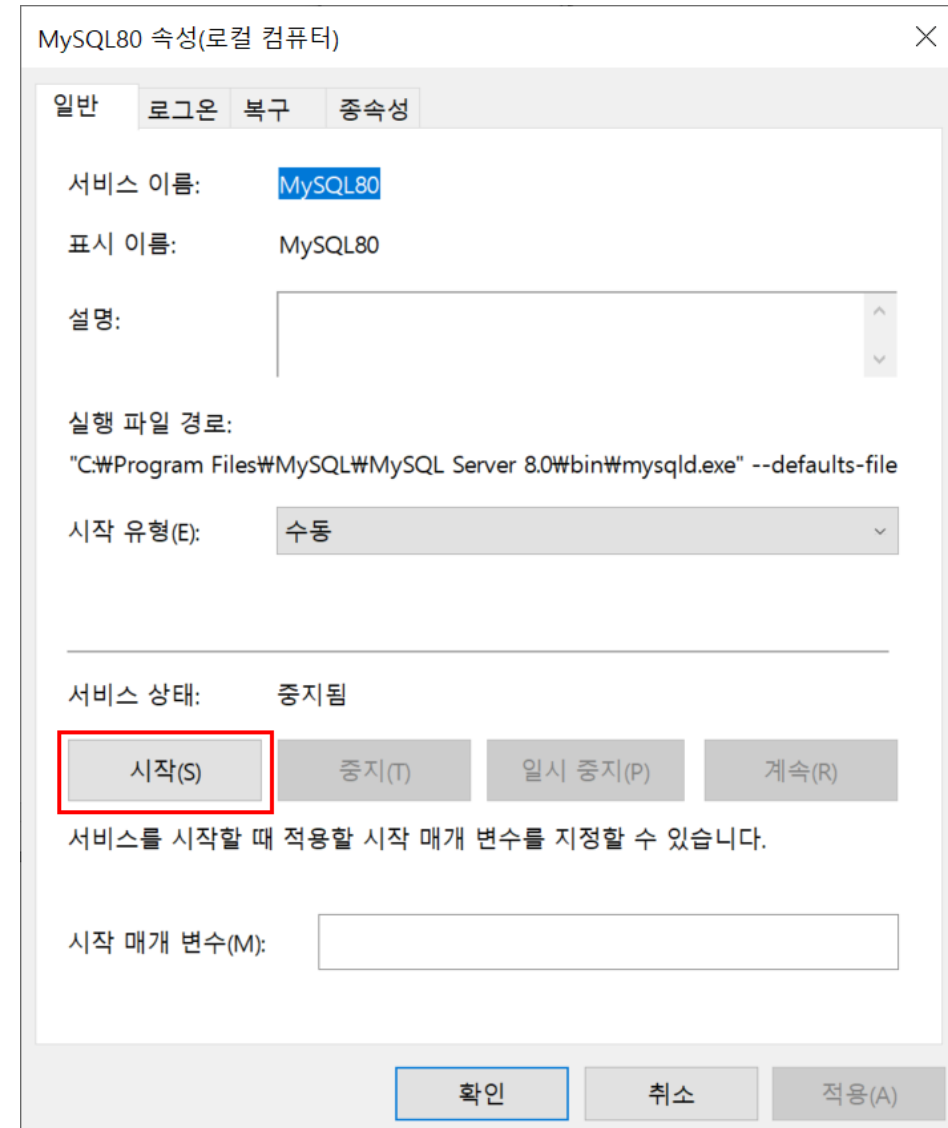
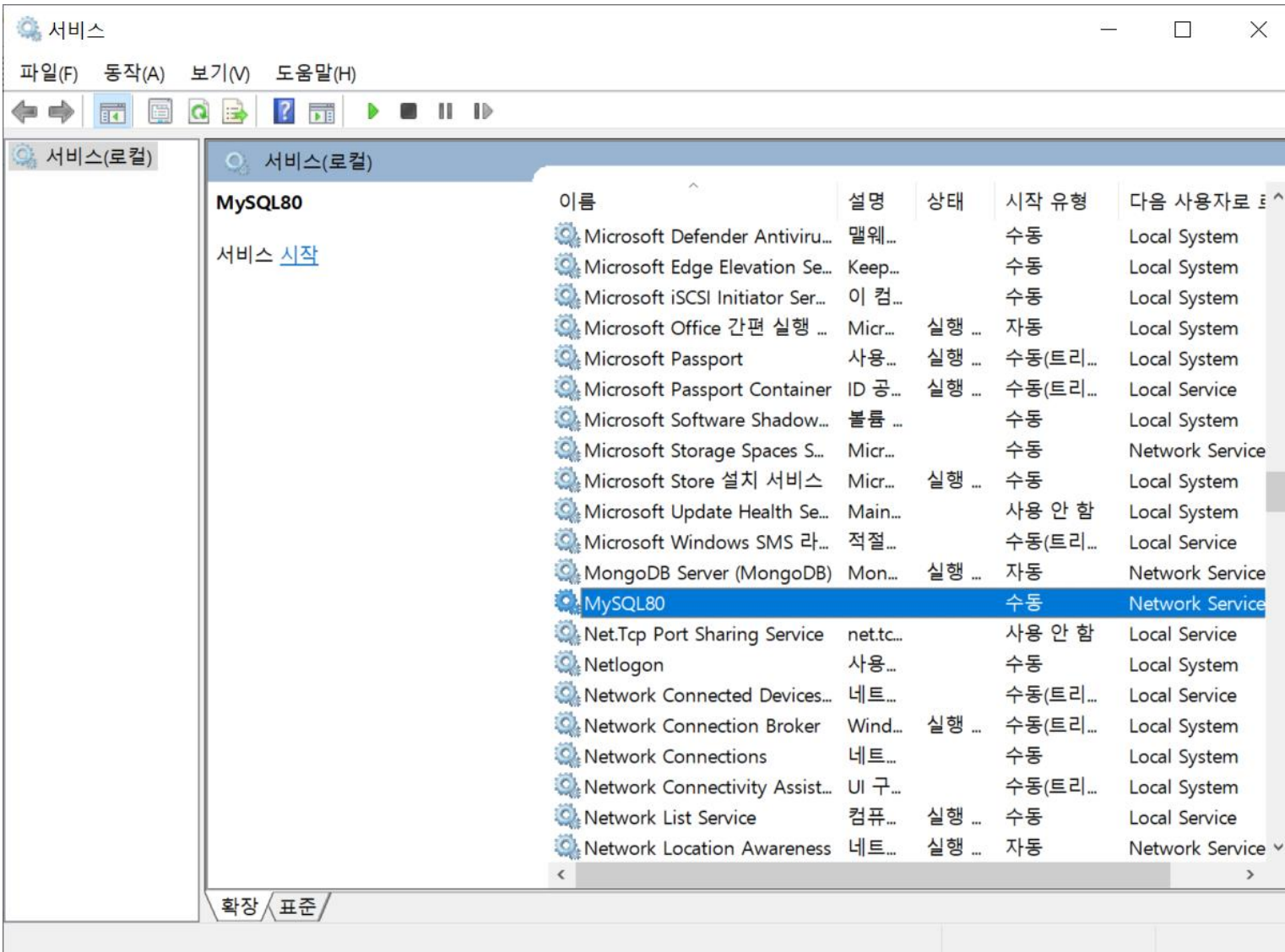


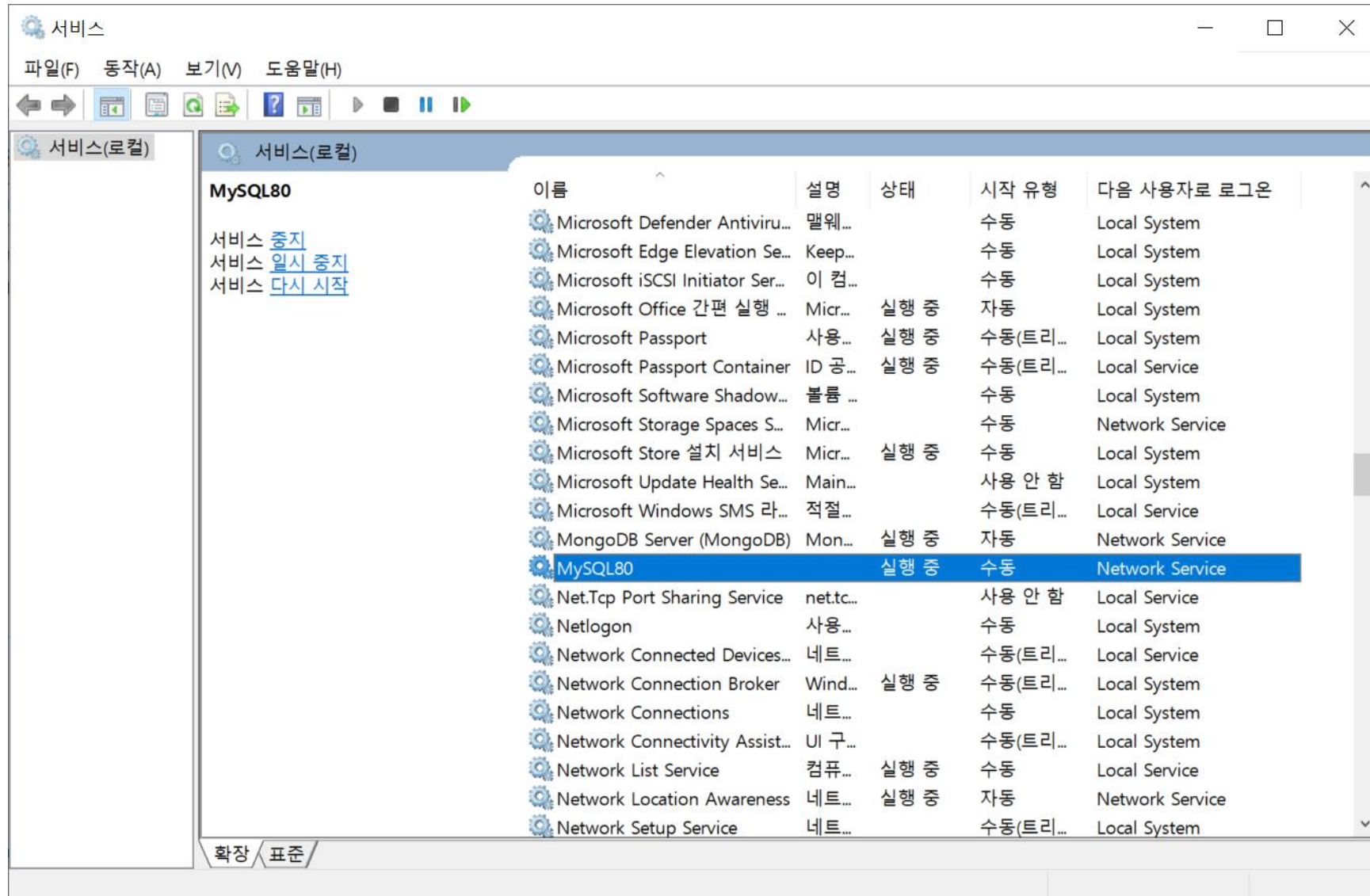


데이터베이스 만들기과 SQL(Structured Query Language)

※ 실습 전 <MySQL 서버> 시작 여부 확인



※ 실습 전 <MySQL 서버> 시작 여부 확인



데이터베이스 만들기과 SQL(Structured Query Language)

- 스마트폰 연락처에서 " 김한성 " 이라는 이름의 전화번호를 검색하시오.

Query : 데이터베이스
테이블에서 조건에 맞는 것을
선택하는 것



- 데이터베이스 = 스마트폰 내부에 존재
- 테이블 = 연락처
- 이름=김한성
- 검색

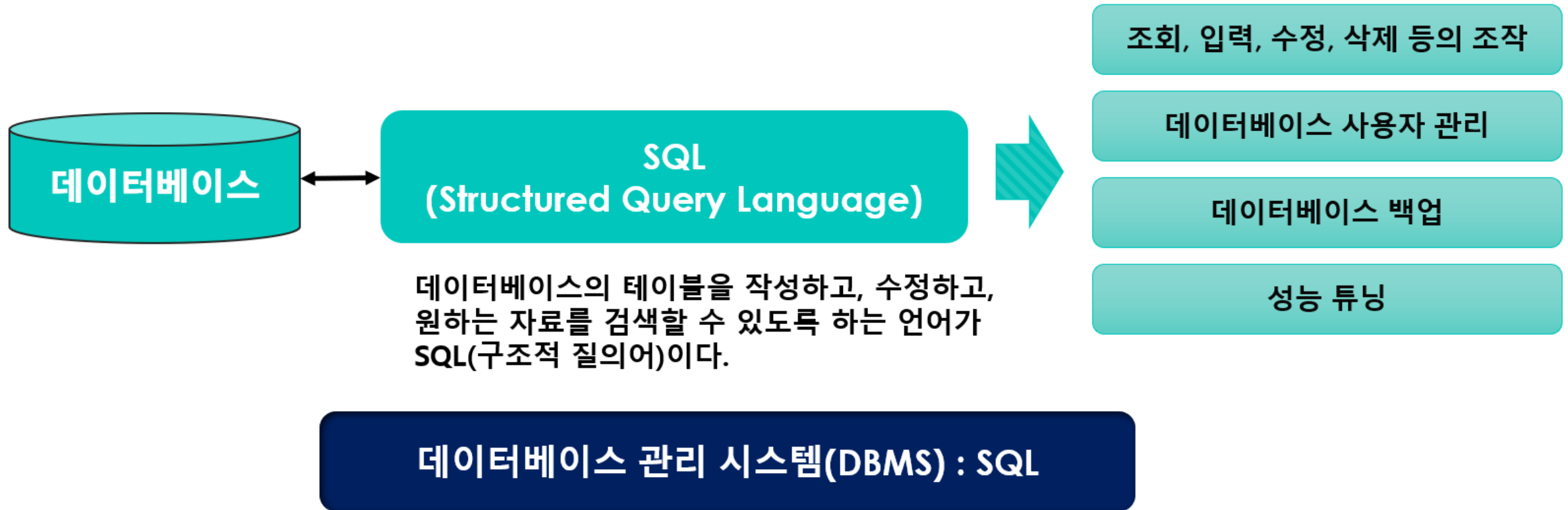
<연락처>

이름	전화번호	직장	...
고모	010-1234-5678	한성대
김한성	010-2345-6789	학생
김영미	010-2345-7890	학생

⋮

SQL : Structured Query Language

- SQL은 사용자와 데이터베이스 시스템 간에 의사소통을 하기 위한 언어.

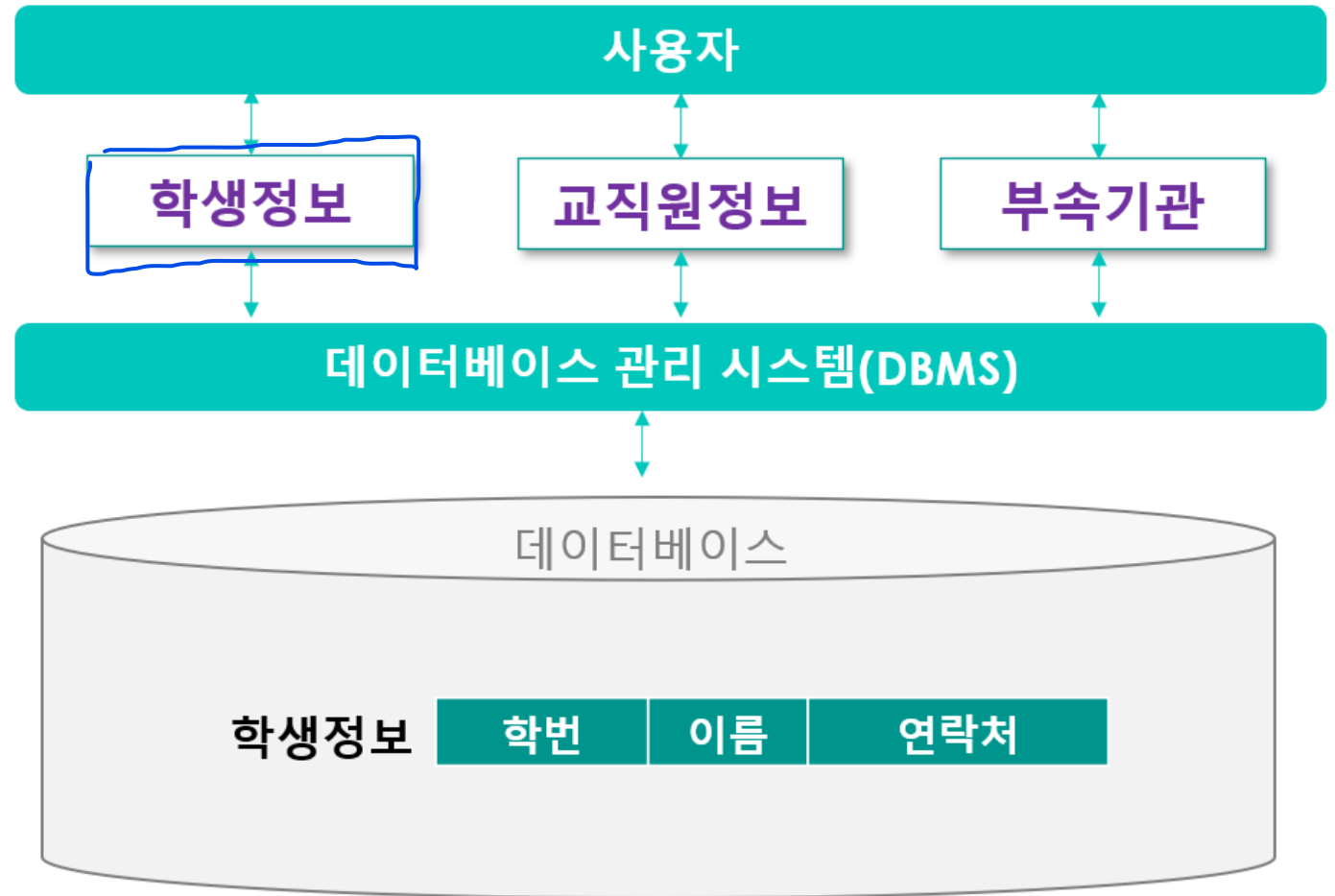


데이터베이스 만들기

<Hansung 데이터베이스 만들기>

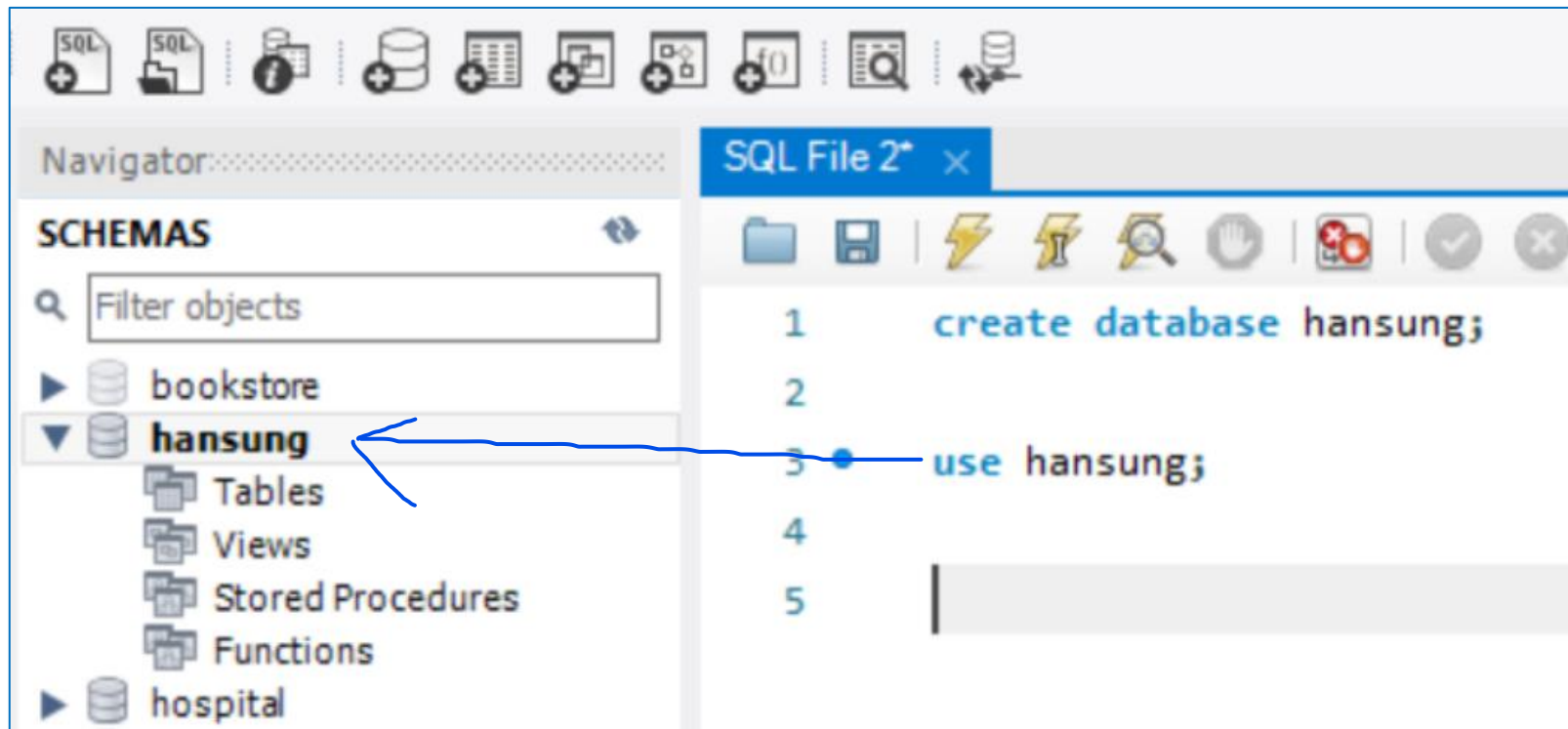
- 학교에는 학생정보, 교직원기록, 부속기관 등
- 학생정보에는 어떤 속성이 필요할까?

- 한성대학교 데이터베이스와
학생 정보 **테이블**을
만들어보자.



데이터베이스 만들기 실습

👉 실습은 오프라인 및 온라인 강의 내용 참고



SQL 명령어

• SQL (Structured Query Language)

- 관계형 데이터베이스 관리 시스템(RDBMS)의 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어, 사용자와 데이터베이스 시스템 간에 의사소통을 하기 위한 언어
- SQL 쿼리문은 데이터베이스에 저장된 값을 불러내기 위한 명령문
- 데이터베이스 관련 명령어와 테이블 관련 명령어로 구분되어 수행
- 일부 명령어를 제외하고 나머지 명령어에 대한 알파벳 대/소문자를 특별히 구분하지 않음

☞ 데이터베이스 관련 명령어

SQL 명령어	의미
CREATE	새로운 데이터베이스를 생성합니다.
DROP	생성되어 있는 데이터베이스를 삭제합니다.
USE	데이터베이스 목록에서 특정 데이터베이스를 선택합니다.

SQL 명령어의 종류

- 사용 방법

```
mysql> CREATE DATABASE 데이터베이스명;
```

```
mysql> DROP DATABASE 데이터베이스명;
```

```
mysql> USE 데이터베이스명;
```

SQL 명령어의 종류

※ show databases;

The screenshot shows a SQL IDE window titled 'SQL File 4*'. The SQL editor contains the following commands:

```
1 create database hansung;  
2  
3 • use hansung;  
4  
5 • show databases;  
6
```

The command 'show databases;' on line 5 is highlighted with a red box. Below the editor, the 'Result Grid' tab is active, displaying a list of databases in a table:

Database
bookstore
hansung
hospital
information_schema
madang
mysql
performance_schema
sakila
sys
testmydb
testmydb1
testmydb10
testmydb2
testmydb3
testmydb4
testmydb5
testmydb6
testmydb7
testmydb8
testmydb9
university
world

테이블 만들기

1. 한성대학교 데이터베이스 만들기

- 앞의 실습과 같이 한성대학교 데이터베이스 생성

INTEGER → INT 도 가능

속성
만들기

2. 한성대학교 데이터베이스 안에 [학생정보] 테이블 만들기

학생정보

학번

이름

연락처

```
CREATE TABLE 학생정보 (
  학번    INTEGER NOT NULL,
  이름    VARCHAR(20),
  연락처  VARCHAR(20),
  PRIMARY KEY(학번)
);
```

테이블명

열이름 저장공간 값의 허용(옵션) 구분값

열이름 저장공간 구분값

열이름 저장공간 구분값

열이름 저장공간 구분값

기본키

학생정보에서 학번을 PRIMARY KEY로 지정.

[참고] 데이터 타입 정리

- MySQL 데이터 타입

- RDBMS 별로 제공되는 데이터 타입이 다름 (예: TEXT는 오라클에서 제공되지 않음)

※ 문자형 데이터 타입

- 아래에서 n은 바이트 크기가 아닌, 글자 수를 의미
 - 예를 들어, 한글 표현을 위해 유니코드가 이용될 경우, 하나의 글자 표현에 2바이트 필요
 - 이 경우, CHAR(n) 유형으로 정의할 수 있는 최대 글자수는 128 ($=255/2$)
- TEXT는 65,535 바이트(MySQL 레코드의 최대 크기)보다 큰 데이터를 저장할 때 이용

👉 다음 페이지 내용도 확인

[참고] 자주 사용하는 데이터 타입

👉 일단 수업에서 활용한 다음 3개만 기억

데이터 타입	설명	ANSI SQL 표준 타입
INTEGER INT	4바이트 정수형	INTEGER, INT SMALLINT
NUMERIC(m,d) DECIMAL(m,d)	전체자리수 m, 소수점이하 자리수 d를 가진 숫자형	DECIMAL(p, s) NUMERIC[(p,s)]
CHAR(n)	문자형 고정길이, 문자를 저장하고 남은 공간은 공백으로 채운다.	CHARACTER(n) CHAR(n)
VARCHAR(n)	문자형 가변길이	CHARACTER VARYING(n) CHAR VARYING(n)
DATE	날짜형, 연도, 월, 날, 시간을 저장한다.	

👉 다음 페이지 내용도 확인

[참고] 추가 설명

구분	데이터 타입	최대 길이	저장 크기
문자열형	char	255	입력한 숫자 byte
	varchar	255	(입력한 숫자 + 1) byte
숫자형	int	-2147483648 ~ 2147483647	정수형 4 byte
	float	-3.402823466E+38 ~ -1.175494351E-38	실수형 8 byte
	double	-1.7976931348623157E+308 ~-2.2250738585072014E-308	실수형 8 byte

※ 이 외에도 굉장히 많은 자료형 존재

※ char 타입과 varchar 타입의 차이

char 타입은 고정폭 문자열로 최대 255자까지 문자를 저장할 수 있으며 varchar 타입은 가변폭 문자열로 최대 255자까지 문자를 저장할 수 있습니다. 2개 모두 255자까지 문자를 입력할 수 있다는 공통점은 가지고 있지만 메모리를 고정 크기로 확보하는 char 타입과 입력하는 문자열의 개수에 따라 메모리를 가변적으로 확보하는 varchar 타입의 차이가 있습니다. 메모리 효율성을 높이기 위해 가급적이면 문자열 데이터 타입은 varchar 타입으로 선언하는 경향이 더 많습니다.

[참고] 테이블 관련 SQL 명령어

👉 테이블 관련 명령어

SQL 명령어	의미
CREATE	새로운 테이블을 생성합니다.
DESC (또는 DESCRIBE)	테이블의 구조를 확인합니다.
ALTER	테이블의 구조를 변경합니다.
RENAME	테이블의 이름을 변경합니다.
DROP	테이블을 삭제합니다.

[참고] 테이블 관련 SQL 명령어

- DESC 또는 DESCRIBE 명령어 : SQL 쿼리문에서 테이블의 구조를 확인할 때 사용

```
mysql> DESC 테이블명;  
mysql> DESCRIBE 테이블명;
```

- 새로운 필드 추가하기 : ALTER 명령어와 옵션 ADD를 사용하여 테이블에 새로운 필드 추가

```
mysql> ALTER TABLE 테이블명 ADD 필드명 데이터 _ 타입 [AFTER 필드명];
```

- 기존 필드 삭제하기: DROP 명령어를 사용하여 기존 테이블에 존재하는 특정 필드 삭제

```
mysql> ALTER TABLE 테이블명 DROP 필드명;
```

[참고] 테이블 관련 SQL 명령어

- RENAME 명령어: SQL 쿼리문에서 테이블의 이름을 변경할 때 사용

```
mysql> ALTER TABLE 기존 _ 테이블명 RENAME 변경 _ 테이블명;
```

- DROP 명령어 : SQL 쿼리문에서 기존 테이블을 삭제할 때 사용

```
mysql> DROP TABLE 테이블명;
```

※ 사용 예) ALTER 명령어와 옵션 ADD를 사용하여 테이블에 새로운 필드 추가

The screenshot displays the SQL Developer interface. On the left, the 'Navigator' pane shows the 'SCHEMAS' tree. Under the 'hansung' schema, the 'Tables' folder is expanded, showing the '학생정보' table. The 'Columns' folder for '학생정보' is also expanded, and the '나이' column is highlighted with a red box. The main editor pane shows the SQL code for 'SQL File 5*'. The code includes a 'drop table' statement, a 'CREATE TABLE' statement for '학생정보' with columns '학번' (INT NOT NULL), '이름' (VARCHAR(20)), and '연락처' (VARCHAR(20)), and a primary key on '학번'. It also includes three 'INSERT INTO' statements for the '학생정보' table. The final statement, 'alter table 학생정보 add 나이 int;', is highlighted with a red box.

```
1
2 • drop table 학생정보;
3
4 • CREATE TABLE 학생정보 (
5     학번    INT NOT NULL,
6     이름    VARCHAR(20),
7     연락처  VARCHAR(20),
8     PRIMARY KEY(학번)
9 );
10
11 • INSERT INTO 학생정보 VALUES (20200001, '김한성', '010-1234-5678');
12 • INSERT INTO 학생정보 VALUES (20200002, '홍길동', '010-2345-6789');
13 • INSERT INTO 학생정보 VALUES (20200003, '박나래', '010-3456-7890');
14
15 • alter table 학생정보 add 나이 int;
16
```

테이블에 데이터 입력하기

참고로 릴레이션 스키마에서 primary key는 다른 속성을 나열하기 전에 먼저 나열하는 것이 일반적

[학생정보] 테이블에 데이터 입력하기

학생정보(학번, 이름, 연락처)

학생정보		
학번	이름	연락처
20200001	김한성	010-1234-5678

만들어진 학생정보 테이블에 한 행씩 자료를 입력하는 명령어는 **INSERT INTO 학생정보 VALUES ()** 괄호안에 속성 순서대로 자료를 입력하며, 문자형 데이터는 ' '로 표시를 한다.
계속해서 같은 방식으로 자료를 추가할 수 있다.

```
INSERT INTO 학생정보 VALUES (20200001, '김한성', '010-1234-5678');
```

테이블에 데이터 입력하기

[학생정보] 테이블에 데이터 입력하기

학생정보 (학번, 이름, 연락처)

학생정보

학번	이름	연락처
20200001	김한성	010-1234-5678
20200002	홍길동	010-2345-6789
20200003	박나래	010-2356-7890

다음과 같이 여러 데이터 입력 가능

INSERT INTO 학생정보 **VALUES** (20200001, '김한성', '010-1234-5678');

INSERT INTO 학생정보 **VALUES** (20200002, '홍길동', '010-2345-6789');

INSERT INTO 학생정보 **VALUES** (20200003, '박나래', '010-3456-7890');

※ 한성대학교 e-class에 (일괄 실행 가능한) 실습 파일을 업로드하였으니 참고하세요.

```
DROP DATABASE IF EXISTS HANSUNG;
```

```
CREATE DATABASE HANSUNG;
```

```
USE HANSUNG;
```

```
CREATE TABLE 학생정보 (  
    학번    INT,  
    이름    VARCHAR(20),  
    연락처  VARCHAR(20),  
    PRIMARY KEY(학번)  
);
```

```
INSERT INTO 학생정보 VALUES (20200001, '김한성', '010-1234-5678');
```

```
INSERT INTO 학생정보 VALUES (20200002, '홍길동', '010-2345-6789');
```

```
INSERT INTO 학생정보 VALUES (20200003, '박나래', '010-3456-7890');
```

```
alter table 학생정보 add 나이 int;
```



SQL 실습
파일 1

	학번	이름	연락처	나이
▶	20200001	김한성	010-1234-5678	NULL
	20200002	홍길동	010-2345-6789	NULL
	20200003	박나래	010-3456-7890	NULL

[참고]

- 참고로 데이터베이스 이름, 테이블 이름, 속성 이름 등은

(수업에서는 직관적인 이해를 위해 한글로 작성하였으나

Java 연동시 문제가 되는 경우가 있으니)

실제 활용할 때는 영어 이름으로 작성해 주시는 것이 좋습니다.



 **T h a n k y o u**

TECHNOLOGY

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Velit ex
plicabo ipsum, labore sed tempora ratione asperiores des
cenderat bore sed tempora rati jgert one bore sed tem!