INTERVAL DAY TO SECOND	일, 시, 분, 초를 이용하여 기간을 저장 두 날짜 값의 정확한 차이를 표현하는데 유용