

# 1장. 오라클 DB 환경설정 및 구성 객체

*Oracle – SQL 디벨로퍼*



# 데이터, 정보, 지식

## 데이터, 정보, 지식

- 데이터가 모이면 정보가 되고 정보가 모이면 지식이 된다.
- 데이터(data) : 관찰의 결과로 나타난 정량적 혹은 정성적인 실제값
- 정보(information) : 데이터에 의미를 부여한 것
- 지식(knowledge) : 사물이나 현상에 대한 이해

예) 관악산의 높이는 데이터, 관악산의 지리적인 특성을 설명한 것은 정보에 관악산을 등산하는 가장 좋은 방법을 소개하는 보고서는 지식에 해당된다.

## 데이터베이스(Database)

- 조직에 필요한 정보를 얻기 위해 논리적으로 연관된 데이터를 모아 구조적으로 통합해 놓은 것
- 데이터들이 모여있는 데이터의 집합으로 **서로 관련 있는 데이터들의 모임이다.**  
(메모장에 두서없이 적어 놓은 단어들의 모임은 데이터베이스가 아님)



# 데이터베이스

## 데이터베이스(Database)와 생활

- 학교 홈페이지에서 수강신청, 성적 조회 - 학사 데이터 베이스에 저장
- 은행의 신용카드 거래 내역 - 은행 데이터베이스
- 병원의 진료 기록 - 건강보험 데이터베이스

학생 테이블

학번	이름	생년월일	학과명
20150001	오상식	1987. 6. 10	컴퓨터공학과
20171010	최정보	1995. 5. 5	전자공학과
20182121	김나래	1993. 12. 1	기계공학과

과목 테이블

과목번호	과목명	담당교수
0303	웹 프로그래밍	송미영
0116	데이터베이스	오용철



# 데이터베이스

## 데이터베이스 시스템의 발전

- 적은 양의 데이터를 저장하고 검색하는 수준에서 지금은 데이터 자체 발생량이 증가하고 인터넷과 소셜 네트워크, 전자상거래 등의 서비스를 제공할수 있게됨

## 데이터베이스 시스템의 예 : 마당서점

[1단계] 마당서점의 시작 – 1970년대, 박마당이라는 사업가가 작은 서점 오픈함

- 도서 100권
- 고객 : 근처 학교의 학생, 지역주민
- 업무 : 회계 업무(계산기 사용), 장부에 기록
- 고객 서비스 : 사장이 직접 도서 안내

[2단계] 컴퓨터의 도입 – 1980년대

- 도서 1,000권
- 고객 : 근처 학교의 학생, 지역주민
- 업무 : 회계 업무(컴퓨터 사용), 파일 시스템
- 고객 서비스 : 컴퓨터를 이용하여 도서 검색, 직원 고용



# 데이터베이스

## 데이터베이스 시스템의 예 : 마당서점

### [3단계] 지점 개설 및 데이터베이스 구축

- 도서 10,000권
- 고객 : 서울지역 고객
- 업무 : 회계 업무(컴퓨터 사용), 데이터 베이스 시스템
- 고객 서비스 : 클라이언트/서버 시스템으로 지점을 연결하여 도서 검색 및 서비스 제공

### [4단계] 홈페이지 구축

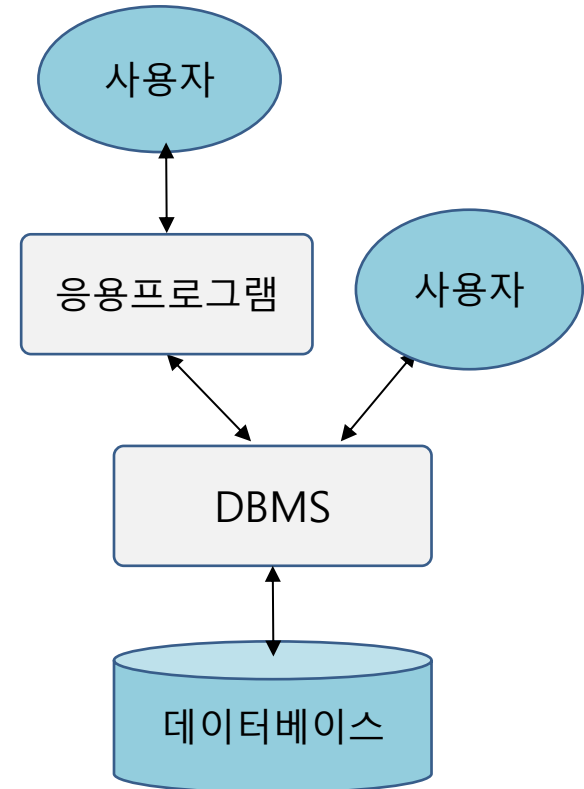
- 도서 100,000권
- 고객 : 국민(전국으로 배송)
- 업무 : 회계/인사 업무(컴퓨터 사용), 웹 DB 시스템으로 지점간 연계
- 고객 서비스 : 인터넷으로 도서 검색 및 주문



# 데이터베이스 관리 시스템

## 데이터베이스 관리 시스템(Database Management System)

- 많은 양의 데이터를 정교하게 구축하고 관리하는 소프트웨어이다.
- 데이터베이스의 정의, 데이터베이스 갱신, 질의 처리, 유지보수, 보안 등의 편리한 기능을 제공한다.
- 주기억장치에서 실행되며 *컴파일러, 질의 처리기, 트랜잭션 관리자* 등을 가지고 있다.
- 대표적으로 오라클 사의 Oracle 과 MySQL, 마이크로소프트사의 MSSQL등이 있다



# 데이터베이스 관리 시스템

## 데이터베이스 관리 시스템의 장점

- **데이터의 중복과 불일치 감소**

데이터가 여러 곳에 분산되어 있으면 중복 저장될 수 있고, 같은 의미의 데이터가 다른 값을 갖게 되는 불일치가 생길 수 있다.

- **질의 처리에 효율적인 저장 구조**

사용자는 질의(Query)를 통해서 데이터베이스에 접근하는데 시간이 소요되지만 DBMS는 시간을 줄이도록 저장 구조가 설계되어 있다.

- **백업(Backup)과 복구(Recovery)**

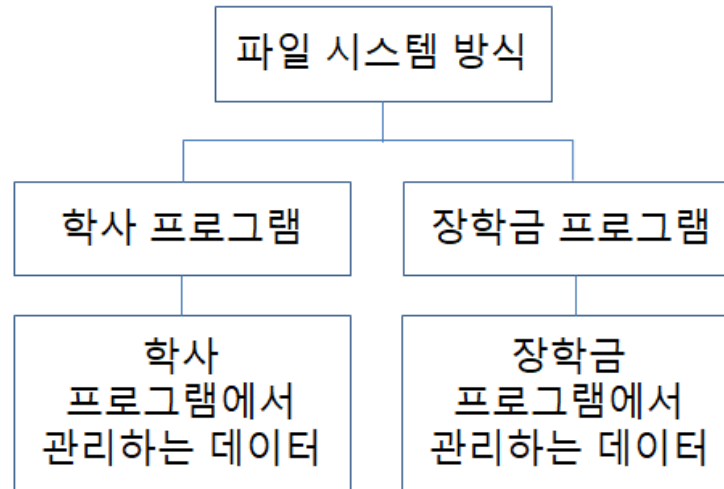
데이터는 저장과 동시에 반드시 백업(따로 복사)되어야 한다. 복구는 트랜잭션(가장 작은 업무 단위)을 관리하여 데이터베이스가 피해를 보기 전 상태로 복구하는 것이다.

※ 단점 : 사용하는 자원이 많고 복잡하며 비싸다.



# 파일 시스템과 DBMS

## 파일시스템과 DBMS



파일 시스템은 서로 다른 여러 응용 프로그램이 제공하는 기능에 맞게 필요한 데이터를 각각 저장하고 관리한다.

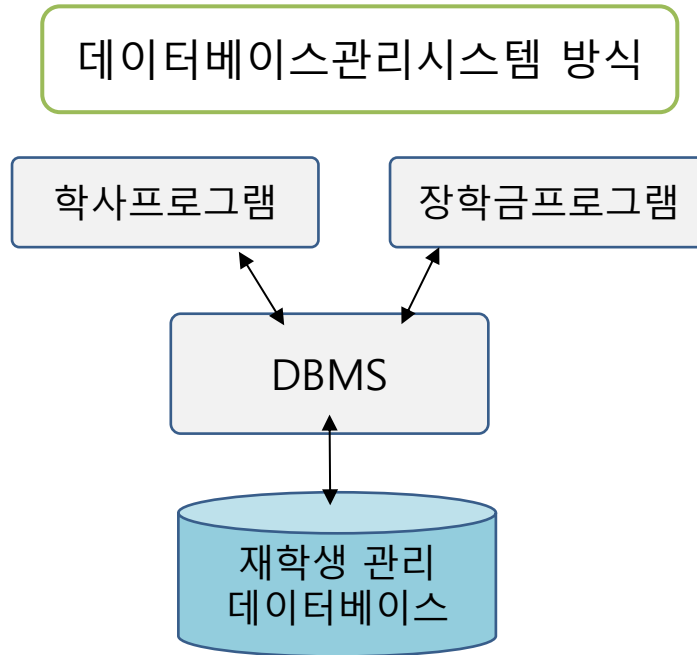
따라서 각 파일에 저장한 데이터는 서로 연관이 없고 중복 또는 누락이 발생할 수 있다.





# 파일 시스템과 DBMS

## 파일시스템과 DBMS



학생과 관련된 일련의 데이터를 한곳에 모아 관리하면 데이터의 오류, 누락, 중복 등의 문제를 해결할 수 있다.



# 파일 시스템과 DBMS

## 파일 시스템 방식의 문제점

이순신 학생이 졸업했는데 업데이트가 되지 않아 재학중으로 되어 있어 장학금 신청이 가능한 걸로 오류 발생

학사 프로그램

학번	이름	학과	상태
2019-0001	홍길동	컴퓨터공학과	군휴학
2019-0002	이순신	경영학과	졸업
2019-0003	유관순	철학과	재학

장학금 신청 프로그램

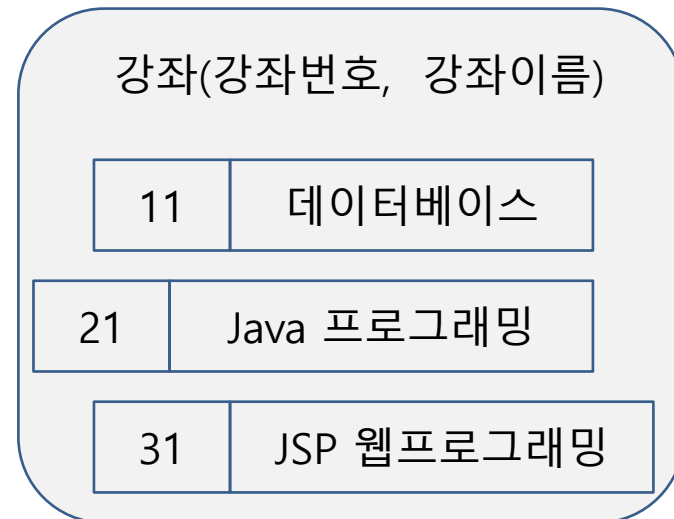
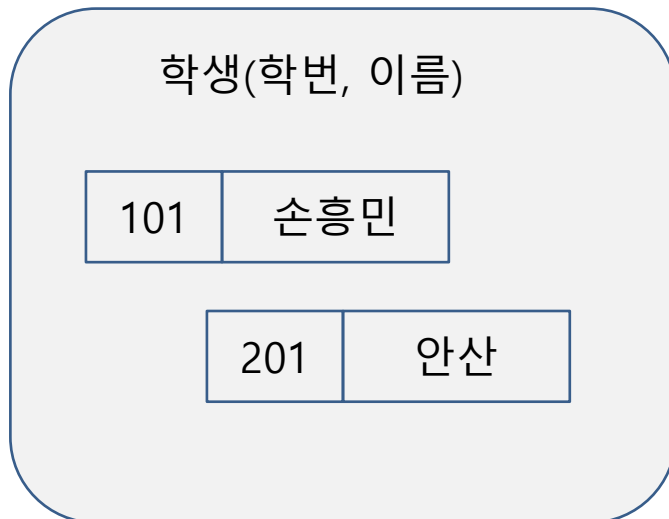
장학금	이름	상태	가능여부
국가	홍길동	군휴학	신청불가
성적	이순신	재학	신청가능
근로	유관순	재학	신청가능



# 데이터 모델

## 데이터 모델

- 컴퓨터에 데이터를 저장하는 방식을 정의해 놓은 개념 모형이다.
- 계층형 데이터 모델, 네트워크형 데이터 모델, 관계형 데이터 모델(가장 많이 사용), 객체-관계 데이터 모델(객체 지향 언어가 널리 사용됨에 따라 관계 데이터 모델과 객체 데이터 모델의 장점을 결합한 모델)
- 학생 테이블과 강좌테이블이 학생-강좌 관계를 표현  
손흥민이 Java 프로그래밍을 안산이 JSP 웹프로그래밍을 수강한다고 가정



# 데이터 모델

## 데이터 모델링(Data Modeling)

- 데이터 베이스의 설계시 클라이언트의 요구를 분석하여 논리모델을 구성하고 물리모델을 사용해 데이터베이스에 반영하는 작업
- 기본 요소

구분	개념	실제 예
엔티티(Entity)	물리적 개념에서는 테이블로 표현	고객, 상품, 주문
속성(Attribute)	물리적 개념에서는 칼럼(Column)으로 표현	고객아이디, 고객명, 주소
관계(Releationship)	기본키와 참조키로 정의 됨(일대일, 일대다)	고객과 주문과의 관계



# 데이터 모델

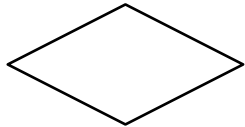
## ◆ 개념적 설계

현실세계를 추상화(특성화)하여 개체 타입과 관계를 파악하여 표현하는 과정

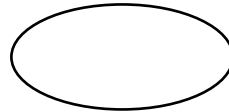
→ 개체 관계도(E-R 다이어그램) : Entity-Relationship Diagram



개체



관계



속성

## ◆ 논리적 설계

개념적 설계에서 만들어진 구조를 논리적으로 구현 가능한 데이터 모델로 변환하는 단계로 사용자가 알아볼 수 있는 형태로 변환하는 과정 -> 테이블(표) 형태

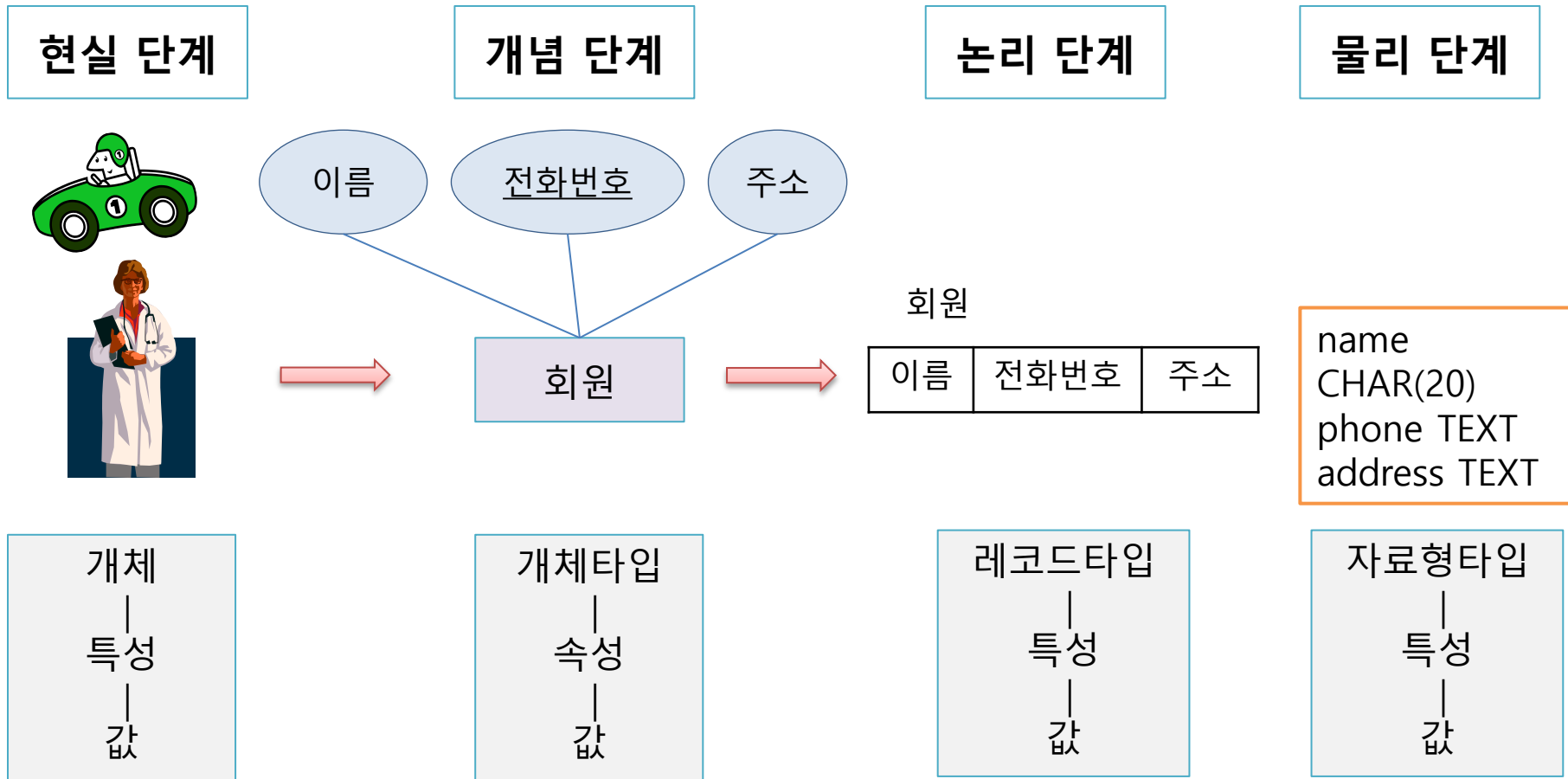
## ◆ 물리적 설계

논리적 데이터베이스 구조를 실제 기계가 처리하기에 알맞도록 내부 저장 장치 구조와 접근 경로 등을 설계하는 과정

예) name char(20) – name은 문자형 20Byte를 의미함



# 데이터 모델



# 관계형 데이터베이스

- 관계형 데이터 모델

- 데이터간의 관계에 초점을 둔 모델로 현재 가장 많이 사용하는 모델이다.

예) 회사의 사원정보, 소속된 부서정보 데이터 관리

- 사원 정보와 부서 정보를 하나의 묶음으로 관리하면 데이터 구조가 간단해진다. 하지만 같은 부서 사원들은 부서 정보가 중복되므로 효율적인 관리가 어려워진다. 왜냐하면 부서 이름이 바뀌면 사원들의 부서 정보를 일일이 찾아서 수정해주어야 한다.

사원 정보	사원 번호	사원 이름	사원 직급	부서이름	위치
사원 번호	0001	홍길동	과장	회계팀	서울
사원 이름	0002	성춘향	대리	연구소	수원
사원 직급	0003	이몽룡	사원	영업팀	분당
부서이름	0004	심청이	사원	회계팀	서울
위치					

데이터 중복발생

<정규화 전의 형태>

# 관계형 데이터베이스



※ 정규화 후의 형태 -> 1대 多의 구조로 변경된다.  
한 부서에는 여러 명의 사원이 존재한다.

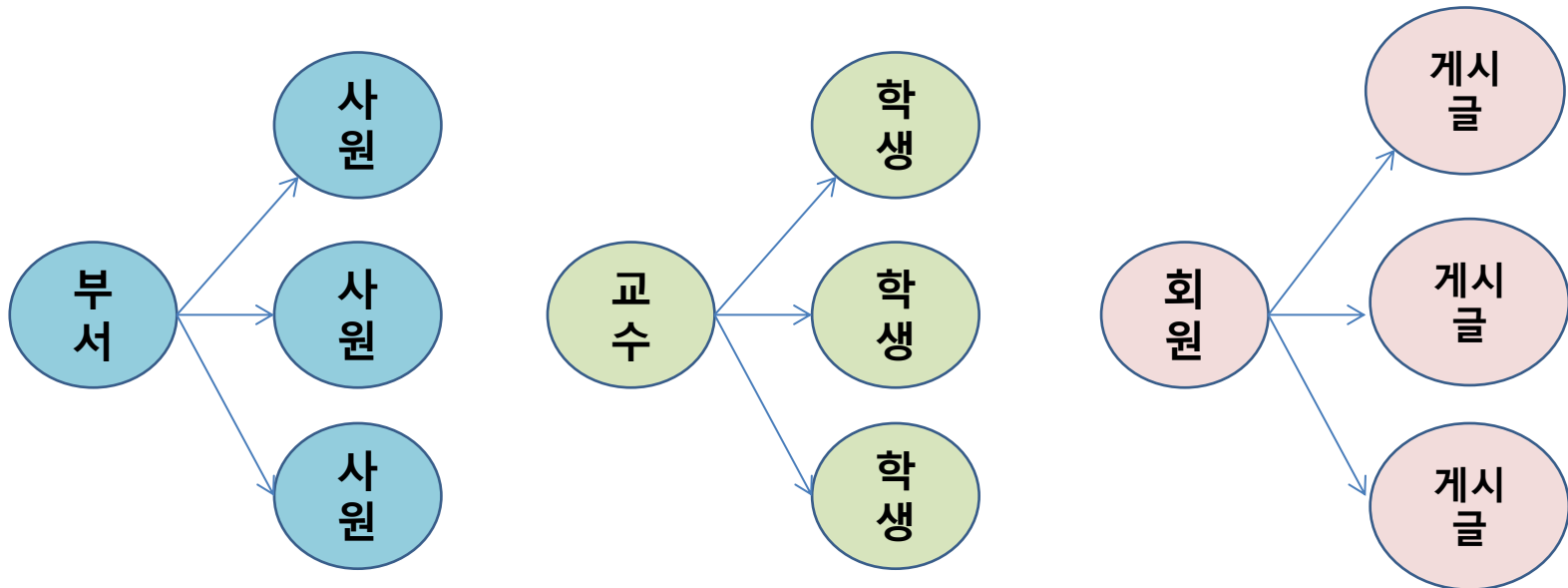




# 관계(Releation)

## 엔티티(Entity) 관계(releation)

- 1대 多의 관계, 1대 1관계, 多 대 多 관계



# 관계형 데이터베이스

## 관계형 데이터베이스의 구성 요소

- 테이블(Table)

표 형태의 데이터 저장 공간을 테이블이라고 한다. 2차원 형태로 행과 열로 구성

행(ROW) - 저장하려는 하나의 개체를 구성하는 여러 값을 가로로 늘어뜨린 형태다.

열(COLUMN) - 저장하려는 데이터를 대표하는 이름과 공통 특성을 정의

학생

학번	이름	생년월일	학과명
20150001	오상식	1987. 6. 10	컴퓨터공학과
20171010	최정보	1995. 5. 5	전자공학과
20182121	김나래	1993. 12. 1	기계공학과

속성, 열, 칼럼, 애트리뷰트

튜플, 레코드, 행



# 관계형 데이터베이스

## 관계형 데이터베이스의 구성 요소

- 특별한 의미를 지닌 열 - 키

### 기본키(Primary Key)

- 테이블의 지정된 행을 식별할 수 있는 유일한 값이어야 한다.
- 값의 중복이 없어야 한다.
- NULL값을 가질 수 없다.

### 보조키

- 대체키 또는 후보키라 하며 후보키 중에서 기본키로 지정되지 않은 열이다.



# 관계형 데이터베이스

## 외래키(FK : Foreign Key)

- 특정 테이블에 포함되어 있으면서 다른 테이블의 기본키로 지정된 키



# SQL이란?

- SQL(Structured Query Language)

- '에스큐엘', 또는 '시퀄'이라 부른다.
- 사용자와 데이터베이스 시스템 간에 의사 소통을 하기 위한 언어이다.
- 사용자가 SQL을 이용하여 DB 시스템에 데이터의 검색, 조작, 정의 등을 요구하면 DB 시스템이 필요한 데이터를 가져와서 결과를 알려준다.

구분	개념
DDL(Data Definition Language) - 데이터 정의어	테이블을 포함한 여러 객체를 생성, 수정, 삭제하는 명령어
DML(Data Manipulation Language) - 데이터 조작어	데이터를 저장, 검색, 수정, 삭제하는 명령어
DCL(Data Control Language) - 데이터 제어어	데이터 사용 권한과 관련된 명령어



# 오라클 데이터 베이스

## ❖ 오라클 데이터베이스와 버전

### ▷ Oracle 데이터베이스

- 오라클사가 만든 DBMS 제품이다.
- 최신 버전은 2021년에 출시한 21c 버전이다.

현재 일반적으로 Express-Edition 버전을 많이 사용한다.

### ▷ Oracle 데이터베이스 설치

1. 계정 생성하고 로그인 하기
2. 다운로드 하기
3. 파일 압축 풀기
4. 설치 프로그램 실행하기



# 오라클 데이터 베이스

❖ 오라클 데이터베이스와 설치 -> 검색 : 오라클 DB 다운로드

데이터베이스

Database Enterprise/Standard Editions

Database Express Edition

Audit Vault and Database Firewall

Berkeley DB

Big Data Connectors

클러스터 검증 유틸리티

## Oracle Database XE Downloads

### Oracle Database 21c Express Edition

Download



Oracle Database 21c Express Edition for Windows x64

### 오라클 계정 로그인

사용자 이름

kiyongee2@gamil.com

암호

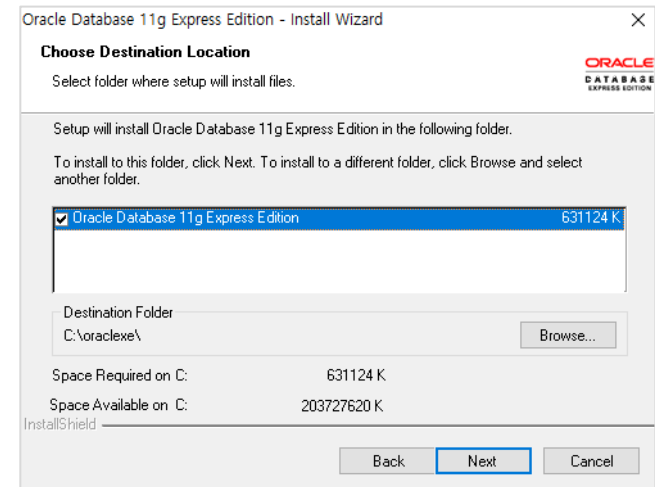
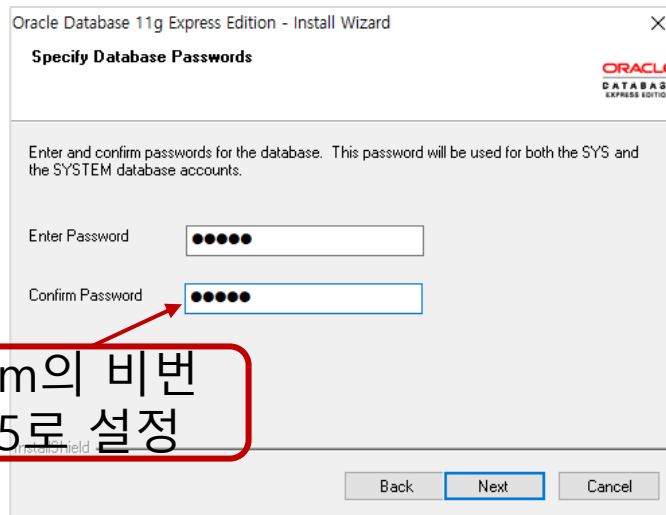
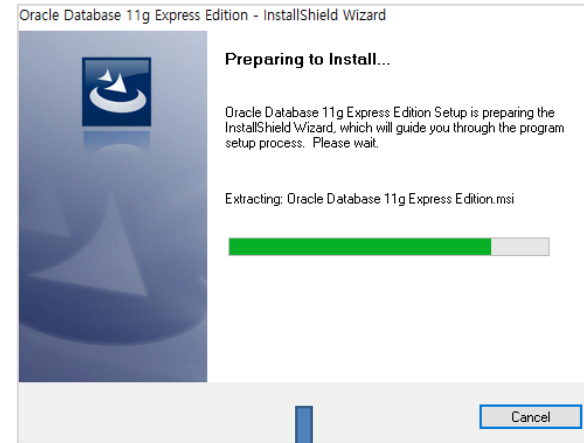
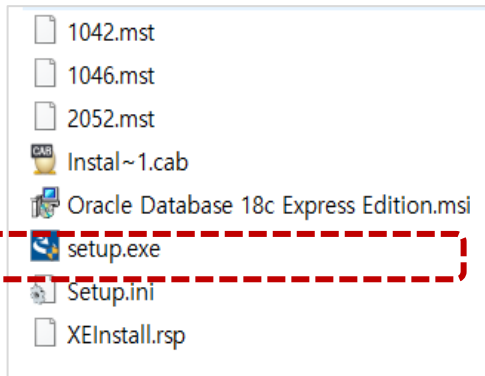
.....

로그인



# 오라클 데이터 베이스

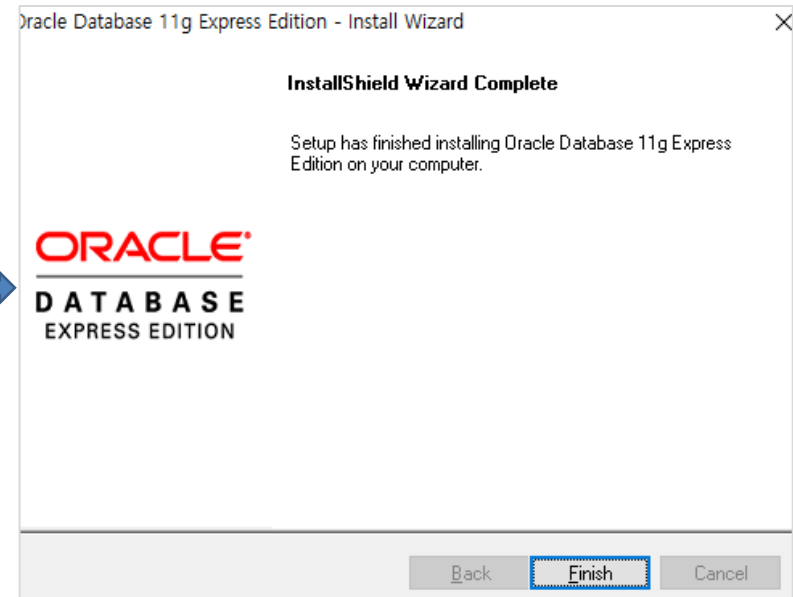
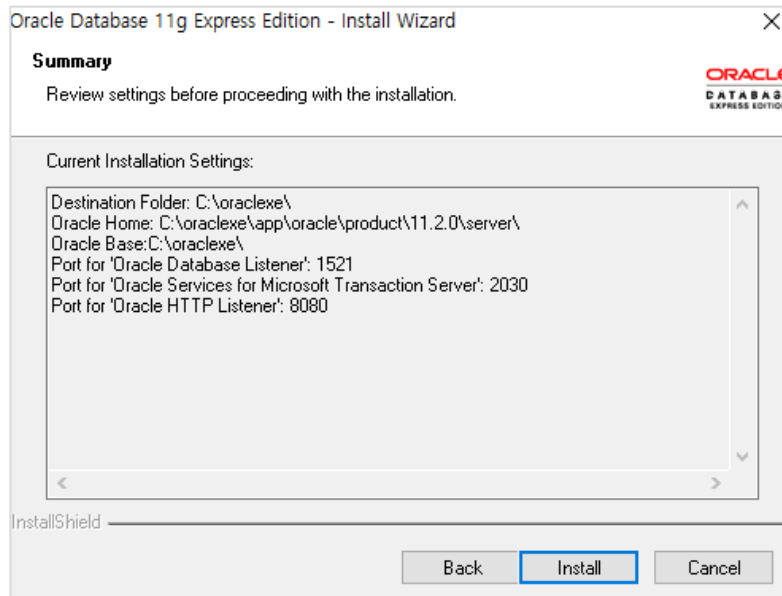
## ❖ 오라클 데이터베이스와 설치





# 오라클 데이터 베이스






## ❖ 오라클 데이터베이스와 설치



# 오라클 데이터 베이스

## ❖ 오라클 데이터베이스와 설치 후 확인

내컴퓨터 – 우측마우스 – 관리 – 서비스 및 응용프로그램 - 서비스

 OracleJobSchedulerXE		사용 안 함
 OracleOraDB21Home1MTSR...	실행 ...	자동
 OracleOraDB21Home1TNSLi...	실행 ...	자동
 OracleServiceXE	실행 ...	자동
 OracleVssWriterXE		자동



## sqlplus

오라클 설치시 자동으로 지원되는 프로그램이다.

- ① 명령프롬프트(cmd) 열기
- ② **sqlplus**(DBMS 소프트웨어) 입력
- ③ 사용자명(user) : **system**
- ④ 비밀번호 : **12345**

```
C:\Users\kikiyon>sqlplus
SQL*Plus: Release 21.0.0.0.0 - Production on 토 7월 2 08:28:19 2022
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

사용자명 입력 : system
비밀번호 입력 :
마지막 성공한 로그인 시간: 토 7월 02 2022 08:21:30 +09:00

다음에 접속됨:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL>
```



# SQL developer

## 오라클 SQL 디벨로퍼 : (sqldeveloper 다운로드 검색)

Oracle SQL developer는 오라클 데이터베이스에서 SQL 작업을 수행하는 통합개발환경 (IDE)이다. SQL과 PL/SQL 코드 작성을 위해 다양한 기능을 제공하며 프리웨어이다.

### 개발자 툴

ADF Faces

Application Express(APEX)

BI Publisher

BPEL Process Manager

Developer Studio

Developer Suite 10g

Oracle REST Data Services(ORDS)

SOA Suite

Spatial Studio

SQL Developer

SQL Developer Data Modeler

StorageTek



# 오라클 SQL 디벨로퍼

## ◆ 오라클 SQL 디벨로퍼 Download

### SQL Developer 22.2 Downloads

Version 22.2.0.173.2018 - June 28, 2022

- [Release Notes](#)
- [Documentation](#)

#### Platform

#### Download

Windows 64-bit with JDK 11 included

 [Download](#) (434 MB)

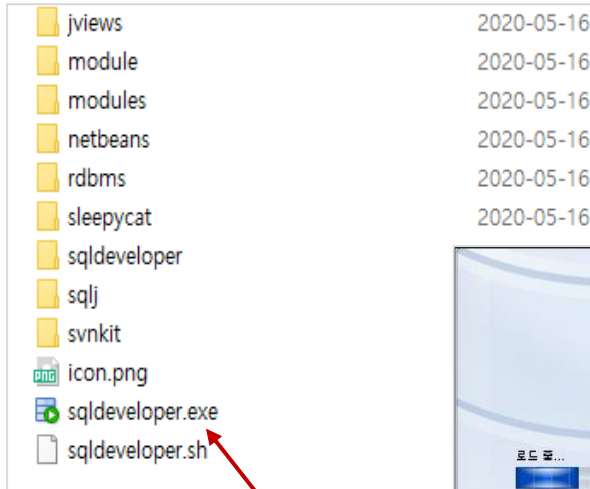
Windows 32-bit/64-bit

 [Download](#) (480 MB)

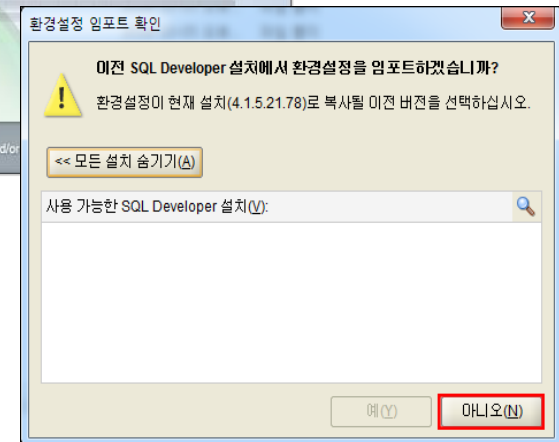


# 오라클 SQL 디벨로퍼

## ◆ Sqldeveloper 설치 – 다운로드후 압축풀기



실행

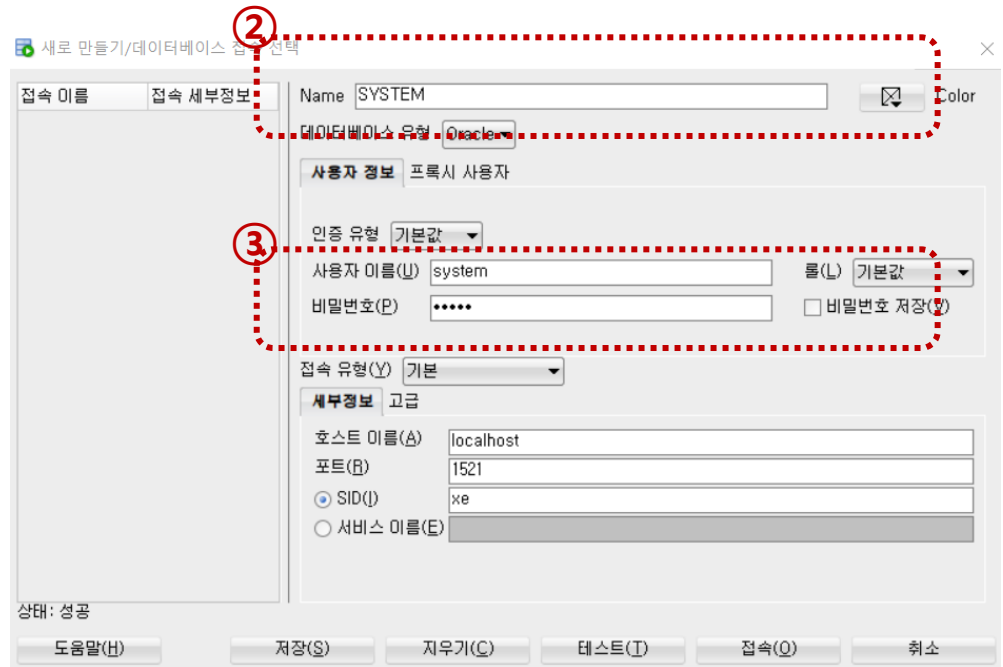
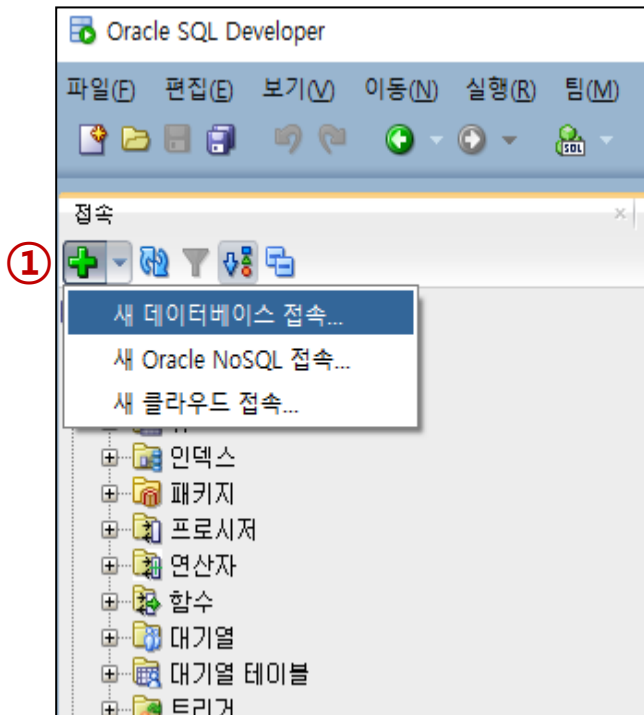


# 데이터베이스 생성

## ◆ 데이터베이스 만들기 및 접속

### 1. 관리자 (system) 계정만들기

Name-SYSTEM, 사용자이름- system, 비밀번호 - 12345

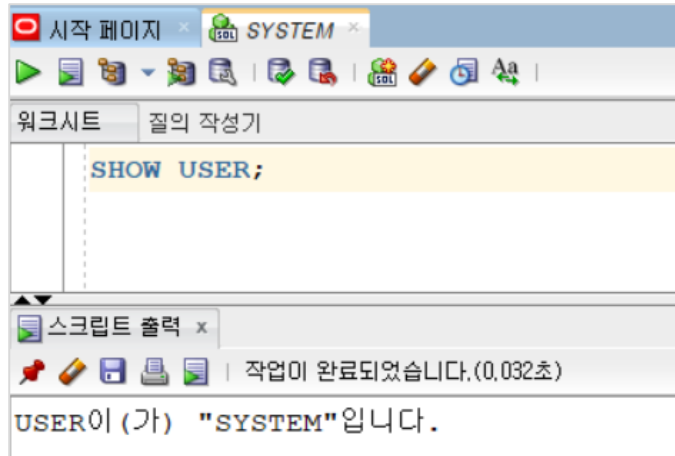


테스트후 성공  
하면 접속



# User 보기

## USER 보기



system 파일 저장하기 – user.sql





# 데이터베이스 구성 객체

## ❖ 데이터베이스 객체의 종류

DB 객체	설명
테이블	데이터를 담고 있는 객체
뷰	하나 이상의 테이블을 연결해 마치 테이블인 것처럼 사용하는 객체
인덱스	테이블에 있는 데이터를 빠르게 찾기 위한 객체
시노님	데이터베이스 객체에 대한 별칭을 부여한 객체
시퀀스	일련번호를 사용하는 객체
함수	특정 연산을 하고 값을 반환하는 객체
프로시저	함수와 비슷하지만 값을 반환하지 않는 객체



# sql 파일 만들기

## sql 파일 만들기

1. Oracle SQL Developer : D:\wdbworks\hdb\table.sql

파일(F) 편집(E) 보기(V) 이동(N) 실행(R) 소스

2. 새 갤러리

범주(C):

- General
  - Deployment Profiles
  - XML
  - 접속
- 데이터베이스 계층
  - 데이터베이스 객체
  - 데이터베이스 파일
- 모든 항목

항목(I):

- SQL 파일

SQL 파일 생성

새 SQL 스크립트를 생성한 다음 편집 및 실행을 위해 엽니다.

파일 이름(F):

제목 없음.sql

디렉토리(D):

C:\Users\kion\AppData\Roaming\SQL Developer

3. 확인

## 테이블(TABLE)

데이터를 삽입, 수정, 삭제하는 데이터를 담고 있는 객체가 테이블이다.

테이블은 DBMS상에서 가장 기본적인 객체로 로우(행)와 컬럼(열)으로 구성된 2차원 형태(표)의 객체로 엑셀과 구조가 비슷하다.

```
CREATE TABLE 테이블명(  
    컬럼1    데이터타입    [NULL, NOT NULL],  
    컬럼1    데이터타입    [NULL, NOT NULL],  
    ...  
);
```



# SQL – DDL

## DDL(Data Definition Language)

데이터를 저장할 테이블의 구조를 만드는 명령어이다.

명령어	설 명
CREATE	테이블을 생성함
ALTER	테이블의 구조를 변경함
DROP	테이블을 삭제함



# DATA TYPE

## 데이터 타입(Type)

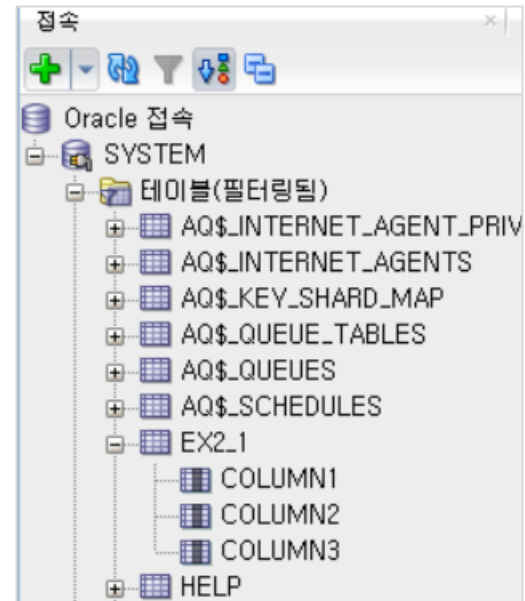
데이터 타입	설명
CHAR	고정길이 문자, 최대 2000byte, 디폴트값은 1byte
VARCHAR2	가변길이 문자, 최대 4000byte, 디폴트값은 1byte
NUMBER	가변 숫자, 십진수 기준 최대 220byte
FLOAT	NUMBER의 하위 타입, 이진수 기준 22byte
DATE	날짜 - 연, 월, 일
TIMESTAMP	날짜 - 연, 월, 일, 시, 분, 초, 밀리초
NULL	'값이 없음'을 의미하며 디폴트값이 NULL이다.



# DDL – CREATE

## 테이블 생성

```
-- 테이블 생성 --  
CREATE TABLE ex2_1(  
    column1    CHAR(10),  
    column2    VARCHAR2(10),  
    column3    NUMBER  
)
```



# SQL – DML

## DML(Data Manipulation Language)

데이터를 조작하는 명령어이다.

데이터를 조작하여 저장하는 일련의 과정을 **트랜잭션(transaction)**이라 하며,

DML은 트랜잭션을 다루는 명령어이다.

명령어	설 명
SELECT	테이블에 있는 행을 검색
INSERT	테이블에 새로운 행 삽입
UPDATE	테이블에 있는 행의 내용 갱신
DELETE	테이블의 행을 삭제
COMMIT	작업을 수행
ROLLBACK	작업 수행을 취소



# SQL – DML

INSERT 문 – 자료 삽입

```
INSERT INTO 테이블 이름(열이름1, 열이름2...)  
VALUES (데이터 값1, 데이터값 2, ...)
```

SELECT 문 – 자료 검색

```
SELECT 열이름 (or 별칭)  
FROM 테이블 이름
```





# SQL – DML

## 문자, 숫자 데이터 타입

```
-- 데이터 삽입 --  
-- 문자는 홀따옴표 가능, 쌍따옴표 불가능  
INSERT INTO ex2_1(column1, column2, column3) VALUES ('abc', 'abc', 10);  
INSERT INTO ex2_1(column1, column2, column3) VALUES ('우리', '우리', 11);  
--INSERT INTO ex2_1(column1, column2) VALUES ('대한민국', '대한민국', 12);  
  
-- 데이터 검색 --  
SELECT * FROM ex2_1;  
  
-- CHAR과 VARCHAR2의 크기 비교 --  
SELECT column1, LENGTH(column1) as len1,  
       column2, LENGTH(column2) as len2  
FROM ex2_1;
```

	COLUMN1	COLUMN2	COLUMN3
1	abc	abc	10
2	우리	우리	11

	COLUMN1	LEN1	COLUMN2	LEN2
1	abc	10	abc	3
2	우리	6	우리	2



# SQL – DML

## 날짜 데이터 타입

	COL_DATE	COL_TIMESTAMP
1	22/07/02	22/07/02 10:50:54.908000000

```
CREATE TABLE ex2_4 (  
    col_date    DATE,  
    col_timestamp TIMESTAMP  
)  
  
INSERT INTO ex2_4 VALUES (SYSDATE, SYSTIMESTAMP);
```



# 제약 조건

## 제약조건

테이블들은 각 속성(칼럼)에 대한 무결성을 유지하기 위한 다양한 제약 조건 (Constraints)이 적용되어 있다.

제약 조건에는 NOT NULL, 기본키, 외래키, CHECK 등이 있다.

## NOT NULL

칼럼을 정의할 때 NOT NULL 제약 조건을 명시하면 반드시 데이터를 입력해야한다.

칼럼명   데이터 타입   **NOT NULL**



# 제약 조건

```
3 CREATE TABLE ex2_5(  
    col_null    VARCHAR2(10),  
    col_not_null VARCHAR2(10) NOT NULL  
)  
  
INSERT INTO ex2_5 VALUES ('Hello', '');
```

오류 보고 -

ORA-01400: NULL을 ("SYSTEM"."EX2\_5"."COL\_NOT\_NULL") 안에 삽입할 수 없습니다



# 제약 조건

## 기본키

기본키는 Primary Key라고도 하며, UNIQUE와 NOT NULL 속성을 동시에 가진 제약 조건으로 테이블 당 1개의 기본키만 생성할 수 있다.

칼럼명 데이터 타입 **PRIMARY KEY**

또는

**CONSTRAINTS** 제약조건명 **PRIMARY KEY**(칼럼명, ...)



# 제약 조건

```
CREATE TABLE ex2_6(  
    id    VARCHAR2(10) PRIMARY KEY,  
    passwd VARCHAR2(10) NOT NULL  
);  
  
INSERT INTO ex2_6 VALUES ('sky2022', '1234mnop');  
INSERT INTO ex2_6 VALUES ('sky2022', '1234abcd');
```

오류 보고 -

ORA-00001: 무결성 제약 조건 (SYSTEM.SYS\_C008316) 에 위배됩니다



# DDL – ALTER

## 테이블 이름 변경

**ALTER TABLE** 테이블 이름 **RENAME TO** 새로운 테이블 이름

```
ALTER TABLE ex2_6 RENAME TO login;
```

## 칼럼 이름 변경

**ALTER TABLE** 테이블명 **RENAME COLUMN** 변경전칼럼명 **TO** 변경후칼럼명

```
ALTER TABLE login RENAME COLUMN id TO userId;
```



# DDL – ALTER

## 칼럼 추가

**ALTER TABLE** 테이블 이름 **ADD** 칼럼명 데이터 타입

```
ALTER TABLE login ADD name VARCHAR2(10);
```

## 칼럼 타입 변경

**ALTER TABLE** 테이블명 **MODIFY** 칼럼명 데이터타입

```
ALTER TABLE login MODIFY name VARCHAR2(20);
```





# DDL – ALTER

```
DESC login;
```

이름	널?	유형
-----		
USERID	NOT NULL	VARCHAR2 (10)
PASSWD	NOT NULL	VARCHAR2 (10)
NAME		VARCHAR2 (20)

```
INSERT INTO login VALUES ('river2022', '1234abcd', '이강산');
```

	USERID	PASSWD	NAME
1	sky2022	1234mnop	(null)
2	river2022	1234abcd	이강산



# DDL – ALTER

## 테이블 복사

```
CREATE TABLE 테이블 이름 AS  
SELECT 칼럼1. 칼럼2.. FROM 복사할 테이블명
```

```
-- 테이블 복사  
CREATE TABLE member AS  
SELECT * FROM login;
```



# DDL – DROP

## 테이블 삭제

테이블 및 모든 데이터를 삭제함.

CASCADE CONSTRAINTS 를 붙이면 삭제할 테이블의 기본키와 참조무결성 제약조건도 자동으로 삭제된다.

**DROP TABLE** 테이블명 **[CASCADE CONSTRAINTS]**

```
DROP TABLE login;
```



# 시퀀스 – SEQUENCE

## Sequence(시퀀스)

자동순번을 반환하는 데이터베이스 객체이다.

현재 사원테이블의 사번은 100부터 206까지인데 숫자형으로 되어 있다. 만약 신입사원이 들어온다면 206다음인 207번부터 생성될 것이다. 그런데 207이란 숫자를 얻으려면 기존 사원번호 중 최대값을 구해 1을 더해야 한다.

```
CREATE SEQUENCE mySeq  
INCREMENT BY 1  
START WITH 1  
MINVALUE 1  
MAXVALUE 1000  
NOCYCLE  
NOCACHE;
```

mySeq는 1부터 시작해  
1씩 증가하며  
최소값 1부터 최댓값 1000까지  
순번을 자동 생성함



# 시퀀스 – SEQUENCE

## Sequence(시퀀스)

```
-- 시퀀스 생성
CREATE SEQUENCE mySeq
INCREMENT BY 1
START WITH 1
MINVALUE 1
MAXVALUE 1000
NOCYCLE
NOCACHE;

-- 이 시퀀스는 1부터 시작해 1씩 증가하며 최소값 1부터 최댓값 1000까지
-- 순번을 자동 생성한다.
-- 이 시퀀스로 board 테이블에 사용
```



# 시퀀스 - SEQUENCE

## BOARD 테이블에 적용

```
-- BOARD 테이블 생성
CREATE TABLE board(
    bno    NUMBER(4) PRIMARY KEY,
    title  VARCHAR2(50) NOT NULL,
    content VARCHAR2(2000) NOT NULL,
    regDate DATE DEFAULT SYSDATE
)

INSERT INTO board (bno, title, content, regDate)
VALUES (mySeq.NEXTVAL, '제목1', '제목1 내용입니다.', SYSDATE);

INSERT INTO board (bno, title, content, regDate)
VALUES (mySeq.NEXTVAL, '제목2', '제목2 내용입니다.', SYSDATE);

SELECT * FROM board;
```

BNO	TITLE	CONTENT	REGDATE
1	제목1	제목1 내용입니다.	22/07/02
2	제목2	제목2 내용입니다.	22/07/02



# 사용자 DB 만들기

## SYSTEM에서 새 데이터베이스 사용자 만들기

```
-- USER(jweb) 생성  
CREATE USER C##jweb IDENTIFIED BY 54321;  
  
-- 세션 권한 부여  
GRANT CREATE SESSION TO C##jweb;  
  
-- 테이블과 테이블 공간 생성  
GRANT CREATE TABLE, RESOURCE TO C##jweb;
```



새로 만들기/데이터베이스 접속 선택

접속 이름	접속 세부정보
JWEBDB	C##JWEB@//l...
SYSTEM	system@//loc...

Name: JWEBDB

데이터베이스 유형: Oracle

사용자 정보: 프록시 사용자

인증 유형: 기본값

사용자 이름(U): C##JWEB

비밀번호(P): .....

접속 유형(Y): 기본

세부정보: 고급

호스트 이름(A): localhost

포트(P): 1521

☒ SID(I): xe

☐ 서비스 이름(E):

상태: 성공

도움말(H)    저장(S)    지우기(C)    테스트(T)    접속(Q)    취소



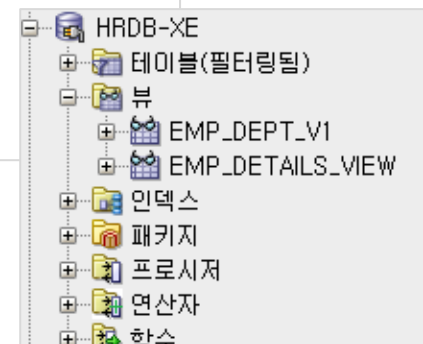
# 데이터베이스 구성 객체

## 뷰(VIEW)

뷰는 하나 이상의 테이블이나 다른 뷰의 데이터를 볼 수 있게 하는 데이터베이스 객체다. 테이블이 아닌 뷰를 사용하는 이유는 원본 테이블의 데이터를 안전하게 유지하면서 필요한 사용자에게 적절한 데이터를 제공할 수 있다.

```
-- 해당 사원이 속한 부서명 검색 --  
SELECT a.employee_id, a.last_name, a.department_id,  
       b.department_name  
FROM employees a,  
     departments b  
WHERE a.department_id = b.department_id;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
100	King	90	Executive
101	Kochhar	90	Executive
102	De Haan	90	Executive
103	Hunold	60	IT
104	Ernst	60	IT
105	Austin	60	IT





# 데이터베이스 구성 객체

## 뷰(VIEW) 생성 및 삭제

CREATE **VIEW** 뷰이름 AS SELECT 문장;

```
-- VIEW 생성
CREATE VIEW emp_dept_v1 AS
SELECT a.employee_id, a.last_name, a.department_id,
       b.department_name
FROM employees a,
     departments b
WHERE a.department_id = b.department_id;
```

**DROP VIEW** 뷰이름

뷰(VIEW) 검색

```
21 SELECT *
22 FROM emp_dept_v1;
```

SQL | 50개의 행이 인출됨(0.005초)

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	100	King	90	Executive
2	101	Kochhar	90	Executive
3	102	De Haan	90	Executive
4	103	Hunold	60	IT
5	104	Ernst	60	IT

# 데이터베이스 구성 객체

## 뷰(VIEW) 검색

```
--view 검색
SELECT * FROM emp_details_view;

-- employees 테이블과 emp_details_view를 조인하여 employee_id가 100인 직원의
-- employee_id, hire_date, department_name, job_title 출력
SELECT A.employee_id, A.hire_date, B.department_name, B.job_title
FROM employees A, emp_details_view B
WHERE A.employee_id = B.employee_id
AND A.employee_id = 100;
```

EMPLOYEE_ID	HIRE_DATE	DEPARTMENT_NAME	JOB_TITLE
100	2003/06/17	Executive	President



# 시노님(SYNONYM) 권한 부여

- MINA(데이터베이스)가 TAMI의 STARBUCKS\_ORDER 테이블 사용하기

①

```
GRANT_SELECT_ONI.sql x SYNONYM.sql x GRANT_SYNONYM.sql x
워크시트 | 질의 작성기
1 -- SYNONIM (동의어) 권한 주기
2 GRANT CREATE SYNONYM TO MINA;
```

SYSTEM에서 SYNONYMN 권한부여

②

```
GRANT_SELECT_ONI.sql x SYNONYM.sql x GRANT_SYNONYM.sql x
워크시트 | 질의 작성기
1 SELECT * FROM STARBUCKS_ORDER;
2
3 -- MINA에게 테이블 검색 권한 부여 --
4 GRANT SELECT ON STARBUCKS_ORDER TO MINA;
```

질의 결과 x 스크립트 출력 x

작업이 완료되었습니다. (0.082초)

Grant을(를) 성공했습니다.

TAMI가 MINA에게 검색권한 부여



# 시노님(SYNONYM)

③

```
GRANT_SELECT_ONI.sql  SYNONYM.sql  GRANT_SYNONYM.sql
-- SYNONYM 권한부여 전 --
SELECT * FROM TAMI.STARBUCKS_ORDER;

-- SYSTEM에서 SYNONYM 권한을 부여한 후 생성 --
CREATE SYNONYM STARBUCKS_ORDER FOR TAMI.STARBUCKS_ORDER;

-- SYNONYM 권한부여 후 --
SELECT * FROM STARBUCKS_ORDER;
```

스크립트 출력 x 질의 결과 x

작업이 완료되었습니다.(0.053초)

Synonym STARBUCKS\_ORDER이(가) 생성되었습니다.

MINA가 TAMI의 STARBUCKS\_ORDER와 같은 이름으로 검색 가능

ORDER_NO	ORDER_DT	BRANCH	ORDER_ITEM	REG_NAME	REG_DTS
1	20190801	강남점	아메리카노	릴리	20/08/30
2	20190801	강남점	카페라떼	봄	20/08/30
3	20190801	강남점	자바칩 프라푸치노	테스	20/08/30
4	20190801	강남점	바닐라 프라푸치노	릴리	20/08/30
5	20190801	강남점	아메리카노	무드	20/08/30
6	20190801	강남점	아메리카노	봄	20/08/30
7	20190801	강남점	쿨라임 피지오	봄	20/08/30
8	20190801	강남점	카페라떼	테스	20/08/30
9	20190802	강남점	아메리카노	무드	20/08/30
10	20190802	강남점	콜드브루	릴리	20/08/30
11	20190802	강남점	아메리카노	릴리	20/08/30
12	20190802	강남점	아메리카노	마샤	20/08/30
13	20190802	강남점	한라봉주스	봄	20/08/30
14	20190802	강남점	카페모카	테스	20/08/30
15	20190802	강남점	아메리카노	릴리	20/08/30