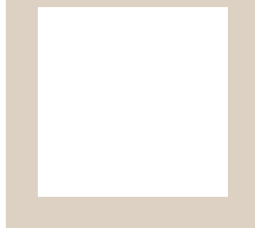


# 데이터베이스시스템



## 09. 물리적 데이터베이스 설계



나 홍 석    교수



고려사이버대학교  
THE CYBER UNIVERSITY OF KOREA



9  
LESSON

# 물리적 데이터베이스 설계

# 학습 목표

- 1 물리적 데이터 모델링 과정을 설명할 수 있다.
- 2 데이터 저장구조와 제약조건을 기반으로 테이블 명세서를 작성할 수 있다.

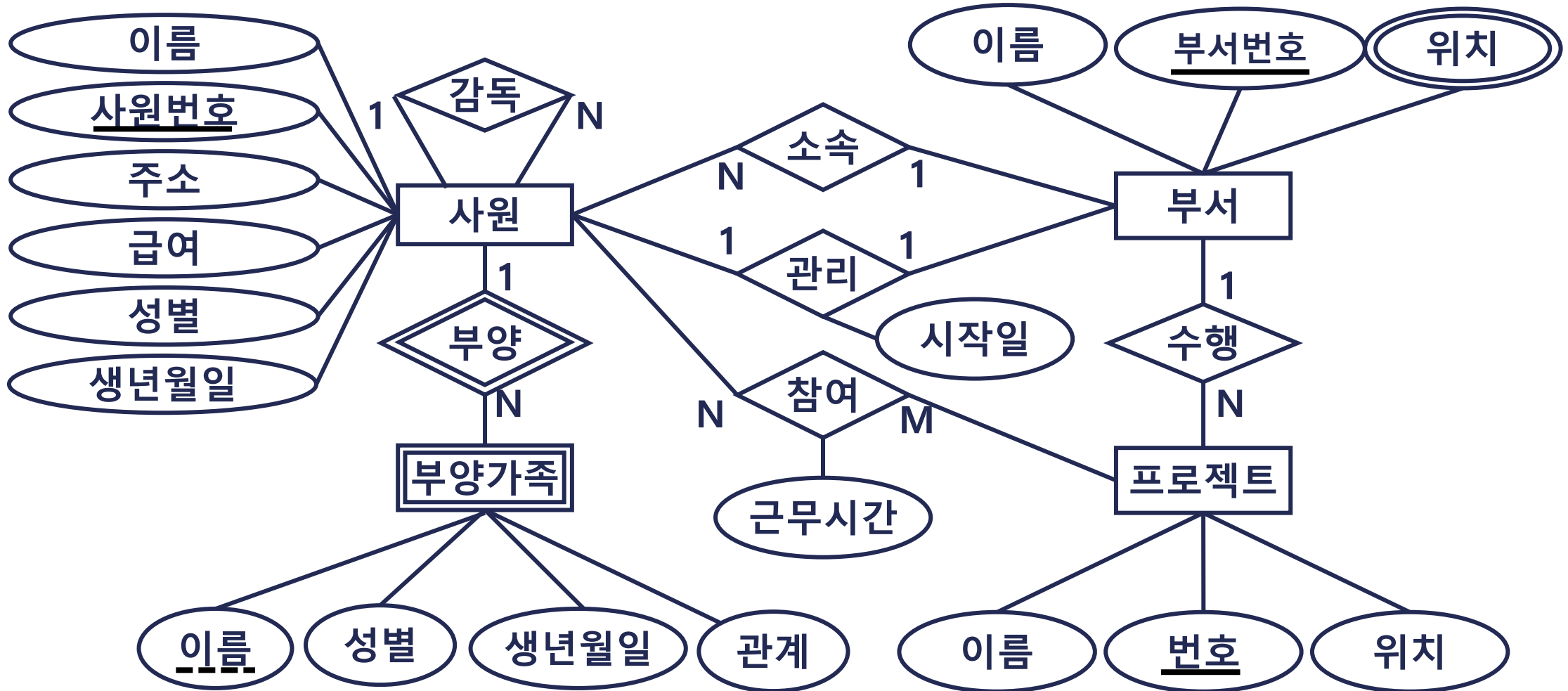
# 학습 내용

- 1 릴레이션 변환 리뷰
- 2 물리적 데이터 모델링
- 3 저장구조 / 제약조건 / 접근경로

# Chapter 01 릴레이션 변환 리뷰

# 1 개체의 릴레이션 변환

## 1 사원관리 시스템 ERD



# 1 개체의 릴레이션 변환

## 2 사원개체, 부서개체, 프로젝트 개체

**사원**

<u>사원번호</u>	이름	주소	급여	성별	생년월일
-------------	----	----	----	----	------

**부서**

<u>부서번호</u>	부서이름
-------------	------

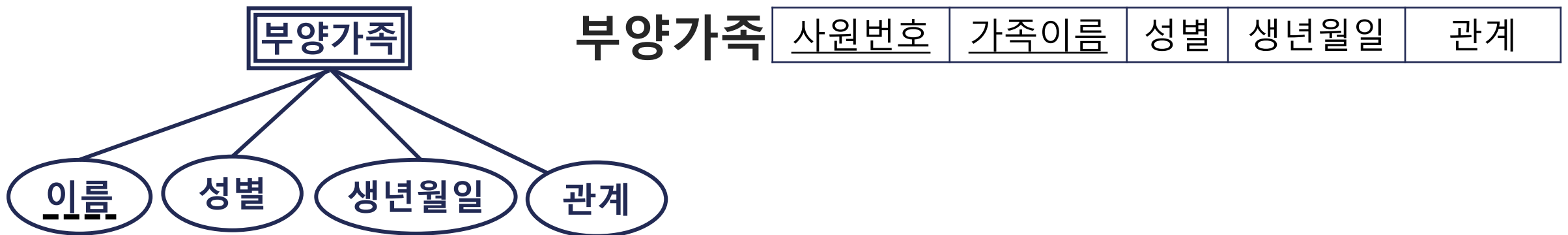
**프로젝트**

<u>프로젝트번호</u>	프로젝트이름	위치
---------------	--------	----

# 1 개체의 릴레이션 변환

## 3 약개체의 릴레이션 변환

- ☑ 약개체의 경우 기본키에 해당하는 속성(집합)이 없음
- ☑ 약개체가 기준으로 하는 개체에서 필요한 속성을 빌려와야 함
- ☑ 기준 개체의 기본키와 약개체의 대표 속성을 합치고, 이를 기본키로 하는 릴레이션 생성





## 2 관계의 릴레이션 변환

### 1 관계의 릴레이션 변환방법

- ☑ 개체와 개체 사이에는 한 개 이상의 관계가 존재할 수 있음
- ☑ N:M 관계의 경우 독립된 릴레이션을 생성하면 됨
- ☑ 1:N 관계의 경우는 N쪽의 릴레이션에 각각의 관계를 표현함
- ☑ 즉, 각각의 관계에서 따라 각각의 외래키를 포함해서 릴레이션 생성
- ☑ 순환 관계에 참여하는 개체의 기본키를 바로 그 개체의 외래키로 넣어서 릴레이션 완성



## 2 관계의 릴레이션 변환

2 소속, 관리, 참여, 수행

사원

<u>사원번호</u>	이름	주소	급여	성별	생년월일	부서번호(FK)
-------------	----	----	----	----	------	----------



부서

<u>부서번호</u>	부서이름	부서관리자(FK)	시작일
-------------	------	-----------	-----



프로젝트

<u>프로젝트번호</u>	프로젝트이름	위치	수행부서(FK)
---------------	--------	----	----------



참여

<u>사원번호(FK)</u>	<u>프로젝트번호(FK)</u>	근무시간
-----------------	-------------------	------



부양가족

<u>사원번호</u>	<u>가족이름</u>	성별	생년월일	관계
-------------	-------------	----	------	----

## 2 관계의 릴레이션 변환

### 3 감독, 부양



사원	<u>사원번호</u>	이름	주소	급여	성별	생년월일	감독자(FK)	부서번호(FK)

부서	부서번호	부서이름	부서관리자(FK)	시작일

프로젝트	<u>프로젝트번호</u>	프로젝트이름	위치	수행부서(FK)

참여	<u>사원번호(FK)</u>	<u>프로젝트번호(FK)</u>	근무시간

부양가족	<u>사원번호(FK)</u>	가족이름	성별	생년월일	관계



### 3 속성의 릴레이션 변환

#### 1 속성의 릴레이션 변환방법

- ☑ 다중값 속성은 하나의 속성에 여러 개의 값이 들어간다는 것을 표현
- ☑ 해당 개체의 기본키와 다중값 속성을 합쳐서 복합속성을 만들고  
그 복합 속성을 기본키로 하는 별도의 릴레이션을 생성



부서위치

부서번호(FK)	위치
----------	----

### 3 속성의 릴레이션 변환

#### 2 사원관리 시스템의 릴레이션 스키마

**사원**

<u>사원번호</u>	이름	주소	급여	성별	생년월일	감독자(FK)	부서번호(FK)
-------------	----	----	----	----	------	---------	----------

**부서**

<u>부서번호</u>	부서이름	부서관리자(FK)	시작일
-------------	------	-----------	-----

**부서위치**

<u>부서번호(FK)</u>	<u>위치</u>
-----------------	-----------

**프로젝트**

<u>프로젝트번호</u>	프로젝트이름	위치	수행부서(FK)
---------------	--------	----	----------

**참여**

<u>사원번호(FK)</u>	<u>프로젝트번호(FK)</u>	근무시간
-----------------	-------------------	------

**부양가족**

<u>사원번호</u>	<u>가족이름</u>	성별	생년월일	관계
-------------	-------------	----	------	----

## Chapter 02 물리적 데이터 모델링

# 1 모델링 과정 리뷰

## 1 데이터 모델링 단계

구 분	설 명
개념적 모델링	<ul style="list-style-type: none"><li>• 업무의 대상이 되는 실제 데이터들에 대해서, 서로의 상관관계를 파악하는 작업</li><li>• 개체, 관계, 속성을 파악하고 이들간의 관련성을 도식화</li><li>• 결과물 : ERD(Entity Relationship Diagram)</li></ul>
논리적 모델링	<ul style="list-style-type: none"><li>• DBMS의 유형(관계형, 객체지향형, 객체-관계형)에 맞추어 DBMS에 저장될 데이터의 골격(스키마)를 만드는 작업</li><li>• 결과물 : 함수종속성 파악, 논리적스키마(릴레이션구조)</li></ul>
물리적 모델링	<ul style="list-style-type: none"><li>• 특정 DBMS에 의존하는 데이터형식, 각종 제약조건, 뷰, 인덱스 등을 설정하는 작업</li><li>• 결과물 : 테이블 정의서, 제약조건 리스트, 인덱스 명세서 등</li></ul>

# 1 모델링 과정 리뷰

## 2 개념적 데이터 모델링

### 개체간의 관계를 정확히 표현하는 단계

- 요구분석 명세로부터 개체, 속성, 관계를 식별해서 결정
- 특히, 개념적 설계에서는 개체와 개체 간의  
연관성(관계:relationship)에 중점을 둠

### 개념적 데이터 모델(개체-관계도: Entity-Relationship Diagram)로 기술

- 개체, 관계, 속성으로 구성
- 이전단계에서 도출된 요구명세와의 상응성을 점검



# 1 모델링 과정 리뷰

## 3 논리적 데이터 모델링

- ☑ 개념적 설계로 만들어진 개념적 구조로부터 특정 목표 DBMS가 처리할 수 있는 스키마를 생성함
- ☑ 이 스키마는 요구조건 명세를 만족해야 하며, 무결성이나 일관성 제약조건도 만족해야 함
- ☑ 관계형 데이터베이스의 경우 논리적 설계 단계에서 함수적 종속성에 대한 분석과 정규화가 이루어짐
- ☑ 논리적 설계 단계부터 DBMS의 특성에 많이 의존됨

## 2 물리적 데이터 모델

### 1

#### 개요

A **physical data model (or database design)** is a representation of a data design which takes into account the **facilities and constraints of a given database management system.**

In the lifecycle of a project, it typically derives from a logical data model, though it may be reverse-engineered from a given database implementation.

- Wikipedia -

## 2 물리적 데이터 모델

### 2

#### 목적

효율적, 구현 가능한 물리적 DB 구조 설계

- ☑ 데이터의 저장 구조나 접근 경로에 대해서 DBMS가 지원하는 방법 중에서 선택함
- ☑ 응답시간, 저장공간의 효율화, 트랜잭션 처리도(Throughput) 등을 고려해야 함

## 2 물리적 데이터 모델

### 3 수행 작업

#### 저장구조 설계

- 데이터베이스 구조, 테이블 구조, 테이블 분할, 이름 영문화

#### 제약조건 지정

- 데이터 형 지정, 기본키 및 기본값 정의, 체크와 규칙 정의

#### 레코드 집중의 분석 및 설계

- 레코드 크기와 물리적 저장 장치의 특성에 의존
- 테이블 분산, 파티셔닝 등

#### 접근경로설계

- 인덱스 설정, 뷰 정의

## 2 물리적 데이터 모델

### 4 용어 정리 - 관계형 DBMS

- ☑ 테이블은 데이터 저장을 위한 가장 기본적인 단위로 논리모델의 릴레이션에 대응
- ☑ 컬럼은 테이블의 열, 로우는 테이블의 행에 해당

**컬럼(Columns)**

**테이블 이름**  
**Student**

**로우(Rows)**

St_ID	St_Name	St_Sex	St_Addr	St_Phn_Mbl	St_Birth	Dept_ID
ST001	최현주	여	서울	010-1234-1234	1973	SE
ST002	강하늘	남	서울	010-2222-2344	1990	BZ
ST003	이성민	남	서울	010-3293-9345	1978	SE
ST004	박정수	여	경기	010-8323-8342	2000	EE
ST005	홍민호	남	대전	010-2342-6547	1985	BZ

## 2 물리적 데이터 모델

### 5 테이블명 및 컬럼명의 영문화 #1

- ☑ DBMS로 구현하고 프로그래밍 환경에 적용하기 위해서는  
영문 테이블명과 컬럼명이 필요
- ☑ 논리적 모델에서 사용된 릴레이션과 애트리뷰트를 영문으로  
전환하는 과정이 필요
- ☑ **표준 용어집**을 작성하거나 사전에 정의된 **명명규칙**에 맞추어서 변환

## 2 물리적 데이터 모델

### 5 테이블명 및 컬럼명의 영문화 #2



표준 용어집의 예

No	한글단어	영문 전체 이름	영문약어
1	학생	Student	St
2	학과	Department	Dept
3	이름	Name	Nm
4	주소	Address	Addr
5	전화번호	Phone Number	Phn
6	집	Home	Hm
7	이동(전화)	Mobile	Mbl
8	번호	Number	No
9	생년월일	Birthday	Birth
10	아이디	Identifier	ID

## 2 물리적 데이터 모델

### 6 모델링 결과 #1

물리적  
데이터베이스  
스키마

- 주변 환경 정보
  - 하드웨어 자원 현황, 운영체제 현황, DBMS 버전 및 파라미터 정보 파악
- 데이터베이스 운영 정보
  - 사용자 관리 정책, 백업/복구 기법 및 정책, 보안관리 정책
- 테이블 정의서, 컬럼 정의서
- 표준 용어집
- 뷰 및 인덱스 정의



## 2 물리적 데이터 모델

### 6 모델링 결과 #2

#### 테이블 정의서

테이블 정의서											
프로젝트	K사이버대학교		시스템명	학사관리시스템		작성일자	2021. 3. 1		작성자	김정식	
테이블명	Student										
테이블정의	K사이버대학교 재학생/휴학생 개인신상 정보										
No	컬럼명	한글명	타입	길이	NULL	UK	PK	FK	참조테이블	참조컬럼	비고
1	St_Name	이름	varchar	20							
2	St_ID	아이디	varchar	10			Y				
3	St_Sex	성별	char	1	Y						F, M
4	St_Phn_Mbl	전화번호	char	13	Y						999-9999-9999
5	St_Addr	주소	varchar	50	Y						
6	St_Birth	생년월일	datetime		Y	Y					1900년 이후 출생
7	Dept_ID	소속학과	char	2				Y	Department	Dept_ID	IT, BZ, MD, ...

## Chapter 03 저장구조 / 제약조건 / 접근경로

# 1 저장구조 설계

## 1 테이블 구조 정의 #1

- ☑ 릴레이션으로부터 테이블명과 컬럼명을 도출하여 각각의 테이블 구조를 정의함
- ☑ 관리상 필요한 컬럼이 추가되기도 함



예) 해당 데이터를 등록한 날짜, 시스템  
번호 등



# 1 저장구조 설계

## 2 데이터타입 #1 - 개요

- ☑ DBMS에 맞는 데이터타입 선택
- ☑ 논리적인 모델의 데이터 타입을 물리적인 DBMS의 특성과 성능을 고려하여 **최적의 데이터 타입을 선택**
- ☑ 크게 **문자(열) 타입, 숫자 타입, 날짜 타입** 등으로 분류되며, DBMS에 따라 세부적으로 더 많은 종류의 타입을 지원

# 1 저장구조 설계

## 2 데이터타입 #2 - 문자열형 데이터 타입

✓ CHAR(n): 고정 길이 데이터 타입

✓ VARCHAR(n): 가변 길이 데이터 타입

ORACLE		MS SQL Server		MySQL, MariaDB	
데이터타입	설명	데이터타입	크기	데이터타입	크기
CHAR(n)	고정길이 문자열 / 최대 2000bytes	char(n)	1 to 8000	CHAR(n)	최대 255자
NCHAR(n)	고정길이 유니코드문자열 / 최대 2000bytes	nchar(n)	1 to 4000		
VARCHAR2(n)	가변길이 문자열 / 최대 4000bytes	varchar(n max)	1 to 8000 max는 2G	VARCHAR(n)	최대 255자
NVARCHAR2(n)	가변길이 유니코드 문자열 / 최대 4000bytes	nvarchar(n max)	1 to 4000 max는 2G		
LONG	가변길이 문자열 최대 / 2Gbytes			LONGTEXT	
CLOB	대용량 텍스트 데이터 타입 / 최대 4Gbytes	text, varchar(max)	2G	TEXT	최대 65535자
NCLOB	대용량 텍스트 유니코드 데이터 타입 / 최대 4Gbytes	ntext, nvarchar(max)	2G		

# 1 저장구조 설계

## 2 데이터타입 #3 - 숫자 데이터 타입

☑ 오라클 → NUMBER(p,s): 가변 숫자 / 최대 22bytes

☑ p: 소수점을 포함한 전체 자리수, s: 소수점 자리수

입력값	타입	저장되는 값
123.89	NUMBER	123.89
123.89	NUMBER(3)	124
123.89	NUMBER(3,2)	Exceeds precision
123.89	NUMBER(4,2)	Exceeds precision
123.89	NUMBER(5,2)	123.89
123.89	NUMBER(6,1)	123.9

MS SQL Server		MySQL, MariaDB	
데이터타입	크기	데이터타입	크기
<b>bigint</b>	8 bytes	<b>BIGINT</b>	8 bytes
<b>int</b>	4 bytes	<b>INT</b>	4 bytes
<b>smallint</b>	2 bytes	<b>SMALLINT</b>	2 bytes
<b>tinyInt</b>	1 bytes	<b>TINYINT</b>	1 bytes
<b>decimal(p,s)</b> <b>numeric(p,s)</b>	고정소수점	<b>DECIMAL</b>	고정소수점
<b>float</b> <b>real</b>	부동소수점	<b>FLOAT</b> <b>DOUBLE</b>	부동소수점

# 1 저장구조 설계

## 2 데이터타입 #4 – 날짜 데이터 타입

- ☑ DATE: 연, 월, 일, 시, 분, 초 까지 입력 가능(오라클)
- ☑ TIMESTAMP: 연도, 월, 일, 시, 분, 초 + 밀리초까지 입력가능 (오라클)

Oracle	SQL Server	MySQL, MariaDB
DATE	datetime	DATETIME
	date	DATE
	smalldatetime	
	time	TIME
TIMESTAMP	datetime2 (DT2)	



# 1 저장구조 설계

## 3 Student 테이블 데이터 타입 정의

- ☑ 전화번호: 계산에 쓰이는 컬럼이 아니므로 고정길이 문자열로 정의함
- ☑ 소속학과: 학과코드(2바이트)가 들어감
- ☑ 날짜: 대부분 일반적인 날짜 형식을 사용함

# 1 저장구조 설계

## 4 Student 테이블 데이터타입 정의 결과

테이블명		Student									
테이블정의		K사이버대학교 재학생/휴학생 개인신상 정보									
No	컬럼명	한글명	타입	길이	NULL	U K	PK	FK	참조테이블	참조컬럼	비고
1	St_Name	이름	varchar	20							
2	St_ID	아이디	varchar	10							
3	St_Sex	성별	char	1							
4	St_Phn_Mbl	전화번호	char	13							
5	St_Addr	주소	varchar	50							
6	St_Birth	생년월일	datetime								
7	Dept_ID	소속학과	char	2							

## 2 제약조건 지정

### 1 개체무결성 제약조건 #1

#### 기본키 / 유일성 제약(Unique)

- 논리 모델의 기본키는 물리 모델의 기본키(Primary Key)로 대응
- 기본키는 개체무결성 제약조건을 만족시키기 위한 조건  
(Unique + Not Null)
- 기본키를 제외한 나머지 후보키 들은 기본적으로 Unique 속성을 가짐
- Not Null 속성을 갖는 지 여부를 다시 한번 검토

## 2 제약조건 지정

### 1 개체무결성 제약조건 #2

#### 자연키(Natural PK)와 인조키(Artificial PK)

- 기본키가 여러 개의 속성으로 구성되거나 크기가 큰 경우 **인조키**를 인위적으로 생성함
- 자연키는 쉽게 이해할 수 있으며, 인조키는 쉽게 만들고 사용할 수 있음
- 기본키를 구성하는 속성들이 기본키의 조건을 만족시킨다는 확신이 없을 경우 인조키를 만들어서 사용하는 것이 좋음

#### **예** 자연키의 예

주문	<u>고객번호</u>	<u>주문일자</u>	상품	개수
----	-------------	-------------	----	----

#### **예** 인조키의 예

주문	<u>주문번호</u>	고객번호	주문일자	상품	개수
----	-------------	------	------	----	----

## 2 제약조건 지정

### 1 개체무결성 제약조건 #3



#### NULL과 NOT NULL

- NULL 값은 매우 주의해서 사용해야 함
- 어떤 컬럼의 값이 생략이 가능하다고 해서 NULL을 허용하는 것이 아니라 **알지 못하는 값을 표현**할 필요가 있는 경우에 사용
- 단순히 값이 생략되어도 좋은 경우라면, **공백, 0, -1 등의 값을 사용**해서 표현하는 것이 좋음
- 특히, **인덱스로 지정이 되는 컬럼**에서는 NULL 값을 허용하지 않는 것을 권장

## 2 제약조건 지정

2

### 참조무결성 제약조건



#### 외래키 지정

- 1:M 관계에서 1(One)에 있는 PK를 M(Many)의 FK로 변환
- 1:1 관계에서는 전체가 참여하는 쪽(Mandatory)에 상대방의 PK를 FK로 생성
- N:M 관계는 관계를 표현하는 테이블을 생성 후 이 테이블에 FK를 생성

## 2 제약조건 지정

### 3 도메인 무결성 제약조건

#### 체크(Check) 제약조건

- 테이블의 값이 **올바른 값만 입력**되도록 제약조건을 두는 것이 체크 제약 조건
- 도메인 무결성을 유지하기 위해서 필요, 허용 양식과 허용 값을 검사할 수 있음
- 일반적으로, 컬럼 단위로 제약조건을 주지만, 여러 컬럼에 걸쳐 제약조건을 줄 수 있음

**예** 전화번호는 숫자(문자)로만 구성되어야 하며, 3자리-4자리-4자리를 갖는다.  
“999-9999-9999”

## 2 제약조건 지정

### 4

#### 기본값 제약조건



#### 기본값(Default Value)

- 데이터 삽입 시 값을 생략하면 **미리 지정되어 있는 기본값이 대신 삽입**하도록 지정
- 예를 들어, 학과 컬럼에 기본값으로 'GE'를 지정하면, 학생의 정보가 입력될 때 학과에 관한 값이 없으면 자동으로 'GE' 라는 값이 들어감
- 값의 생략은 허용하면서, NULL 값은 허용하지 않는 경우에 유용하게 사용됨



## 2 제약조건 지정

### 5 제약조건 지정 결과 #1



### Student 테이블 정의

테이블명		Student									
테이블정의		K사이버대학교 재학생/휴학생 개인신상 정보									
No	컬럼명	한글명	타입	길이	NULL	UK	PK	FK	참조테이블	참조컬럼	비고
1	St_Name	이름	varchar	20							
2	St_ID	아이디	varchar	10			Y				
3	St_Sex	성별	char	1	Y						F, M
4	St_Phn_Mbl	전화번호	char	13	Y	Y					999-9999-9999
5	St_Addr	주소	varchar	50	Y						
6	St_Birth	생년월일	Datetime		Y						1900년 이후 출생
7	Dept_ID	소속학과	char	2				Y	Department	Dept_ID	IT, BZ, MD, ...

## 2 제약조건 지정

### 5 제약조건 지정 결과 #2

#### Course 테이블 정의

테이블명		Course									
테이블정의		K사이버대학교 과목 정보 테이블									
No	컬럼명	한글명	타입	길이	NULL	UK	PK	FK	참조테이블	참조컬럼	비고
1	Co_Num	과목번호	char	5			Y				
2	Co_Name	과목이름	varchar	20		Y					
3	Co_Location	과목장소	varchar	10	Y						

## 2 제약조건 지정

### 5 제약조건 지정 결과 #3

#### Enrol 테이블 정의

테이블명		Enrol									
테이블정의		K사이버대학교 등록 정보 테이블									
No	컬럼명	한글명	타입	길이	NULL	UK	PK	FK	참조테이블	참조컬럼	비고
1	St_ID	학생번호	char	6			Y	Y	Student	St_ID	
2	Co_Num	과목번호	char	5			Y	Y	Course	Co_Num	
3	Grade	학점	char	2	Y						A, B, C, D, F Default 'U'
4	Mid	중간성적	int		Y						0이상,100이하
5	Final	기말성적	int		Y						0이상,100이하

# 3 접근경로 설정

## 1 인덱스(Index)

- ☑ 데이터에 대한 논리적 포인터의 집합
- ☑ 책의 찾아보기와 같은 역할
- ☑ 질의문의 빠른 수행과 컬럼값의 유일성을 보장하기 위해서 사용

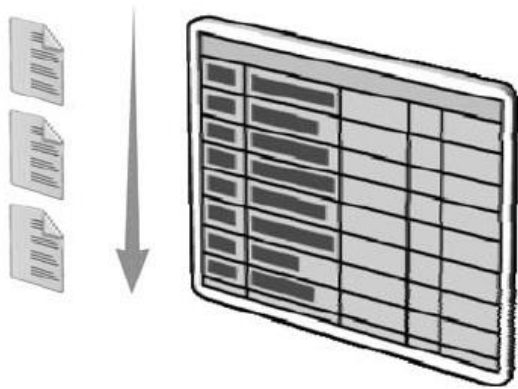
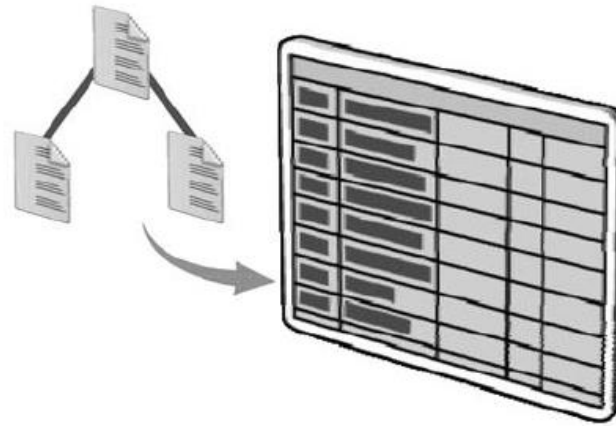


TABLE SCAN



INDEX SCAN

# 3 접근경로 설정

## 2 인덱스 종류

- ☑ 단일(Single) 칼럼 인덱스와 결합(Composite) 인덱스
- ☑ Unique 인덱스와 Non-Unique 인덱스
- ☑ 함수기반(Function-based) 인덱스
- ☑ 물리적 구현방식: B-tree 구조 인덱스, 비트맵 인덱스, 클러스터링 인덱스

# 3 접근경로 설정

## 3 인덱스 장단점

### 인덱스의 장점

- 빠르게 데이터를 찾아낼 수 있음
- 유일성 인덱스로 만들면  
유일성 제약 조건도 강화됨



### 인덱스의 단점

- 인덱스 자체가 추가적인 공간을 차지함
- 인덱스를 유지 관리하는 데  
추가적인 시간이 소비됨
- 데이터를 추가하고 수정할 때는  
인덱스 때문에 시간이 더 걸림

# 학습 정리

## 데이터 모델링 단계

구 분	설 명
개념적 모델링	<ul style="list-style-type: none"> <li>• 업무의 대상이 되는 실제 데이터들에 대해서, 서로의 상관관계를 파악하는 작업</li> <li>• 개체, 관계, 속성을 파악하고 이들간의 관련성을 도식화</li> <li>• 결과물 : ERD(Entity Relationship Diagram)</li> </ul>
논리적 모델링	<ul style="list-style-type: none"> <li>• DBMS의 유형(관계형, 객체지향형, 객체-관계형)에 맞추어 DBMS에 저장될 데이터의 골격(스키마)를 만드는 작업</li> <li>• 결과물 : 함수종속성 파악, 논리적스키마(릴레이션구조)</li> </ul>
물리적 모델링	<ul style="list-style-type: none"> <li>• 특정 DBMS에 의존하는 데이터형식, 각종 제약조건, 뷰, 인덱스 등을 설정하는 작업</li> <li>• 결과물 : 테이블 정의서, 제약조건 리스트, 인덱스 명세서 등</li> </ul>

# 학습 정리



## 데이터베이스 물리적 설계

- 효율적, 구현 가능한 물리적 DB 구조 설계 계
  - 데이터의 저장 구조나 접근 경로에 대해서 DBMS가 지원하는 방법 중에서 선택함
  - 응답시간, 저장공간의 효율화, 트랜잭션 처리도(Throughput) 등을 고려해야 함



# 학습 정리



## 물리적 설계단계 할 일

### 저장구조 설계

- 데이터베이스 구조, 테이블 구조, 테이블 분할, 이름 영문화

### 제약조건 지정

- 데이터 형 지정, 기본키 및 기본값 정의, 체크와 규칙 정의

### 레코드 집종의 분석 및 설계

- 레코드 크기와 물리적 저장 장치의 특성에 의존
- 테이블 분산, 파티셔닝 등

### 접근경로설계

- 인덱스 설정, 뷰 정의

# 참고 문헌



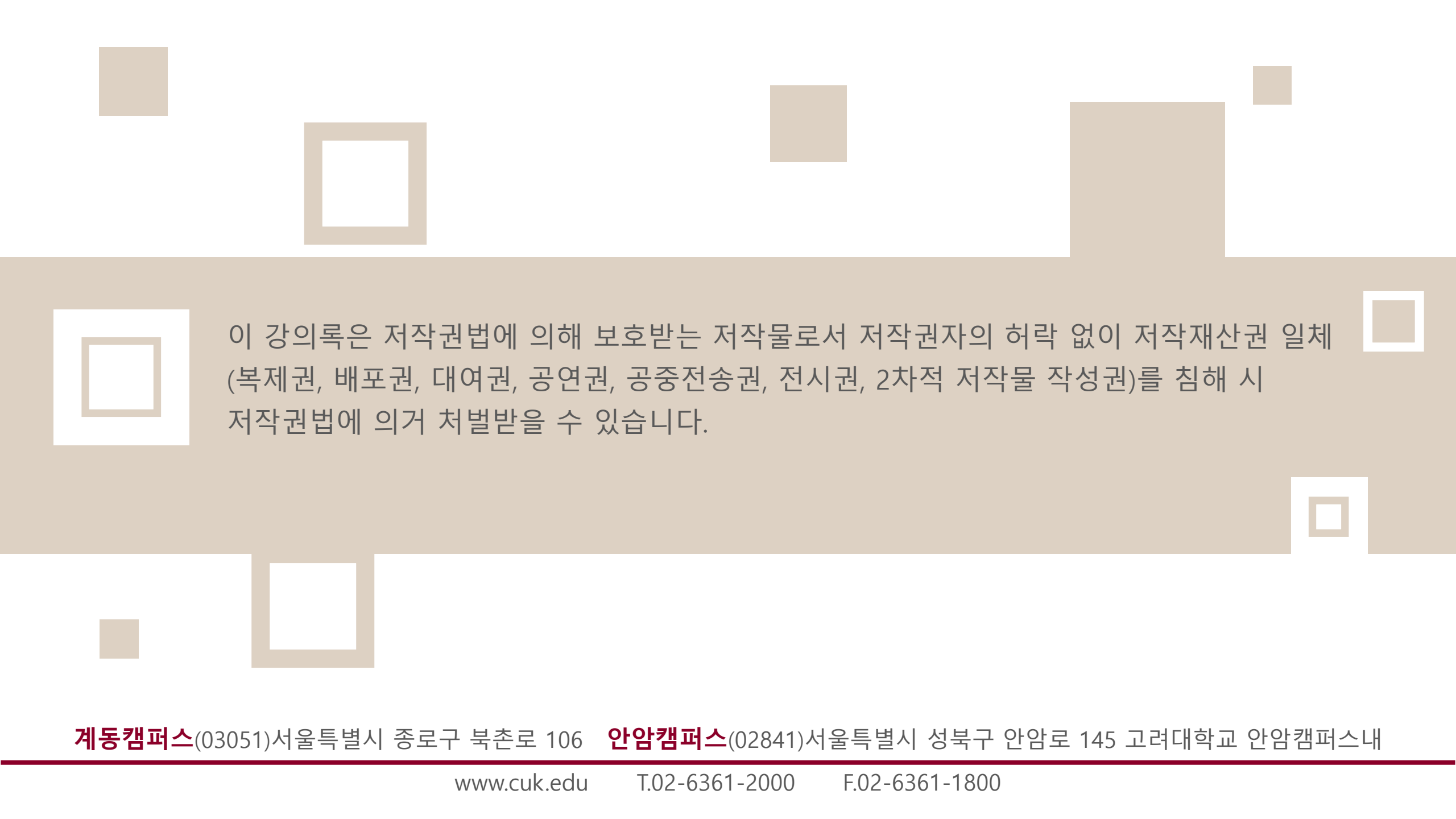
데이터베이스 시스템 7판,  
Ramez Elmasri , Shamkant B. Navathe  
지음, 황규영 등 옮김, 홍릉과학출판사,  
2018년 8월

---



[www.wikipedia.org](http://www.wikipedia.org)

---



이 강의록은 저작권법에 의해 보호받는 저작물로서 저작권자의 허락 없이 저작권재산권 일체 (복제권, 배포권, 대여권, 공연권, 공중전송권, 전시권, 2차적 저작물 작성권)를 침해 시 저작권법에 의거 처벌받을 수 있습니다.