

# SQL 기초 문법

## 데이터베이스와 테이블 관리

SQL 기초 문법 | SQL 코딩 테스트 준비하기

강사 나동빈

# SQL 기초 문법

데이터베이스와 테이블 관리

## SQL 기초 문법 데이터베이스와 테이블

## MySQL 접속하기

## SQL. 기초 다지기

- MySQL Command Line Client를 실행하여 다음과 같이 MySQL에 접속할 수 있다.
- MySQL Workbench를 사용해도 동일하게 실습을 진행할 수 있다.

```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

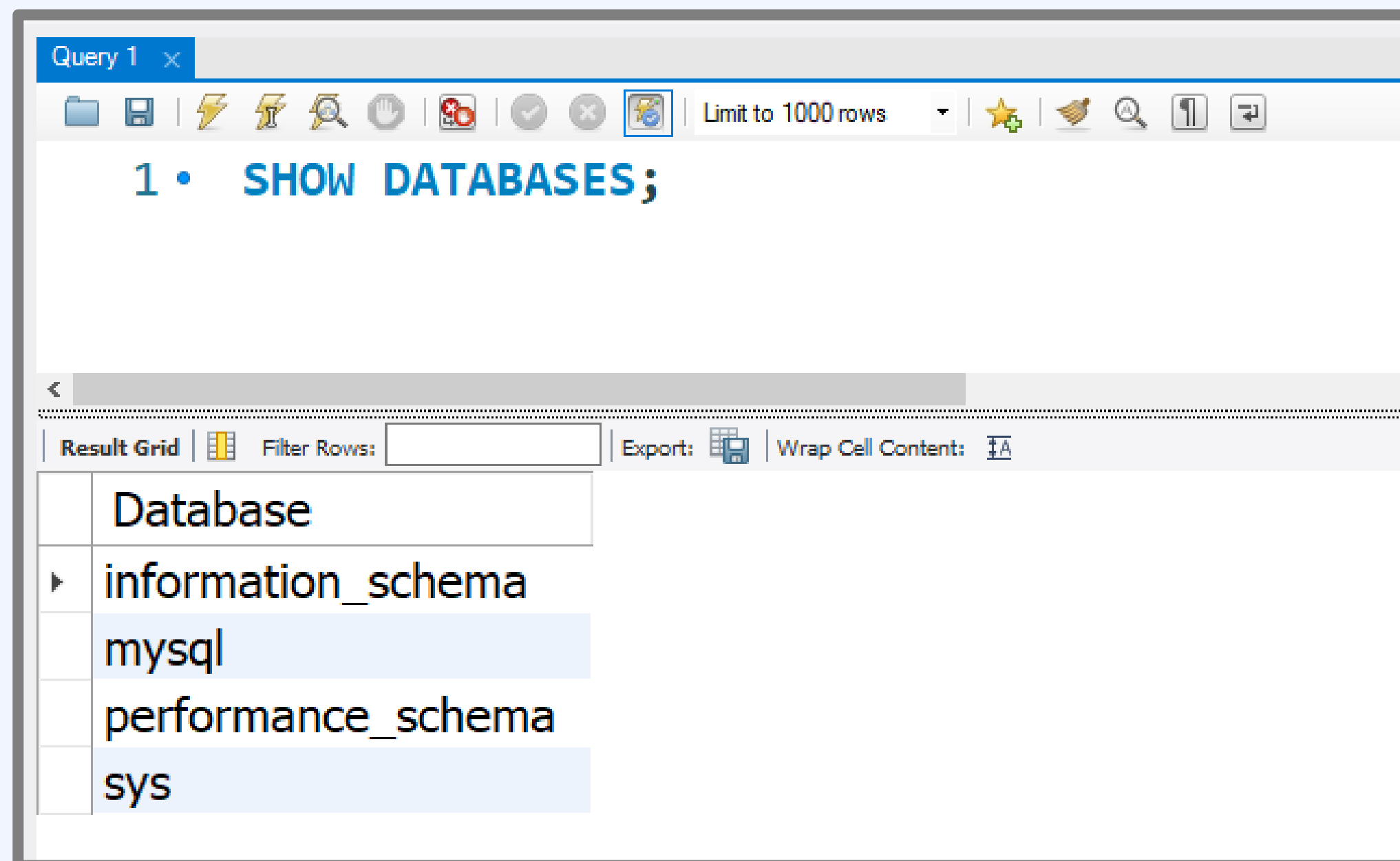
## SQL 기초 문법 데이터베이스와 테이블

## 데이터베이스 조회

**SQL.**  
기초 다지기

- SHOW 명령어를 이용해 데이터베이스를 조회할 수 있다.

```
SHOW DATABASES;
```



## SQL 기초 문법 데이터베이스와 테이블

## 데이터베이스 생성

SQL.  
기초 다지기

- CREATE 명령어를 사용하여 데이터베이스를 생성할 수 있다.
- 한국어를 사용하기 위해 UTF-8 인코딩(encoding) 설정을 기입한다.

```
CREATE DATABASE `데이터베이스 이름` CHARACTER SET utf8 COLLATE utf8_general_ci;
```

- MySQL에서는 기본적으로 데이터베이스를 생성하고 제거하는 명령어는 한 번 수행되면 영구적으로 적용되므로, 다음에 접속할 때에도 유지된다.
- 데이터베이스 및 테이블 이름을 지을 때는 그레이브(`) 기호를 사용할 수 있다.

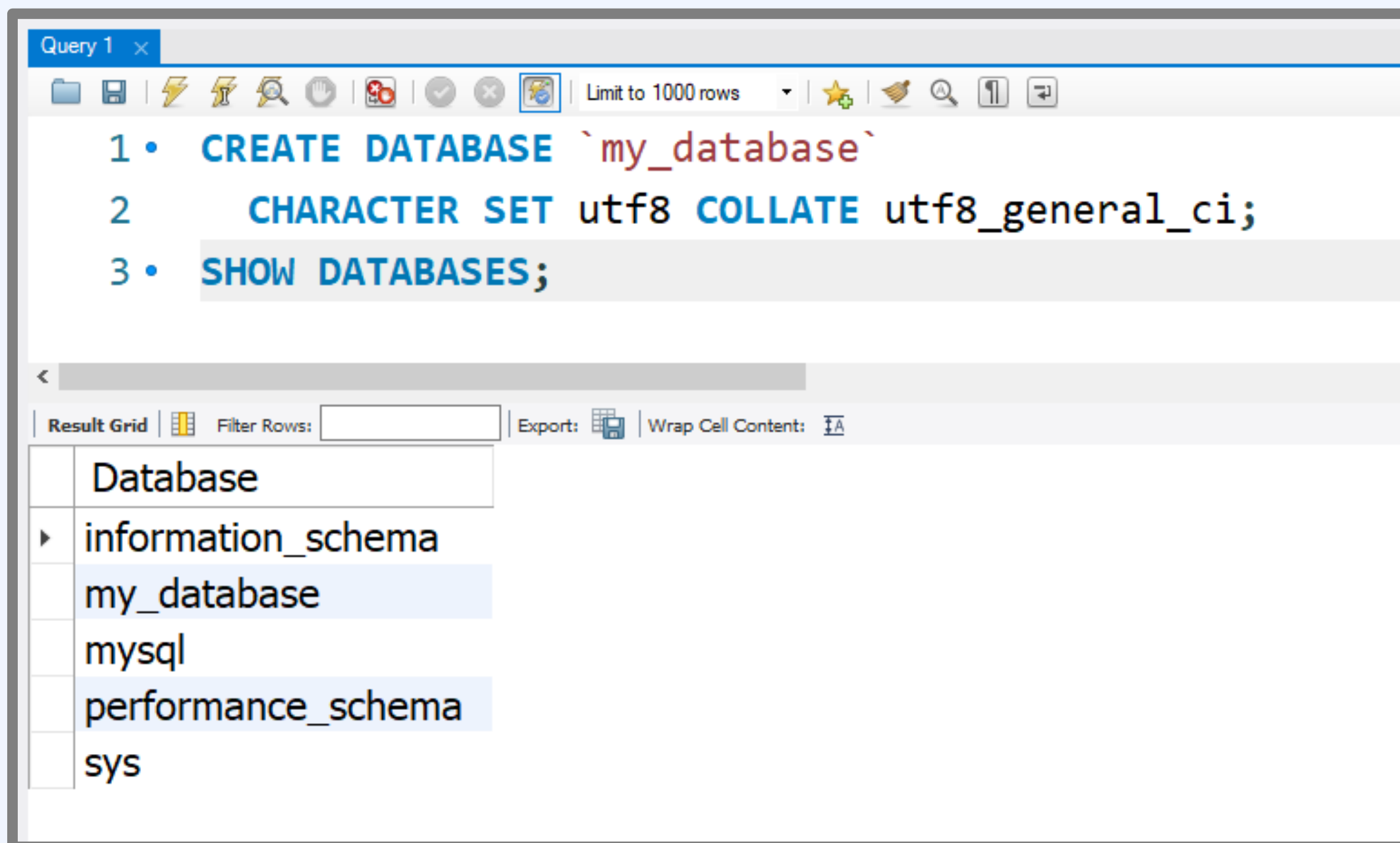
## SQL 기초 문법 데이터베이스와 테이블

## 데이터베이스 생성

## SQL. 기초 다지기

- 데이터베이스 생성의 구체적인 예시는 다음과 같다.

```
CREATE DATABASE `my_database` CHARACTER SET utf8 COLLATE utf8_general_ci;
```



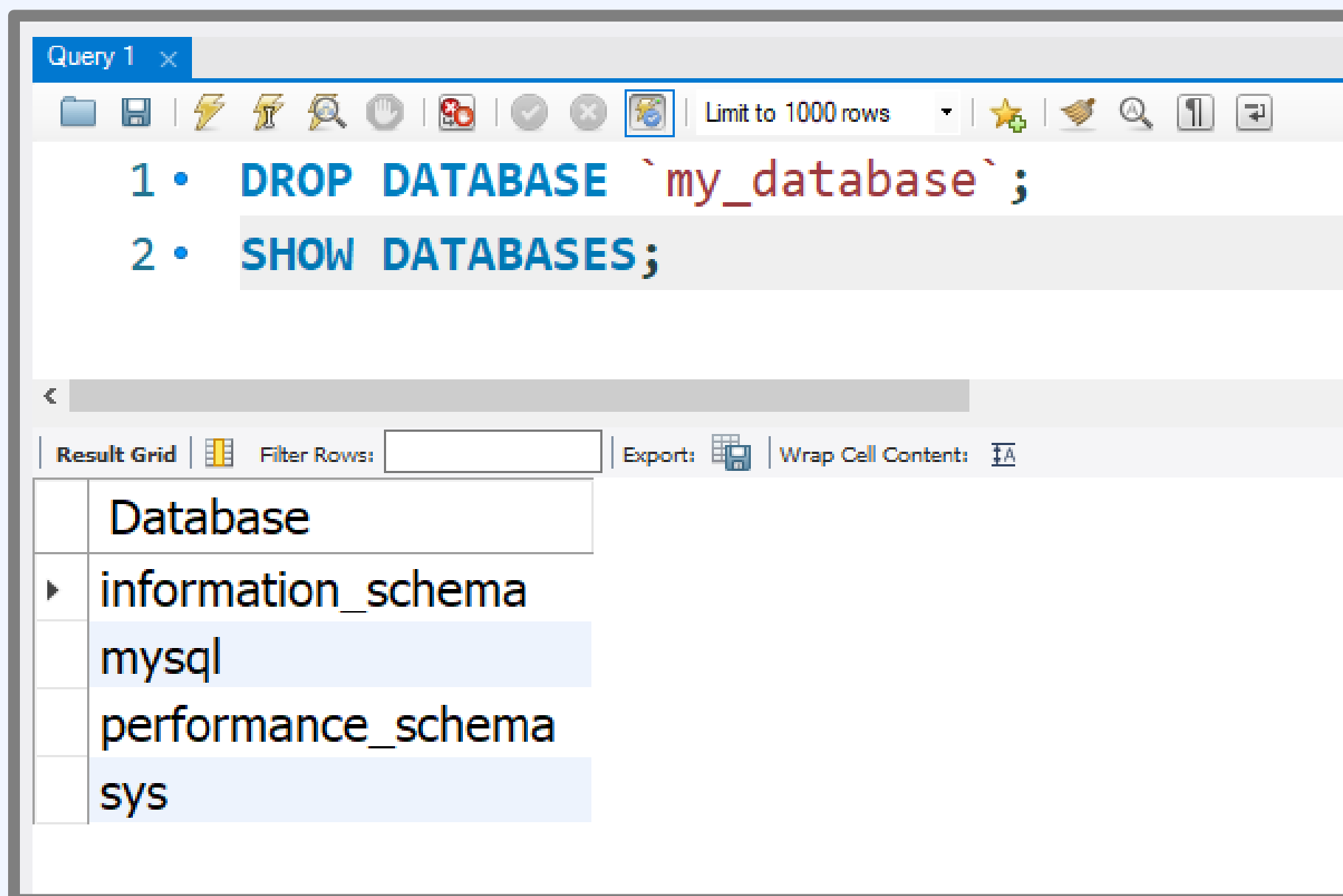
## SQL 기초 문법 데이터베이스와 테이블

## 데이터베이스 삭제

## SQL. 기초 다지기

- DROP 명령어를 사용해 데이터베이스를 삭제할 수 있다.

```
DROP DATABASE `데이터베이스 이름`;
```



## SQL 기초 문법 데이터베이스와 테이블

# 원하는 데이터베이스 사용하기

SQL.  
기초 다지기

- 데이터를 추가/수정하기 위해 데이터베이스에 접속할 때는 USE 명령어를 사용한다.
- 현재 세션(session)에서 특정 데이터베이스에 접속하여 명령어를 사용할 수 있다.

```
USE `데이터베이스 이름`;
```



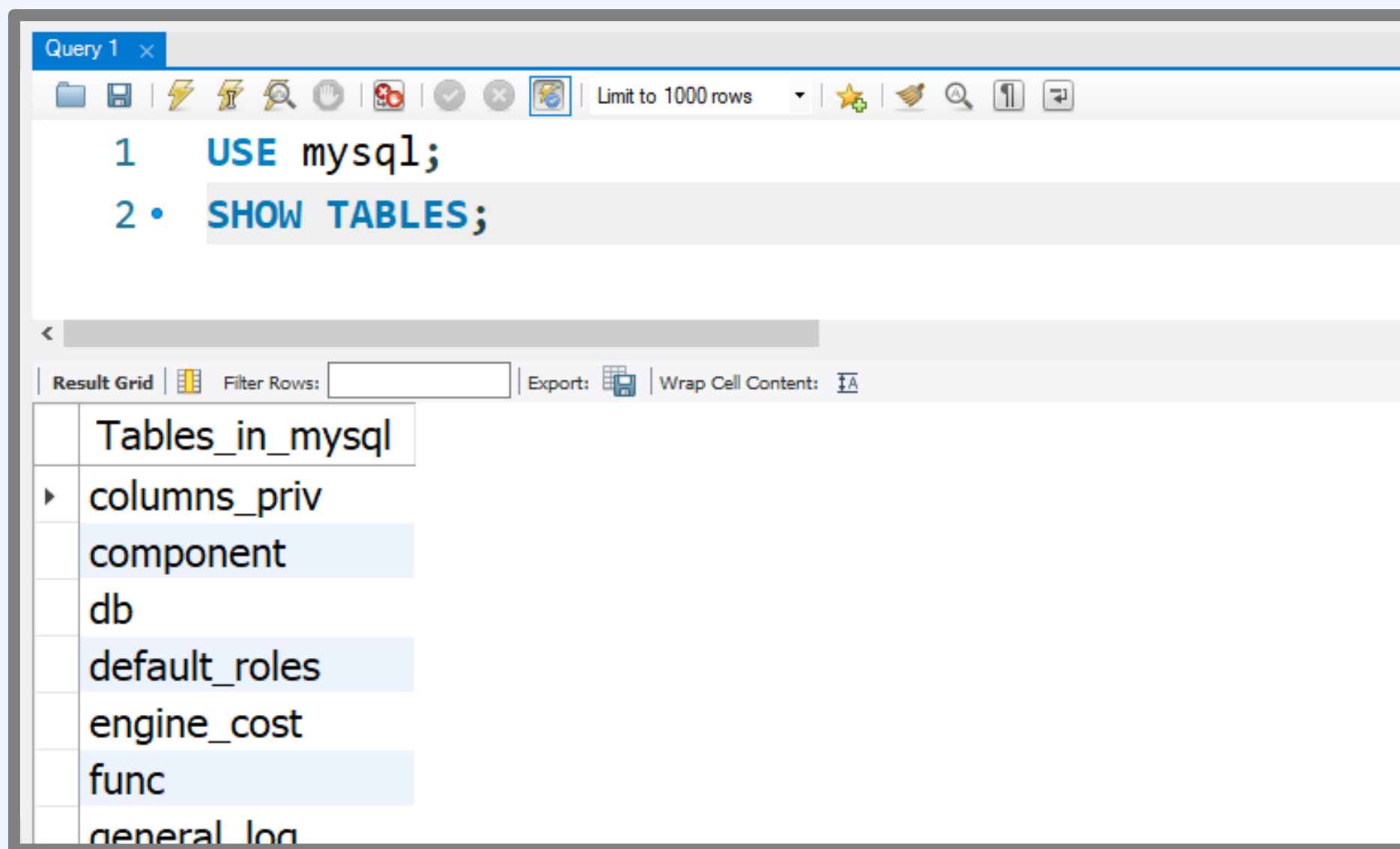
## SQL 기초 문법 데이터베이스와 테이블

## 원하는 데이터베이스 사용하기

SQL.  
기초 다지기

- 구체적인 예시는 다음과 같다.

```
USE mysql;  
SHOW TABLES;
```



## SQL 기초 문법 데이터베이스와 테이블

## MySQL 코드 컨벤션(conventions)

SQL.  
기초 다지기

- SELECT, INSERT, WHERE 등 예약어는 대문자를 사용한다.
- 기본적으로 테이블(table), 열(column)에 소문자를 사용한다.
- 단어를 축약하지 않는다: employee\_date (O), emp\_date (X)
- 능동태를 사용한다: update\_student (O), updated\_student (X)
- 라인별로 띄어쓰기 2칸 혹은 4칸을 사용하거나 탭(tab)을 사용한다.
- 명령어 마지막에는 세미콜론(;)을 사용한다.

## SQL 기초 문법 데이터베이스와 테이블

## MySQL 코드 컨벤션(conventions)

SQL.  
기초 다지기

- MySQL 코드 컨벤션을 따르는 대표적인 코드 예시는 다음과 같다.

```
SELECT
    student_id,
    name,
    age,
    grade
FROM student
WHERE student_id = 3 OR student_id = 7
    OR (student_id >= 10 AND student_id <= 15);
```

## SQL 기초 문법 데이터베이스와 테이블

## 테이블(Table)

**SQL.**  
기초 다지기

- 하나의 데이터베이스는 여러 개의 테이블을 가진다.
- 하나의 테이블은 엑셀 시트(Sheet)와 유사한 형태를 보인다.
- 테이블 내 각 데이터는 행(row)과 열(column)이 만나는 지점에 데이터가 들어간다.
- 행(row) = 레코드(record) = 튜플(tuple)
- 열(column) = 필드(field) = 속성(attribute)

학생 번호	학생 이름	학생 성적
1	홍길동	75
2	나동빈	83
3	이순신	95
4	임꺽정	88

## SQL 기초 문법 데이터베이스와 테이블

## 테이블 및 컬럼 컨벤션(conventions)

SQL.  
기초 다지기

- 테이블(table)의 이름은 단수형을 사용한다.
- 이름을 붙일 때는 snake case를 사용한다: student, retired\_employee
- 테이블의 key는 {테이블 이름의 단수형}\_id 형태를 쓴다: student\_id (0) 혹은 단순히 id (0)

### [student 테이블]

- 속성 1: student\_id
- 속성 2: name
- 속성 3: age
- 속성 4: grade

## SQL 기초 문법 데이터베이스와 테이블

## 테이블(Table) 생성

SQL.  
기초 다지기

- 테이블을 정의할 때는 CREATE 명령어를 사용한다.

```
CREATE TABLE `테이블 이름`  
(  
    {컬럼명 1} {자료형 1},  
    {컬럼명 2} {자료형 2},  
    {컬럼명 3} {자료형 3},  
    ...  
);
```

## SQL 기초 문법 데이터베이스와 테이블

## 테이블(Table) 생성

SQL.  
기초 다지기

- 한 명의 학생에 대한 테이블(table)을 다음과 같이 생성할 수 있다.

```
CREATE TABLE student (  
    student_id INT PRIMARY KEY,  
    student_name VARCHAR(20) NOT NULL,  
    student_birth_date DATE NOT NULL  
);
```

## SQL 기초 문법 데이터베이스와 테이블

## 테이블(Table) 생성

## SQL. 기초 다지기

- 한 명의 학생에 대한 테이블(table)을 다음과 같이 생성할 수 있다.

The screenshot shows a SQL query editor window titled 'Query 1'. The query text is as follows:

```

1 CREATE TABLE student (
2     student_id INT PRIMARY KEY,
3     student_name VARCHAR(20) NOT NULL,
4     student_birth_date DATE NOT NULL
5 );
6 • DESC student;
7

```

Below the query editor, the 'Result Grid' is displayed, showing the table structure:

Field	Type	Null	Key	Default	Extra
student_id	int	NO	PRI		
student_name	varchar(20)	NO			
student_birth_date	date	NO			



## SQL 기초 문법 데이터베이스와 테이블

# 테이블의 구조 확인하기

SQL.  
기초 다지기

- 테이블의 구조를 확인할 수 있다.
- 이때 구조를 스키마(scheme)라고도 부른다.

```
DESC 테이블명;
```

- NOT NULL: NULL 값 비허용, 중복 허용
- UNIQUE: NULL 값 허용, 중복 비허용
- PRIMARY KEY: NULL 비허용, 중복 비허용, 테이블당 하나씩
- DEFAULT: 해당 컬럼의 기본 값을 설정

```
CREATE TABLE student
(
    student_id INT PRIMARY KEY,
    student_phone VARCHAR(20) UNIQUE,
    student_name VARCHAR(20) NOT NULL,
    student_address VARCHAR(20) DEFAULT 'seoul'
);
```

## SQL 기초 문법 데이터베이스와 테이블

# 테이블 제약 조건(Constraint) - 외래키

SQL.  
기초 다지기

- 특정한 학생(student)이 다른 강의(lecture)를 등록하는 상황을 고려하자.
- 이때 등록(registration) 테이블은 "학생 번호"와 "강의 번호"를 참조해야 한다.
- 이러한 기능을 위해 외래키(foreign key)가 제공된다.

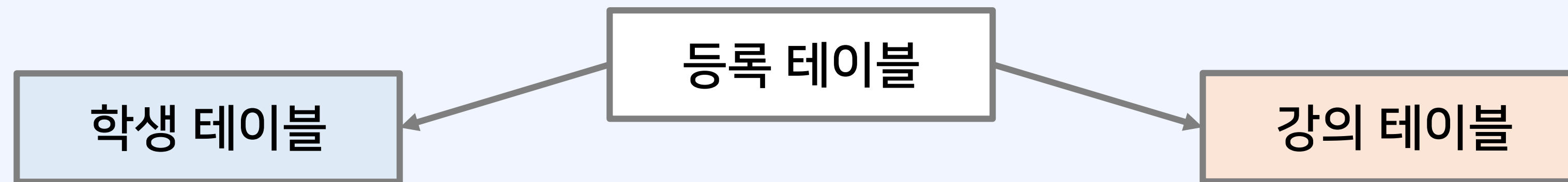


## SQL 기초 문법 데이터베이스와 테이블

## 테이블 제약 조건(Constraint) – 외래키

SQL.  
기초 다지기

- 등록(registration) 테이블은 "학생 번호"와 "강의 번호"를 참조해야 한다.
- 만약 [학생 테이블]에 없는 학생을 추가하려는 경우 오류가 발생한다.



존재하지 않는 학생인 경우

- 학생 번호: 338283
- 강의 번호: 5
- 등록 날짜: 2012-05-05

오류 발생!

- 외래키(foreign key)는 하나의 테이블이 다른 테이블을 참조할 때 사용한다.
- 학생(student) ← 등록(registration) → 강의(lecture)

```
CREATE TABLE registration
(
    student_id INT,
    lecture_id INT,
    date DATETIME,
    FOREIGN KEY (student_id) REFERENCES student(student_id),
    FOREIGN KEY (lecture_id) REFERENCES lecture(lecture_id)
);
```

## SQL 기초 문법 데이터베이스와 테이블

# DROP 명령어 사용하기

SQL.  
기초 다지기

- DROP 명령어로 테이블을 삭제할 수 있다.

```
DROP TABLE student;
```

## SQL 기초 문법 데이터베이스와 테이블

## IF EXISTS 사용하기

SQL.  
기초 다지기

- 만약에 테이블이 존재하는 경우라면 IF EXISTS를 사용할 수 있다.

```
DROP TABLE IF EXISTS student;
```

## SQL 기초 문법

### 데이터베이스와 테이블

## 테이블 수정

**SQL.**  
기초 다지기

- 테이블을 수정할 때는 ALTER를 사용할 수 있다.
  - 1) ADD: 새로운 필드를 추가할 때 사용
  - 2) DROP: 기존의 필드를 제거할 때 사용
  - 3) MODIFY: 기존의 필드를 변경할 때 사용



## SQL 기초 문법 데이터베이스와 테이블

## 테이블 수정

## SQL. 기초 다지기

- ALTER를 이용해 특정한 테이블에 새로운 필드를 추가할 수 있다.

```
ALTER TABLE student ADD student_age INT;
```

## SQL 기초 문법 데이터베이스와 테이블

## 테이블 수정

## SQL. 기초 다지기

- ALTER를 이용해 특정한 테이블의 기존 필드를 제거할 수 있다.

```
ALTER TABLE student DROP student_age;
```

## SQL 기초 문법 데이터베이스와 테이블

## 테이블 수정

## SQL. 기초 다지기

- ALTER를 이용해 특정한 테이블의 기존 필드 타입을 수정할 수 있다.

```
ALTER TABLE student MODIFY COLUMN student_phone VARCHAR(30);
```