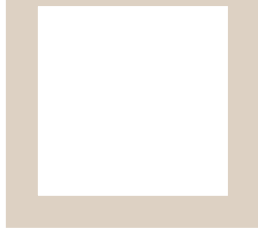


데이터베이스시스템



10. SQL(1/3) - 데이터 정의어



나 홍 석 교수





SQL(1/3) – 데이터 정의어

학습 목표

- 1 SQL의 기본 개념을 이해하고, 그 종류를 설명할 수 있다.
- 2 데이터베이스의 물리적 구조를 설명할 수 있다.
- 3 SQL을 사용하여 데이터베이스와 테이블을 생성할 수 있다.

학습 내용

1 SQL 개요

2 데이터베이스 생성하기

3 테이블 생성하기

4 데이터베이스 구현

Chapter 01 SQL 개요

1 SQL 정의

1 SQL(Structured Query Language)

- ☑ 데이터베이스에 접근할 수 있는 데이터베이스 언어
- ☑ 1974년 IBM 연구소에서 발표한 SEQUEL(Structured English QUery Language)에서 유래
- ☑ 관계대수와 관계 해석을 기초로 한 고급 데이터 언어
- ☑ MySQL, DB2, SQL Server, ORACLE, INFORMIX, SYBASE 등
관계형데이터베이스에서 모두 사용

1 SQL 정의

2 SQL 발전과정

연도	이름	별칭	내용
1986	SQL-86	SQL-87	ANSI에서 최초로 표준 제정(ISO는 1987년에 제정)
1989	SQL-89	FIPS 127-1	소규모 개정, FIPS(연방정부 정보처리 표준) 127-1로 채택
1992	SQL-92	SQL2	대규모 개정(ISO 9075), FIPS 127-2로 채택
1999	SQL:1999	SQL3	정규표현식, 재귀쿼리, 트리거, 객체지향 개념 등 추가
2003	SQL:2003	-	XML관련 특성, 윈도우 함수, 표준화된 시퀀스, 자동 생성값을 가진 컬럼 등이 소개됨
2006	SQL:2006	-	XML과 결합되어 SQL이 사용되는 방법을 정의
2008	SQL:2008	-	커서정의 밖 ORDER BY, INSTEAD OF 트리거 추가
2011	SQL:2011	-	시간(Temporal) 데이터베이스 지원 향상
2016	SQL:2016	-	행 패턴 일치, 폴리모픽 테이블 함수, JSON 추가
2019	SQL:2019	-	다차원 배열(Mdarray) 타입 및 연산 추가

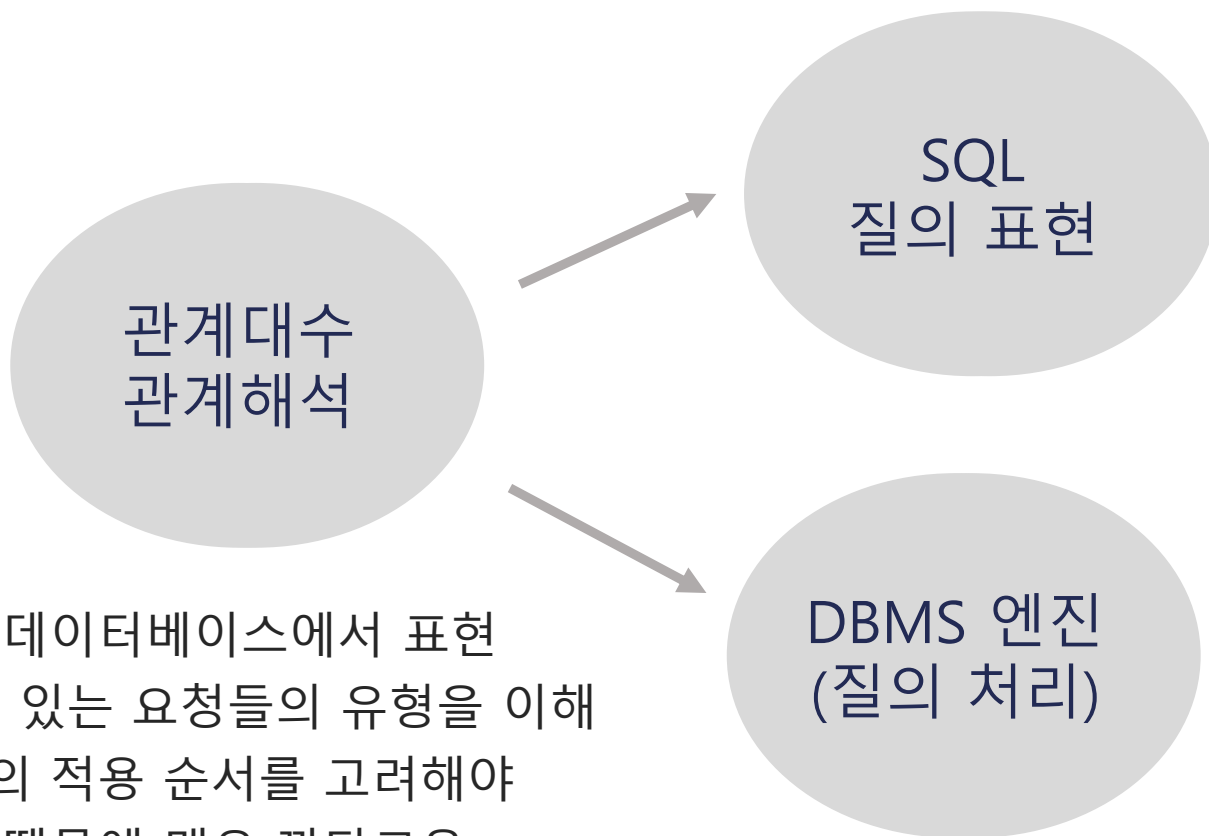
1 SQL 정의

3 SQL 특징

- ☑ 자연어와 유사한 언어로 프로그래밍 경험이 별로 없는 사용자도 쉽게 학습
- ☑ 비절차적언어로 최종 사용자까지 손쉽게 활용함으로써 시스템 개발 및 유지 보수에 소요되는 시간을 줄일 수 있음
- ☑ 단순히 검색만을 위한 질의어가 아니라 데이터 정의, 조작, 제어기능을 모두 제공
- ☑ ANSI, ISO에서 국제 표준으로 채택

1 SQL 정의

4 관계 연산과 SQL



- 관계 데이터베이스에서 표현할 수 있는 요청들의 유형을 이해
- 연산의 적용 순서를 고려해야 하기 때문에 매우 까다로움

- 절차(연산의 적용순서)를 몰라도 원하는 결과를 표현하면 됨
- 데이터 정의어, 데이터 조작어, 데이터 제어어 포함

- SQL로 작성된 질의를 관계연산식으로 분해하고, 재배열하여 최적화된 질의 수행
- 수행 결과는 항상 테이블의 형태

2 SQL 명령어

1 SQL 명령어 종류

명령어 유형	명령어	설명
DDL (데이터정의어)	CREATE, ALTER, DROP, RENAME, TRUNCATE, COMMENT	<ul style="list-style-type: none">구조를 생성하거나, 구조 변경, 삭제 등 데이터 구조에 대한 명령어
DML (데이터조작어)	SELECT INSERT UPDATE DELETE	<ul style="list-style-type: none">가장 활용도가 높은 명령어로서 원하는 데이터를 검색하기 위해 사용하는 명령어새로운 행을 입력하거나 기존의 행을 수정하고자 할 때, 또는 원치 않는 데이터를 삭제하는 등 데이터를 조작하는 것에 대한 명령어
DCL (데이터제어어)	COMMIT, ROLLBACK, SAVEPOINT, GRANT REVOKE	<ul style="list-style-type: none">논리적인 작업의 단위로 DML에 의해 조작된 결과를 다루는 명령어데이터베이스에 접근하는 권한을 부여하고 객체들을 사용하도록 주고 받는 명령어

2 SQL 명령어

2 SQL 명령어 예 #1

☑ 모든 학생의 이름, 학과를 보여라

☑ SQL문 → **SELECT 이름, 학과 FROM 학생**

학생

학번	이름	성별	주소	학과
ST001	최현주	여	서울	소프트웨어
ST002	강하늘	남	서울	경영
ST003	이성민	남	서울	소프트웨어
ST004	박정수	여	경기	전기전자
ST005	홍민호	남	대전	경영



SQL> SELECT 이름, 학과 FROM 학생

이름	학과
최현주	소프트웨어
강하늘	경영
이성민	소프트웨어
박정수	전기전자
홍민호	경영

2 SQL 명령어

2 SQL 명령어 예 #2

☑ 소프트웨어학과 학생의 학번, 이름을 출력하라.

☑ SQL문 → **SELECT 학번, 이름 FROM 학생 WHERE 학과 = '소프트웨어'**

학생

학번	이름	성별	주소	학과
ST001	최현주	여	서울	소프트웨어
ST002	강하늘	남	서울	경영
ST003	이성민	남	서울	소프트웨어
ST004	박정수	여	경기	전기전자
ST005	홍민호	남	대전	경영

SQL> SELECT 학번, 이름 FROM 학생
WHERE 학과 = '소프트웨어'



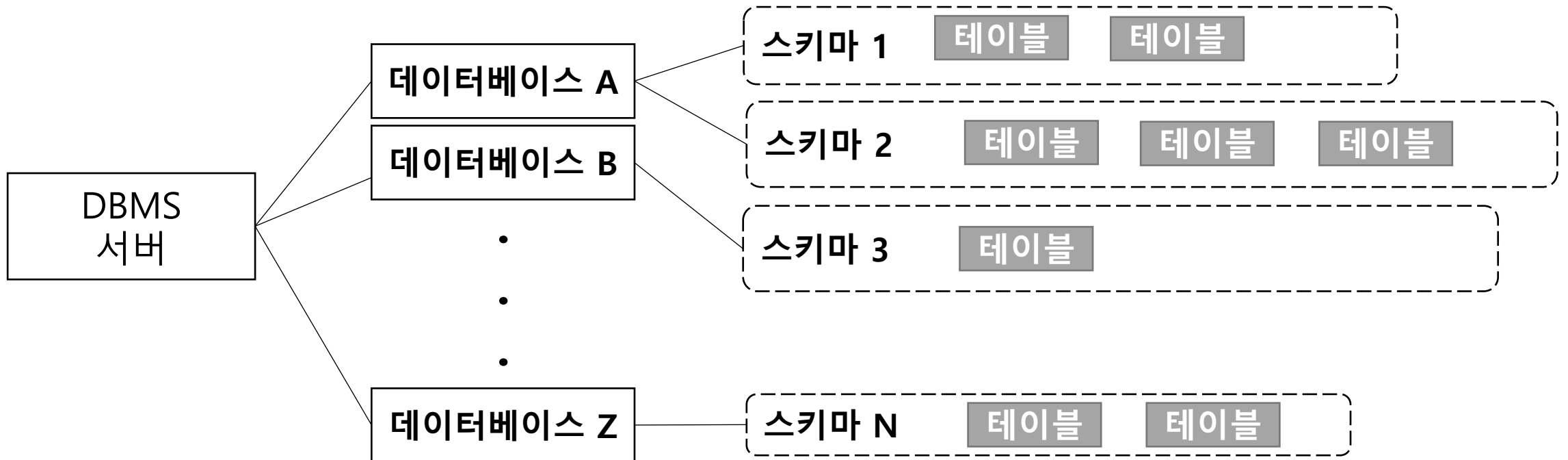
학번	이름
ST001	최현주
ST003	이성민

Chapter 02 데이터베이스 생성하기

1 데이터 저장구조

1 관계형 DBMS

- ☑ 데이터베이스 = 릴레이션(테이블)의 집합
- ☑ DBMS는 데이터베이스를 구현할 수 있는 방법을 제공해야 함



1 데이터 저장구조

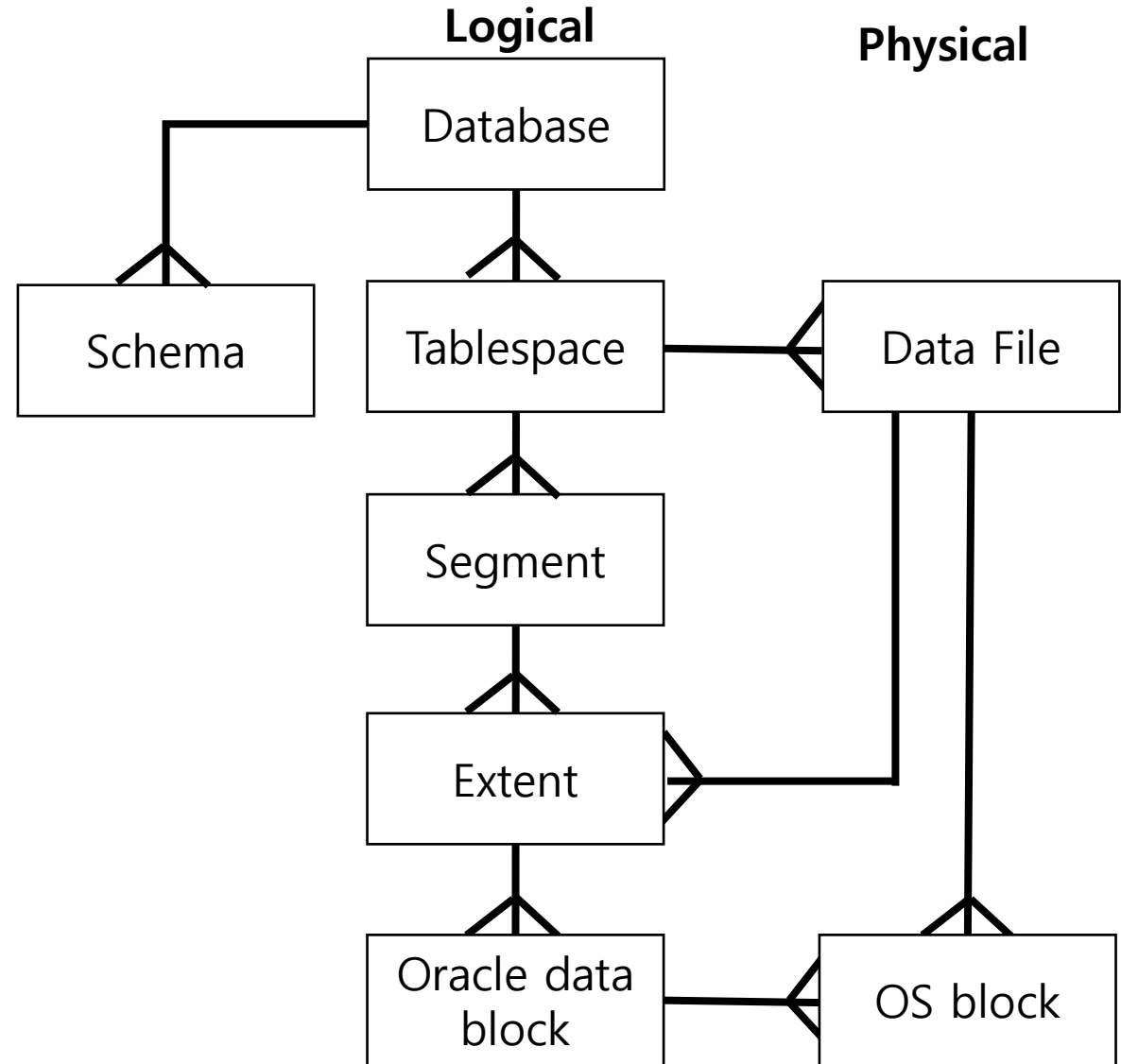
2 ORACLE 데이터 저장구조 #1

논리적 구조

- 테이블스페이스
- 세그먼트
- 익스텐트
- 데이터블록(페이지)

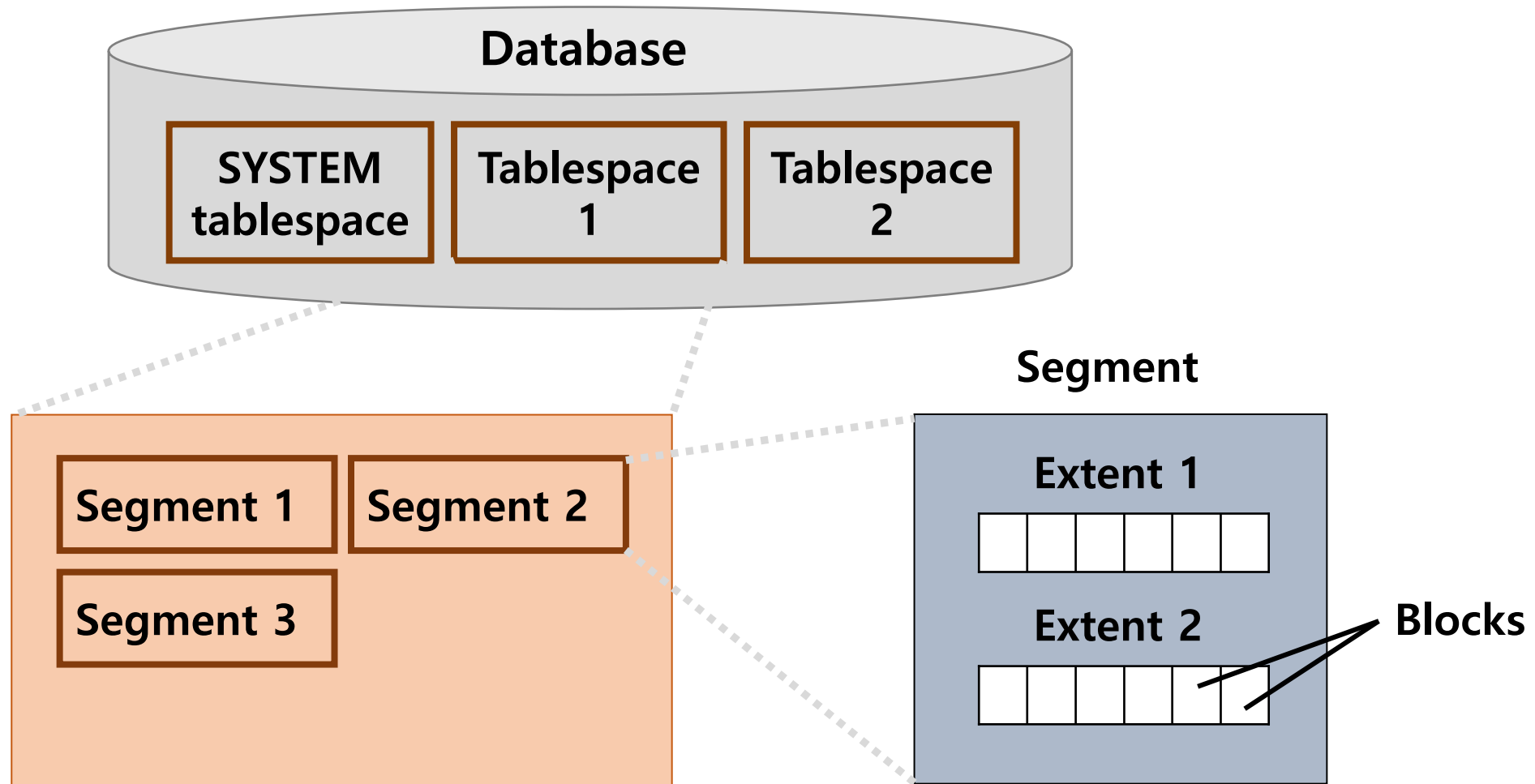
물리적 구조

- 데이터 파일
- 로그파일
- 컨트롤 파일



1 데이터 저장구조

2 ORACLE 데이터 저장구조 #2



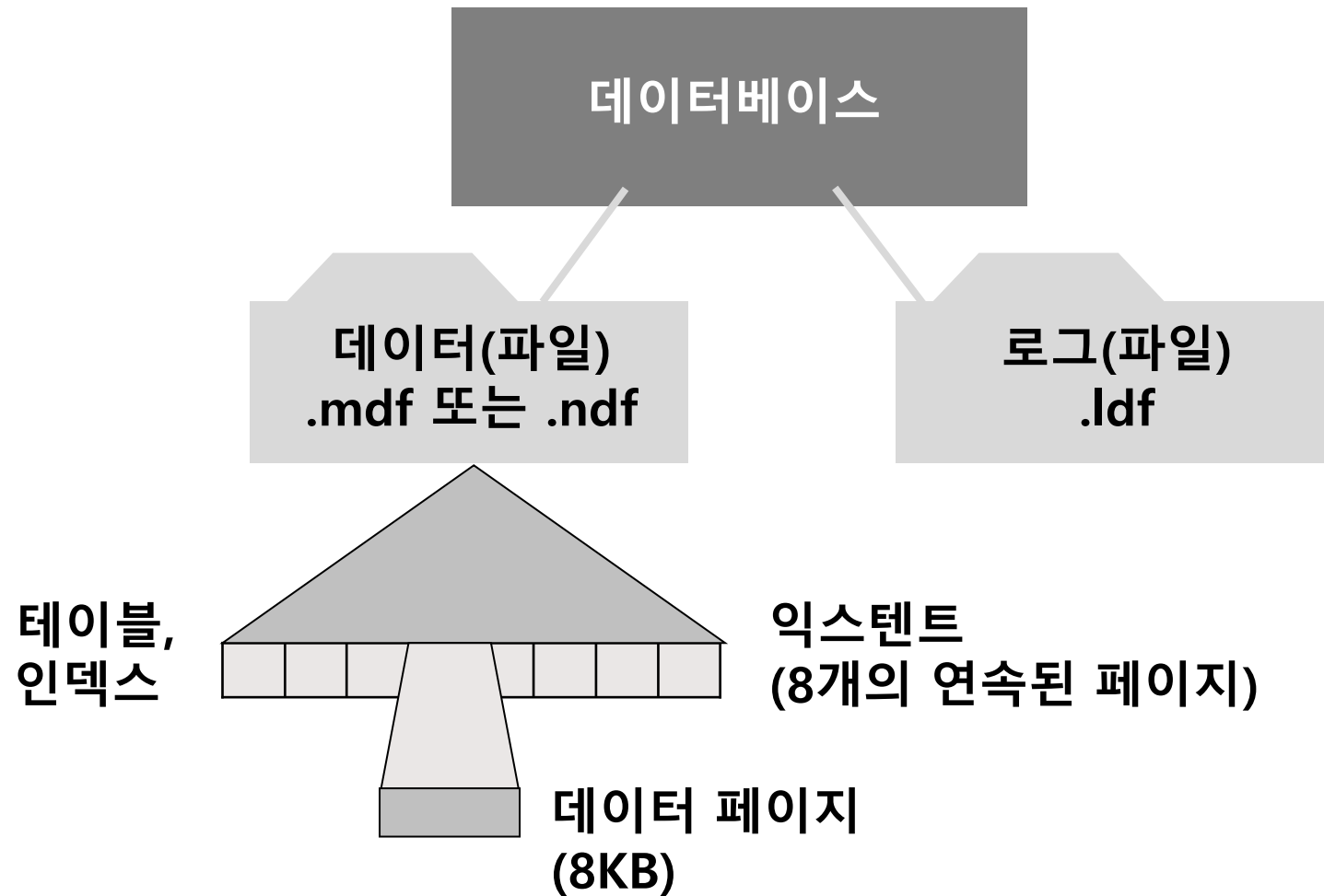
1 데이터 저장구조

3 SQL server 데이터 저장구조 #1

- ☑ DBMS는 여러 개의 데이터베이스를 동시에 운영함
- ☑ 하나의 데이터베이스는 데이터 파일(.mdf)과 로그파일(.ldf)로 구성됨
- ☑ 데이터 파일에서 데이터는 페이지(8k)라는 최소의 저장단위에 저장됨
- ☑ 8개의 연속된 페이지 → 익스텐트(Extent)
- ☑ 로그 파일에는 데이터의 변경에 대한 내용이 기록됨

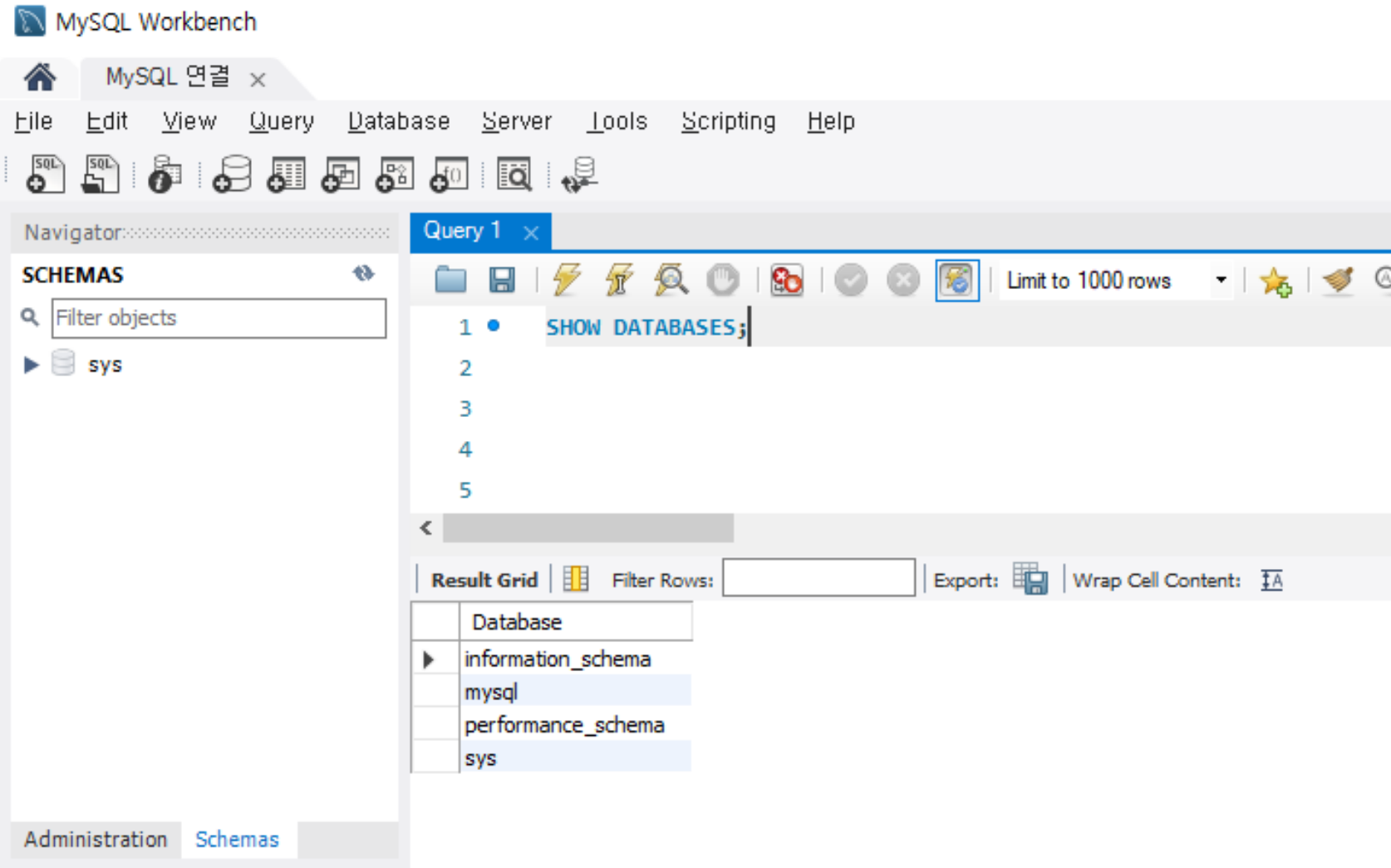
1 데이터 저장구조

3 SQL server 데이터 저장구조 #2



1 데이터 저장구조

4 MySQL/MariaDB 저장구조



☑ 오라클과 유사한 구조

☑ Database(Schema)

→ Table

2 SQL을 이용한 데이터베이스 생성

1 CREATE DATABASE 명령어 - SQL Server, MySQL/MariaDB

- ☑ 하나의 새로운 데이터베이스를 생성
- ☑ 데이터베이스 생성을 위한 기본파일(데이터파일, 로그파일 등)과 함께 관리를 위한 시스템테이블 등을 생성
- ☑ 데이터베이스 삭제 → DROP DATABASE



예) CREATE DATABASE academy
DROP DATABASE academy

2 SQL을 이용한 데이터베이스 생성

2 ORACLE

 CREATE TABLESPACE 명령어

 데이터가 저장될 파일경로와 이름도 함께 지정



예)

```
CREATE TABLESPACE academy
```

```
DATAFILE 'C:\ORACLE\ORADATA\academy_data.dbf'
```

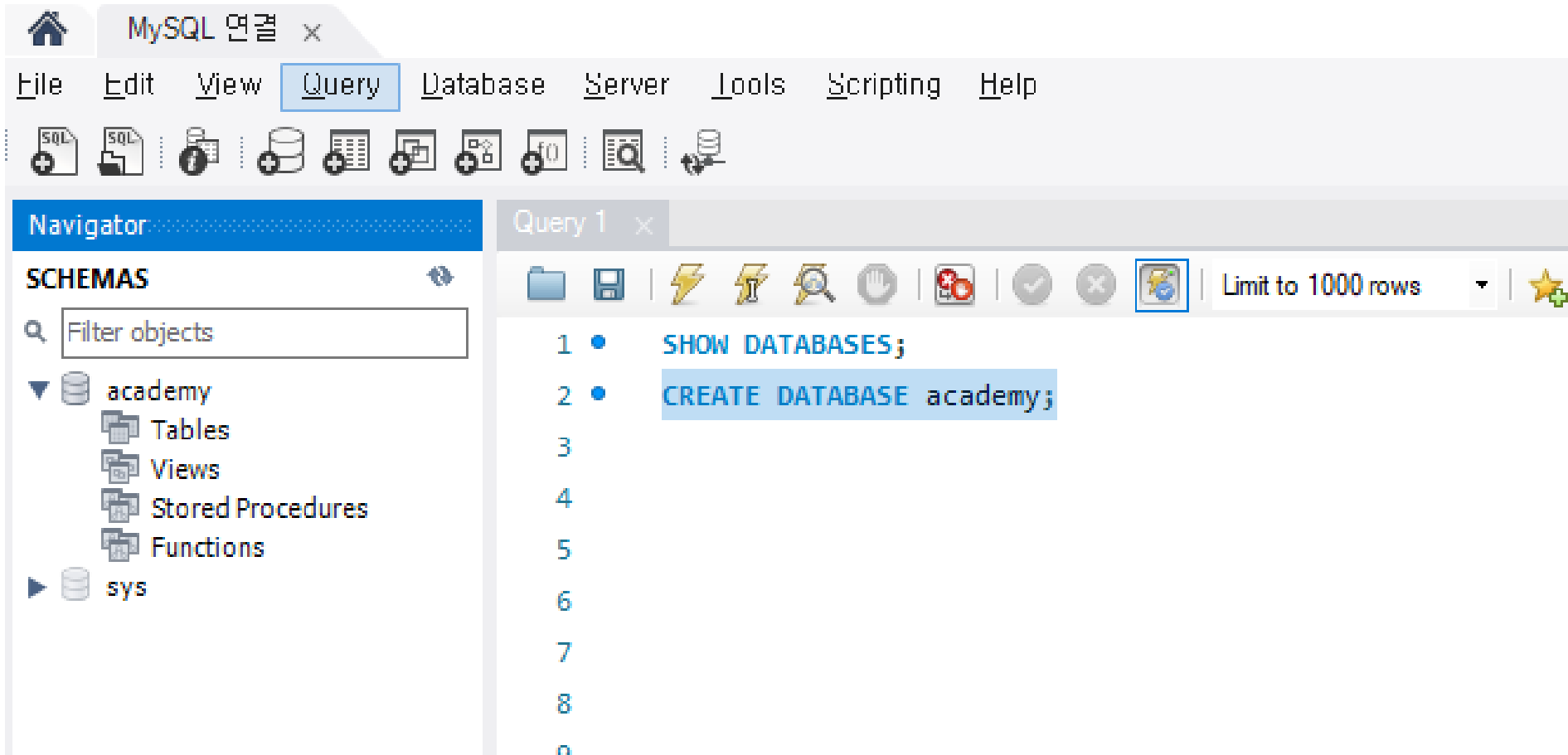
```
SIZE 100M
```

2 SQL을 이용한 데이터베이스 생성

3 CREATE DATABASE 명령어 실행 #1

MySQL Workbench

CREATE DATABASE: 데이터 생성 명령어



2 SQL을 이용한 데이터베이스 생성

3 CREATE DATABASE 명령어 실행 #2

SHOW DATABASES: 현재 생성된 데이터베이스를 보여주는 명령어

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the 'SCHEMAS' section with a search filter and a tree view showing 'academy' and 'sys' databases. The 'Query 1' editor in the center contains two SQL commands: 'SHOW DATABASES;' and 'CREATE DATABASE academy;'. The 'Result Grid' at the bottom displays the output of the first command, listing the databases: 'academy', 'information_schema', 'mysql', 'performance_schema', and 'sys'.

MySQL Workbench

MySQL 연결 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

▼ academy

- Tables
- Views
- Stored Procedures
- Functions

► sys

Query 1 x

1 • SHOW DATABASES;

2 • CREATE DATABASE academy;

3

4

5

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Database
academy
information_schema
mysql
performance_schema
sys

Administration Schemas

Chapter 03 테이블 생성하기

1 SQL 테이블 정의

1 테이블 구조

☑ 데이터베이스는 여러 개의 테이블로 구성

☑ 애트리뷰트 → 테이블의 컬럼(열), 튜플 → 테이블의 로우(행, 레코드)

테이블 이름

Student

컬럼(Columns)

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

1 SQL 테이블 정의

2 CREATE TABLE 명령어

형식	<pre>CREATE TABLE <i>table_name</i> ({ <i>column_name</i> datatype column_constraint } [, { <i>column_name</i> datatype column_constraint }] ...)</pre>
컬럼제약	<ul style="list-style-type: none">▪ NOT NULL: NULL 값이 올 수 없다.▪ PRIMARY KEY: 중복될 수 없으며, NULL이 될 수 없다.▪ UNIQUE: 중복될 수 없다.▪ DEFAULT value: 컬럼의 기본 값을 지정한다.▪ FOREIGN KEY: 외래키를 지정한다.▪ CHECK: 컬럼이 가질 수 있는 값의 범위나 형식을 지정한다.
테이블 제약	<ul style="list-style-type: none">▪ PRIMARY KEY: 테이블의 기본키를 설정한다.▪ FOREIGN KEY: 테이블의 외래키를 설정한다.▪ CHECK: 하나이상의 컬럼에 제약조건을 지정한다.

1 SQL 테이블 정의

3 테이블 생성 SQL문 #1

Student 테이블 정의 - 일부 수정

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							F, M
4	st_phn_mbl	전화번호	char	13	Y	Y					
5	st_addr	주소	varchar	50	Y						
6	st_birth	생년월일	Datetime		Y						
7	dept_id	소속학과	char	2	Y				department	dept_id	IT, BZ, MD, ...

1 SQL 테이블 정의

3 테이블 생성 SQL문 #2

 Student 테이블 생성 SQL 문

```
CREATE TABLE student
(
    st_name          VARCHAR(20) ,
    st_id            VARCHAR(10),
    st_sex           CHAR(1),
    st_phn_mbl       CHAR(13),
    st_addr          VARCHAR(50),
    st_birth         DATETIME,
    dept_id          CHAR(2)
);
```

1 SQL 테이블 정의

3 테이블 생성 SQL문 #3

MySQL Workbench

MySQL 연결 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

academy

Tables

Views

Stored Procedures

Functions

sys

Query 1 x

```
1 • SHOW DATABASES;
2 • CREATE DATABASE academy;
3
4 • USE academy;
5
6 • CREATE TABLE student
7 (
8     st_name VARCHAR(20),
9     st_id VARCHAR(10),
10    st_sex CHAR(1),
11    st_phn_mbl CHAR(13),
12    st_addr CHAR(50),
13    st_birth DATETIME,
14    dept_id CHAR(2)
15 );
```

Administration Schemas

Information

MySQL Workbench

MySQL 연결 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

academy

Tables

student

Columns

- st_name
- st_id
- st_sex
- st_phn_mbl
- st_addr
- st_birth
- dept_id

Indexes

Foreign Keys

Triggers

Views

Administration Schemas

Information

No object selected

Object Info Session

Query 1 x

Limit to 1000 rows

```
1 • SHOW DATABASES;
2 • USE academy;
3
4 • CREATE TABLE student
5 (
6     st_name VARCHAR(20),
7     st_id VARCHAR(10),
8     st_sex CHAR(1),
9     st_phn_mbl CHAR(13),
10    st_addr CHAR(50),
11    st_birth DATETIME,
12    dept_id CHAR(2)
13 );
14
15
16
```

Output

Action Output

#	Time	Action	Message
✓ 2	22:42:18	USE academy	0 row(s) affected
✓ 3	22:42:22	DROP TABLE employee	0 row(s) affected
✓ 4	22:44:47	CREATE TABLE student (st_name VARCHAR(20), st_id VA...	0 row(s) affected

1 SQL 테이블 정의

3 테이블 생성 SQL문 #4

DROP TABLE: 테이블 삭제 명령어

The screenshot displays the MySQL Workbench interface. On the left, the 'Navigator' pane shows a tree view of the 'academy' database schema, including tables, columns, indexes, foreign keys, triggers, and views. The 'student' table is expanded, showing its columns: st_name, st_id, st_sex, st_phn_mbl, st_addr, st_birth, and dept_id. The 'Administration' tab is selected, and the 'Schemas' sub-tab is active. The main query editor, titled 'Query 1', contains the following SQL code:

```
1 • SHOW DATABASES;
2 • USE academy;
3
4 • CREATE TABLE student
5 (
6     st_name    VARCHAR(20) ,
7     st_id      VARCHAR(10),
8     st_sex     CHAR(1),
9     st_phn_mbl CHAR(13),
10    st_addr     CHAR(50),
11    st_birth    DATETIME,
12    dept_id     CHAR(2)
13 );
14
15 • DROP TABLE student;
16
17
```

The 'DROP TABLE student;' statement is highlighted in blue. The status bar at the bottom indicates 'No object selected'.

2 제약조건

1 정의

- ☑ 데이터베이스 무결성을 구현하기 위한 도구
- ☑ 컬럼이나 테이블에 저장되는 값이 준수해야 할 조건을 명시

NOT NULL

PRIMARY KEY

UNIQUE

DEFAULT

CHECK

FOREIGN KEY

2 제약조건

2 PRIMARY KEY 제약조건

- ☑ PRIMARY KEY = UNIQUE + NOT NULL
- ☑ 테이블당 하나만 가질 수 있으며, 자동으로 인덱스가 생성됨

```
CREATE TABLE student
(
    st_name          VARCHAR(20),
    st_id            VARCHAR(10) PRIMARY KEY,
    st_sex           CHAR(1),
    st_phn_mbl       CHAR(13),
    st_addr          CHAR(50),
    st_birth         DATETIME,
    dept_id          CHAR(2)
);
```


2 제약조건

3 NOT NULL 제약조건

- ☑ NULL 값을 허용하지 않겠다는 것을 의미
- ☑ 기본키를 제약조건 = 유일해야 하고, 널값을 허용하지 않음

```
CREATE TABLE student
(
    st_name          VARCHAR(20) NOT NULL,
    st_id            VARCHAR(10) PRIMARY KEY,
    st_sex           CHAR(1) NOT NULL,
    ...
);
```

2 제약조건

4 DEFAULT 제약조건

- ☑ 데이터를 삽입할 때 컬럼의 값을 지정하지 않으면 미리 정의된 값을 넣어주는 제약조건
- ☑ 스칼라 값(문자, 숫자)과 스칼라 함수를 이용할 수 있음

사용예

address
regdate

VARCHAR(30)
DATETIME

DEFAULT '서울' ,
DEFAULTL getdate() ,

2 제약조건

5 CHECK 제약조건

- ☑ 삽입되는 데이터를 검사해서 해당되는 영역의 데이터이면 진행하고, 그렇지 않으면 삽입작업을 취소하는 역할을 수행
- ☑ True 혹은 False 값을 만족하는 어떤 조건도 포함될 수 있음

```
CREATE TABLE student
(
    ...
    st_sex          CHAR(1)  CHECK (st_sex IN ('M','F')),
    ...
);
```

2 제약조건

6 UNIQUE 제약조건

- ☑ 중복값을 허용하지 않음(NULL값은 허용)
- ☑ 테이블에 여러 개가 존재할 수 있음

```
CREATE TABLE student  
(  
    ...  
    st_phn_mbl CHAR(13) UNIQUE,  
    ...  
);
```

2 제약조건

7 FOREIGN KEY 제약조건 #1 - 테이블 정의 시

```
CREATE TABLE student
```

```
(
```

```
  st_name          VARCHAR(20) NOT NULL,  
  st_id            VARCHAR(10) PRIMARY KEY,  
  st_sex           CHAR(1)  NOT NULL,  
  st_phn_mbl       CHAR(13) UNIQUE,  
  st_addr          CHAR(50),  
  st_birth         DATETIME,  
  dept_id          CHAR(2),
```

```
  FOREIGN KEY (dept_id) REFERENCES department(dept_id)
```

```
);
```

- 참조하는 테이블(컬럼)에 데이터가 삽입될 때에는 항상 참조되는 테이블(컬럼)에 있는 값으로 들어gå야 함

2 제약조건

7 FOREIGN KEY 제약조건 # - 제약조건 추가

 ALTER TABLE 명령어를 이용해서 FOREIGN KEY 추가

```
ALTER TABLE student  
ADD FOREIGN KEY (dept_id) REFERENCES department(dept_id) ;
```

[참조되는 테이블의 값이 삭제될 경우 널로 처리하는 방법]

```
ALTER TABLE student  
ADD FOREIGN KEY (dept_id) REFERENCES department(dept_id)  
ON DELETE SET NULL ;
```

3 SQL 테이블 변경

1 테이블 삭제 명령어

- ✓ DROP TABLE 명령어를 이용해서 테이블을 삭제함
- ✓ 테이블의 모든 데이터와 함께 테이블이 삭제됨
- ✓ (참고) TRUNCATE TABLE 명령어는 테이블은 남겨놓고 내용만 삭제함

형식	DROP TABLE <테이블명>[,<테이블명>, ...]
예제	DROP TABLE student;

3 SQL 테이블 변경

2 테이블 변경 명령어

- ✓ ALTER TABLE 명령어를 이용해서 테이블 구조 변경
- ✓ 컬럼 추가 혹은 삭제, 컬럼 타입 변경, 컬럼명이나 테이블명을 변경 등

형식	ALTER TABLE <테이블명> ADD COLUMN <컬럼명> <컬럼타입> ALTER TABLE <테이블명> DROP COLUMN <컬럼명> ALTER TABLE <테이블명> ALTER COLUMN <이전컬럼명> <새컬럼명> <새컬럼타입> ALTER TABLE <이전 테이블명> RENAME AS <새테이블명>
예제	ALTER TABLE student ADD COLUMN st_addr VARCHAR(50); ALTER TABLE student DROP COLUMN st_addr; ALTER TABLE student ALTER COLUMN st_addr VARCHAR(100);

3 SQL 테이블 변경

3 컬럼의 추가

- ☑ 항상 기존의 열 끝에 추가됨
- ☑ NOT NULL 이면서 기본 값도 없는 열은 추가할 수 없음
- ☑ NULL로 설정하거나 기본값을 정의해야 함

```
ALTER TABLE student  
ADD COLUMN st_email VARCHAR(30) NOT NULL ;
```

```
ALTER TABLE student  
ADD COLUMN st_email VARCHAR(30) NOT NULL DEFAULT "----- NO EMAIL -----";
```

3 SQL 테이블 변경

4 컬럼의 수정

- ☑ 기본값, 체크 등의 제약조건이나 개체 기본값, 규칙 또는 인덱스가 설정된 열은 수정불가
- ☑ 데이터형의 수정은 호환이 있는 것으로 해야 함
- ☑ 숫자형의 경우 작은 크기로 수정할 수 없으며, 기존의 열에 들어있던 가장 긴 문자열보다 길이가 짧은 문자 데이터형으로 수정불가

```
ALTER TABLE student
```

```
ALTER COLUMN st_addr VARCHAR(100) NOT NULL
```

3 SQL 테이블 변경

5 컬럼의 삭제

- ☑ 기본값, 체크 등의 제약조건이나 개체 기본값, 규칙 또는 인덱스가 설정된 열은 삭제불가함
- ☑ 제약조건을 없애고 삭제해야 함
- ☑ 열 삭제는 데이터를 영구적으로 없애므로, 반드시 백업한 후에 수행해야 함

```
ALTER TABLE student DROP COLUMN st_email;
```

Chapter 04 데이터베이스 구현

1 학생 테이블 생성

1 테이블 정의서

student 테이블 정의서 - 외래키 제약조건 생략

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							F, M
4	st_phn_mbl	전화번호	char	13	Y	Y					
5	st_addr	주소	varchar	50	Y						
6	st_birth	생년월일	Datetime		Y						
7	dept_id	소속학과	char	2	Y						

1 학생 테이블 생성

2 SQL문 작성 및 실행

The screenshot displays a database management interface. On the left, the 'Navigator' pane shows the 'academy' schema with a 'student' table. The table's columns are listed: st_name, st_id, st_sex, st_phn_mbl, st_addr, st_birth, and dept_id. On the right, the 'Query 1' editor shows the SQL code for creating the 'student' table. The code is as follows:

```
4 • USE academy;
5
6 • CREATE TABLE student
7 (
8     st_name    VARCHAR(20)    NOT NULL,
9     st_id      VARCHAR(10)    PRIMARY KEY,
10    st_sex      CHAR(1)        NOT NULL,
11    st_phn_mbl  CHAR(13)       UNIQUE,
12    st_addr     CHAR(50),
13    st_birth    DATETIME,
14    dept_id     CHAR(2)
15 );
16
```

2 과목 테이블 생성

1

테이블 정의서



course 테이블 정의서

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 과목 테이블 생성

2 SQL문 작성 및 실행

The screenshot displays a database management interface. On the left, the 'Navigator' pane shows the 'academy' schema with a tree view of its objects: Tables (course, enrol, student), Views, Stored Procedures, and Functions. The 'course' table is selected. The main area, titled 'Query 1', shows a SQL script for creating the 'course' table. The script includes a partial definition for a table with columns 'st_birth' (DATETIME) and 'dept_id' (CHAR(2)), followed by a complete 'CREATE TABLE course' statement with columns 'co_num' (CHAR(5) PRIMARY KEY), 'co_name' (VARCHAR(20) NOT NULL UNIQUE), and 'co_location' (VARCHAR(20)).

```
13      st_birth      DATETIME,  
14      dept_id       CHAR(2)  
15  );  
16  
17  • CREATE TABLE course  
18  (  
19      co_num         CHAR(5)    PRIMARY KEY,  
20      co_name        VARCHAR(20) NOT NULL UNIQUE,  
21      co_location    VARCHAR(20)  
22  );  
23
```


3 등록 테이블 생성

1

테이블 정의서



enrol 테이블 정의서

테이블명		enrol									
테이블정의		K사이버대학교 등록 정보 테이블									
No	컬럼명	한글명	타입	길이	NULL	UK	PK	FK	참조테이블	참조컬럼	비고
1	st_id	학생번호	varchar	10			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 등록 테이블 생성

2 SQL문 작성 및 실행

The screenshot displays a database management interface. On the left, the 'Navigator' pane shows the 'academy' schema with tables 'course' and 'enrol'. The 'Columns' section for 'enrol' lists 'st_id', 'co_num', 'grade', 'mid', and 'final'. The main area, 'Query 1', contains the following SQL code:

```
23  
24 • CREATE TABLE enrol  
25 (  
26     st_id      VARCHAR(10),  
27     co_num     CHAR(5),  
28     grade      CHAR(2),  
29     mid        int,  
30     final      int,  
31  
32     PRIMARY KEY (st_id, co_num),  
33     FOREIGN KEY (st_id) REFERENCES student(st_id),  
34     FOREIGN KEY (co_num) REFERENCES course(co_num)  
35 );  
36
```

The interface includes a toolbar with icons for file operations, execution, and search. The 'Limit to 1000 rows' option is visible. The bottom status bar shows 'Name: MySQL 연결'.

4 테이블 생성 확인

1 SHOW TABLES 명령문

The screenshot shows a database management interface. On the left, the 'Navigator' pane displays the 'academy' schema with tables 'course', 'enrol', and 'student'. The 'course' table is selected. The main query editor shows the command 'SHOW TABLES;' at line 37. The results pane at the bottom displays a table with the following data:

Tables_in_academy
course
enrol
student

학습 정리



SQL

- 관계대수와 관계 해석을 기초로 한 고급 데이터 언어

명령어 유형	명령어	설명
DDL (데이터정의어)	CREATE, ALTER, DROP, RENAME, TRUNCATE, COMMENT	<ul style="list-style-type: none"> 구조를 생성하거나, 구조 변경, 삭제 등 데이터 구조에 대한 명령어
DML (데이터조작어)	SELECT INSERT UPDATE DELETE	<ul style="list-style-type: none"> 가장 활용도가 높은 명령어로서 원하는 데이터를 검색하기 위해 사용하는 명령어 새로운 행을 입력하거나 기존의 행을 수정하고자 할 때, 또는 원치 않는 데이터를 삭제하는 등 데이터를 조작하는 것에 대한 명령어
DCL (데이터제어어)	COMMIT, ROLLBACK, SAVEPOINT, GRANT REVOKE	<ul style="list-style-type: none"> 논리적인 작업의 단위로 DML에 의해 조작된 결과를 다루는 명령어 데이터베이스에 접근하는 권한을 부여하고 객체들을 사용하도록 주고 받는 명령어

학습 정리

데이터베이스 생성과 삭제

- MySQL/MariaDB - `CREATE DATABASE` 명령문 이용
- MS SQL Server - `CREATE DATABASE` 명령문 이용
- ORACLE `CREATE TABLESPACE` 명령문 이용

학습 정리



테이블 생성과 삭제

- 데이터베이스는 여러 개의 테이블로 구성
- **CREATE TABLE** : 테이블 생성 명령어
- **ALTER TABLE** : 테이블 변경 명령어
- **DROP TABLE** : 테이블 삭제 명령어

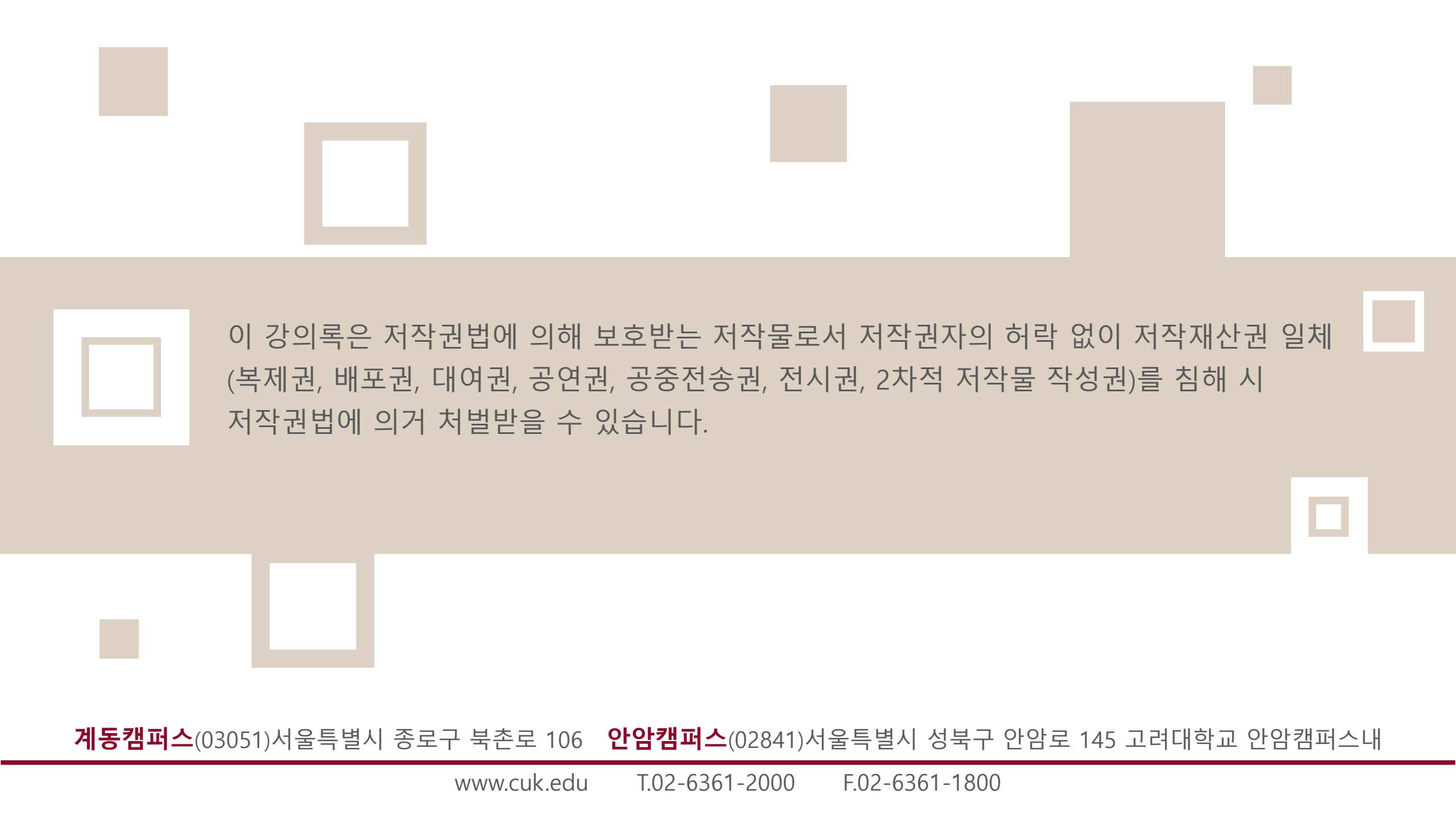
참고 문헌



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



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.