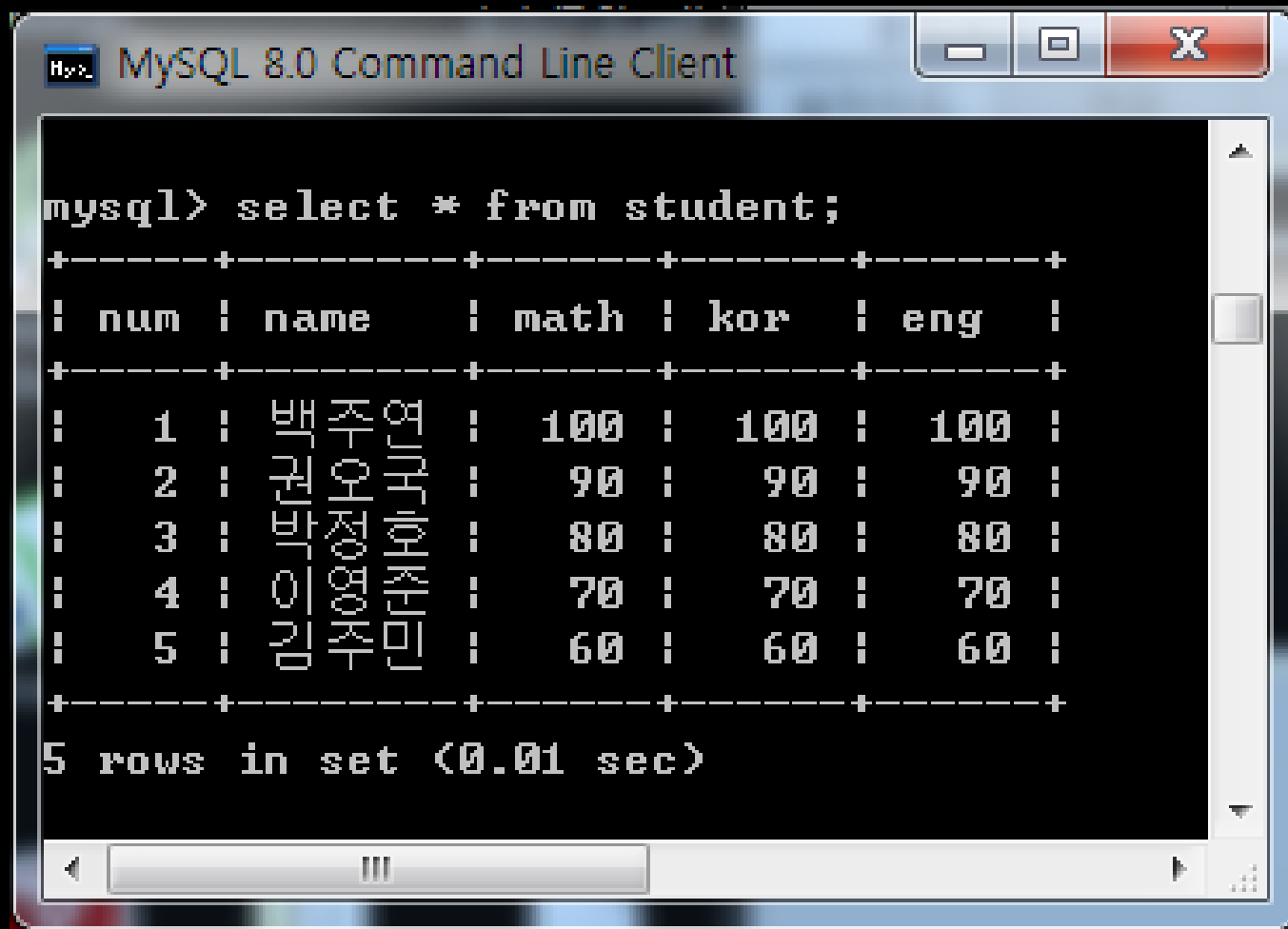


# 데이터베이스

## 2차원 배열 내부에 값을 저장하는 방식



```
mysql> select * from student;
```

num	name	math	kor	eng
1	백주영	100	100	100
2	권아름	90	90	90
3	박정호	80	80	80
4	이재민	70	70	70
5	김주민	60	60	60

```
5 rows in set (0.01 sec)
```

# mysql

---

1. 무료
2. 관계형 데이터베이스
3. 리눅스, 유닉스, 윈도우 등 거의 모든 운영체제에서 사용가능
4. 처리속도가 상당히 빠름
5. 보안성이 우수
6. 설치가 쉽고 초보자도 빠르게 배울 수 있음

# 실행방법 1

## 시작표시줄의 아이콘으로 실행

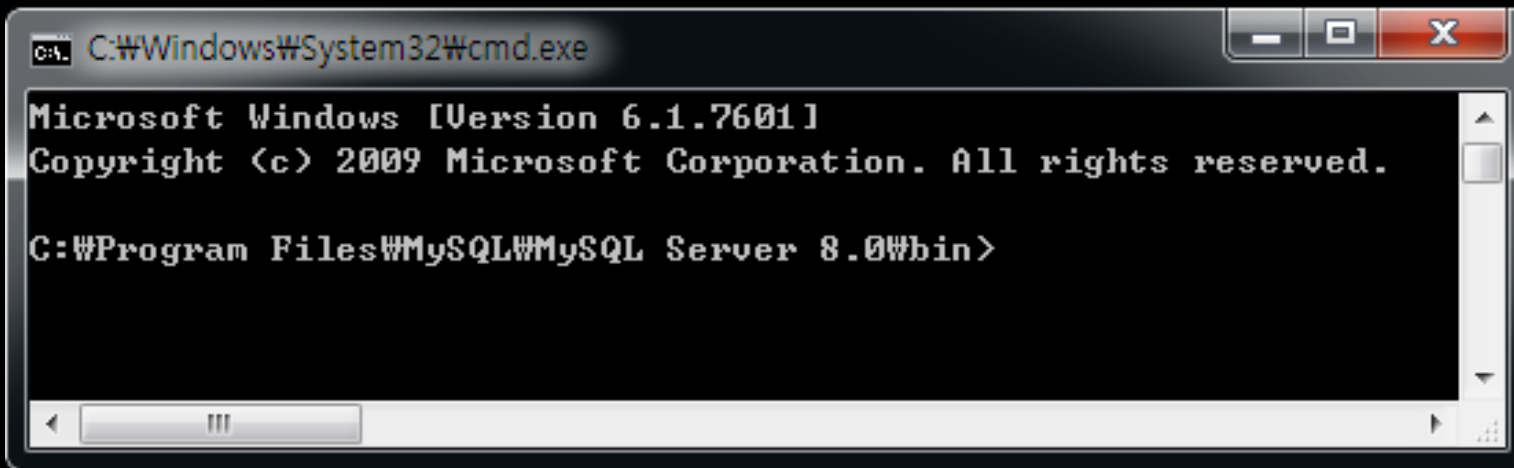


# 실행방법 2

## 시작표시줄에 아이콘이 없는 경우



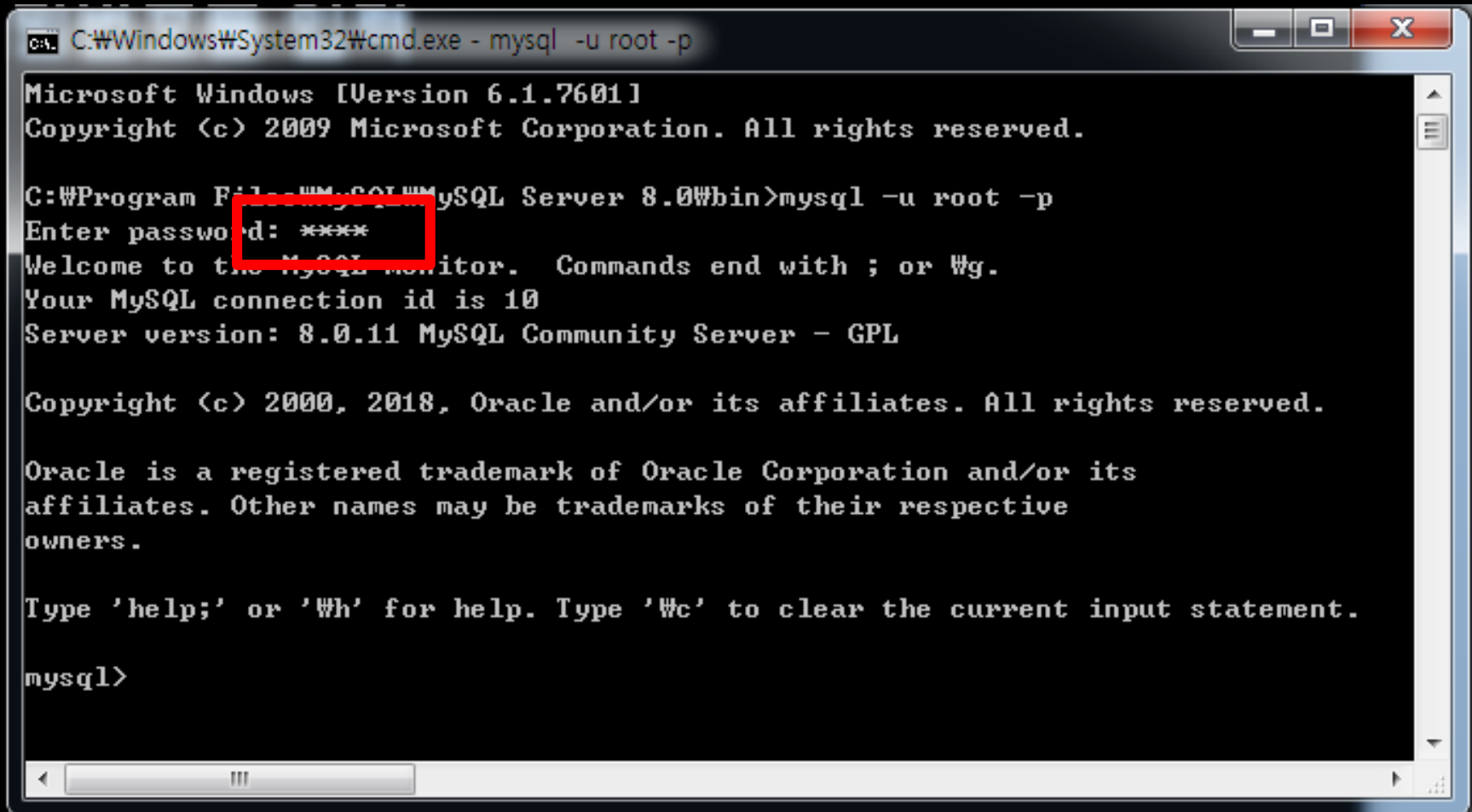
## 위의 설치 경로에서 CMD실행



**Mysql -u root -p 입력**

# 실행방법 3

## 비밀번호를 입력



```
C:\Windows\System32\cmd.exe - mysql -u root -p

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.11 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

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>
```

# 데이터베이스 생성

---

Create database [데이터베이스 이름];

```
mysql> create database sample;
```

```
Query OK, 1 row affected (0.06 sec)
```

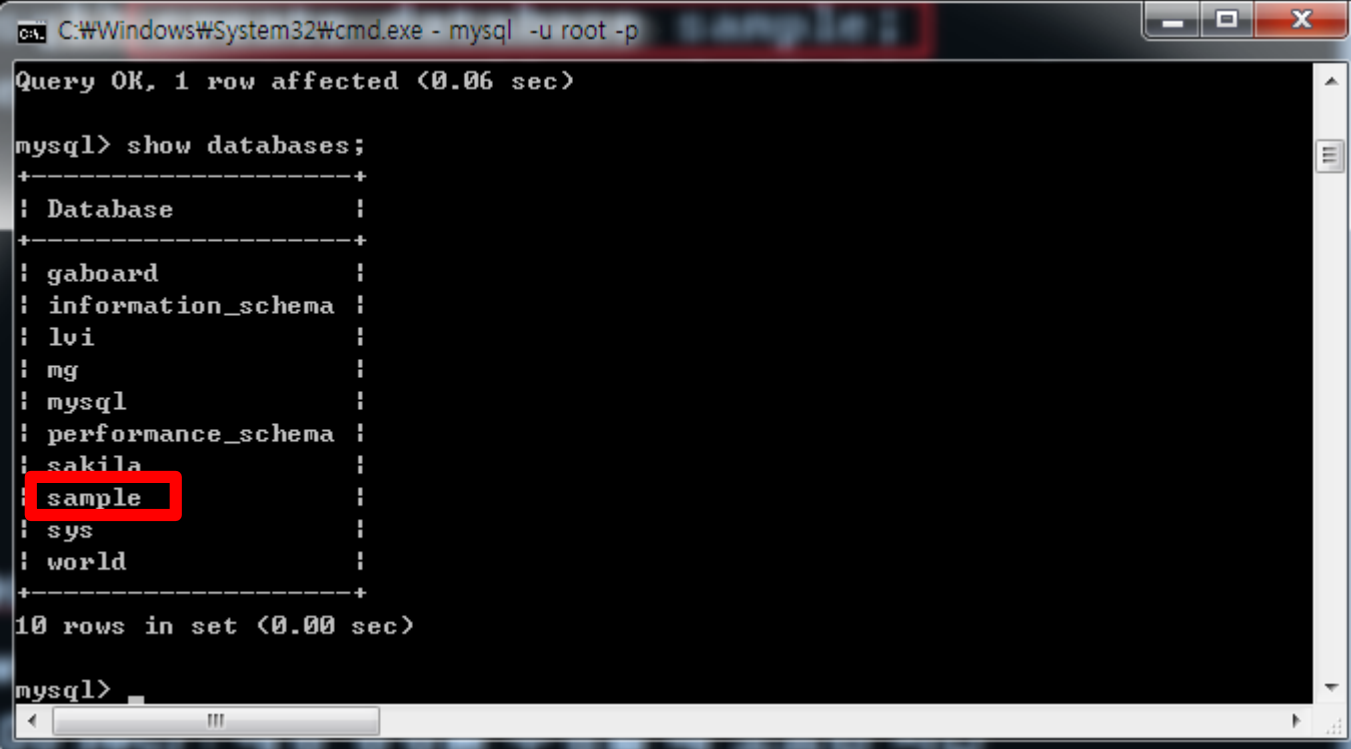
Query : 질의어, 컴퓨터에게 제시한 데이터베이스 명령문

row : 한줄, 해당 명령어가 한 줄에 영향을 미쳤음을 알려줌

(0.06 sec) : 해당 명령을 수행할 때 소요된 시간

# 데이터베이스 확인

show databases;



```
C:\Windows\System32\cmd.exe - mysql -u root -p sample1
Query OK, 1 row affected (0.06 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| gaboard  |
| information_schema |
| lvi      |
| mg       |
| mysql    |
| performance_schema |
| sakila   |
| sample   |
| sys      |
| world    |
+-----+
10 rows in set (0.00 sec)

mysql>
```

set : 질의어를 통해 만들어진 답변

row : 한줄, 단변이 총 10개 줄로 만들어져 있음을 의미

(0.00 sec) : 소수점 둘째자리 이하 무시

# 데이터베이스 삭제

**drop database [데이터베이스 이름];**

```
mysql> drop database sample;  
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| gaboard  |  
| information_schema |  
| lvi      |  
| mg       |  
| mysql    |  
| performance_schema |  
| sakila   |  
| sys      |  
| world    |  
+-----+  
9 rows in set (0.00 sec)
```

**데이터베이스가 사라진것을 확인**

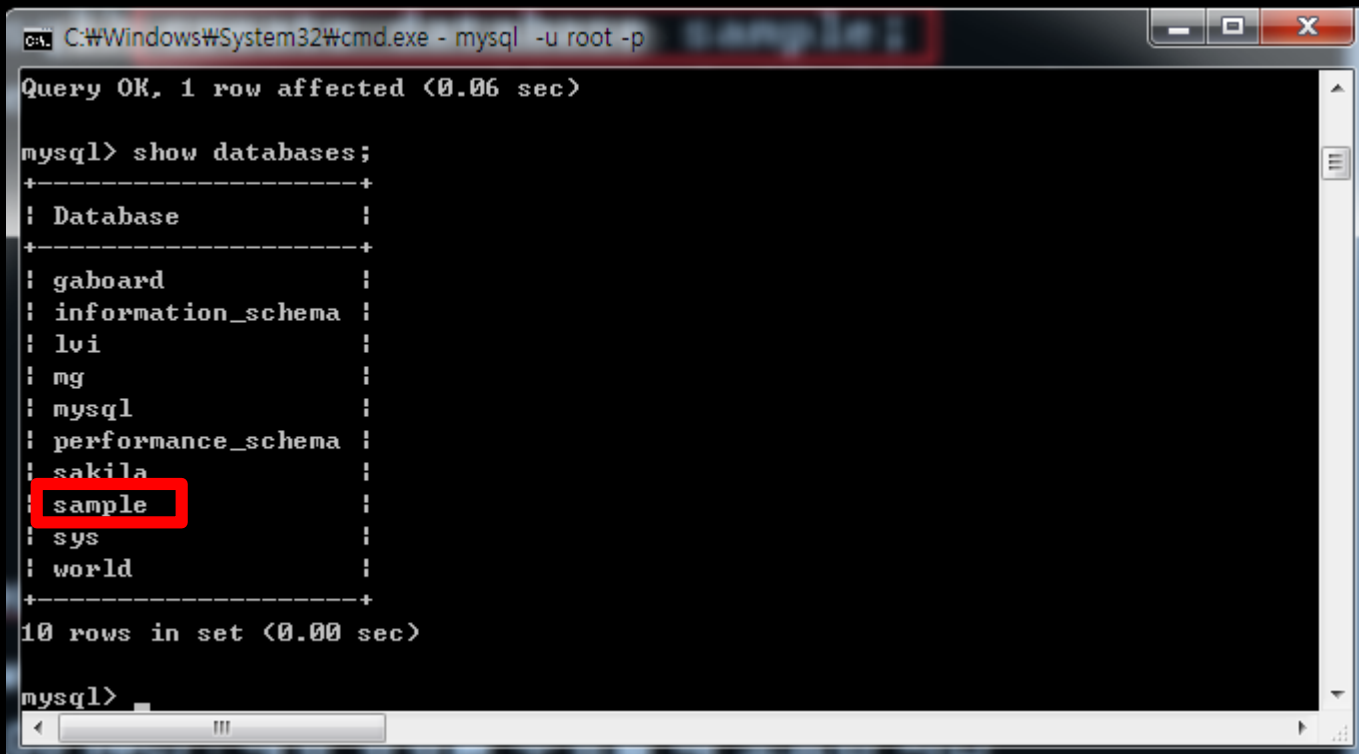
```
mysql> _
```



# 실습

이름이 sample인 데이터베이스를 제작

잘 생성되었는지를 확인



The screenshot shows a Windows command prompt window with the title bar "C:\Windows\System32\cmd.exe - mysql -u root -p sample". The command prompt displays the output of the MySQL command "show databases;". The output is a table with 10 rows and 1 column, listing the databases: gaboard, information\_schema, lvi, mg, mysql, performance\_schema, sakila, sample, sys, and world. The "sample" database is highlighted with a red rectangle. The command prompt also shows the status "Query OK, 1 row affected (0.06 sec)" and "10 rows in set (0.00 sec)".

```
C:\Windows\System32\cmd.exe - mysql -u root -p sample
Query OK, 1 row affected (0.06 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| gaboard  |
| information_schema |
| lvi      |
| mg       |
| mysql    |
| performance_schema |
| sakila   |
| sample   |
| sys      |
| world    |
+-----+
10 rows in set (0.00 sec)

mysql>
```

# 데이터베이스 사용하기

---

Use [데이터베이스 이름];

```
mysql> use sample;  
Database changed  
mysql> _
```

Database changed : 데이터베이스 변경이 잘 되었음을 의미

# 테이블 만들기 1

---

```
Create table [테이블 이름](  
    필드명1 자료형(자료크기),  
    필드명2 자료형(자료크기),  
);
```

Varchar() : JAVA의 String과 같은 자료형  
Int() : JAVA의 int와 같은 자료형

# 테이블 만들기 2

```
mysql> create table student(  
  -> num int(11) primary key auto_increment,  
  -> name varchar(11),  
  -> math int(11),  
  -> kor int(11),  
  -> eng int(11)  
  -> );
```

```
Query OK, 0 rows affected (0.07 sec)
```

Cmd는 ;를 만나기 전까지는 명령어가 완성된것으로 보지 않아  
계속적으로 입력하라고 ->표시를 보여주도록 되어 있음  
한줄로 적어도 무관하나 여러줄로 나누는 것이 오류가 발생한 지  
점을 특정하기 편하므로 줄을 나누어주는 것이 편리

Primary key : 중복값을 가질 수 없는 고유한 속성  
Auto\_increment : 자동으로 1씩 증가하는 속성

# 테이블 확인하기

---

show table;

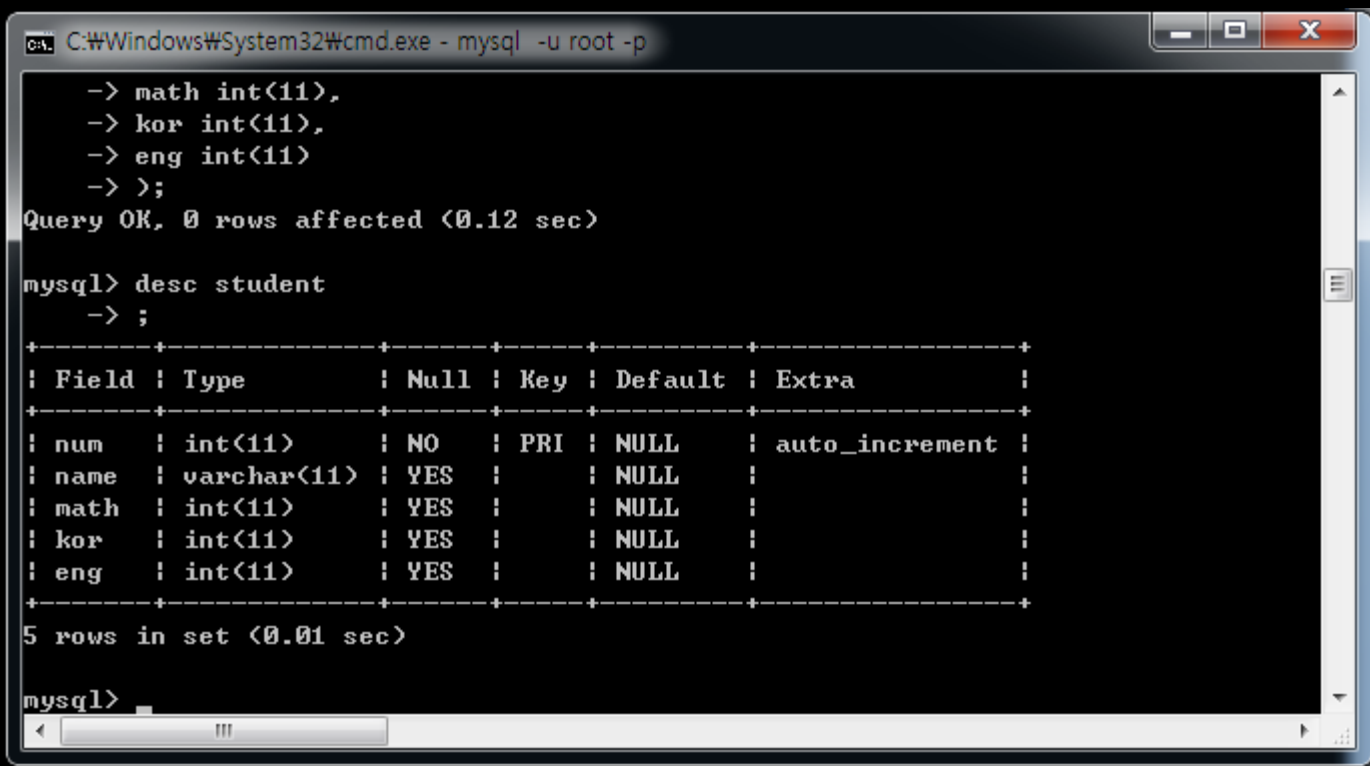
```
mysql> show tables;
+-----+
! Tables_in_sample !
+-----+
! student          !
+-----+
1 row in set (0.01 sec)

mysql>
```

---

# 테이블 구조 확인하기

**desc [테이블 이름];**



```
C:\Windows\System32\cmd.exe - mysql -u root -p

-> math int(11),
-> kor int(11),
-> eng int(11)
-> ;

Query OK, 0 rows affected (0.12 sec)

mysql> desc student
-> ;

+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| num   | int(11)       | NO   | PRI | NULL    | auto_increment |
| name  | varchar(11)   | YES  |     | NULL    |                |
| math  | int(11)       | YES  |     | NULL    |                |
| kor   | int(11)       | YES  |     | NULL    |                |
| eng   | int(11)       | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+

5 rows in set (0.01 sec)

mysql>
```

**Field** : 한 줄의 제목을 의미

**Type** : 해당 자료의 형태와 크기를 의미

**Null** : 공백으로 값을 지정할 수 있는지의 여부

**Default** : 값이 지정되지 않을 경우 넣어줄 값을 의미

# 테이블 삭제하기

---

**drop table [테이블 이름];**

```
mysql> drop table student;
```

```
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> show tables;
```

```
Empty set (0.00 sec)
```

```
mysql>
```

---

**Empty set : 쿼리의 결과가 아무것도 없는 경우**

# 실습

다음과 같은 구조를 가지는  
student 테이블을 제작

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	auto_increment
name	varchar(11)	YES		NULL	
math	int(11)	YES		NULL	
kor	int(11)	YES		NULL	
eng	int(11)	YES		NULL	

테이블의 내부 구조를 확인



# 테이블에 데이터 넣기

---

Insert into [테이블 이름](  
필드명1, 필드명2, 필드명3) values (  
값1, 값2, 값3);

```
mysql> insert into student(name, math, kor, eng) values("홍길동", 100,100,100);  
Query OK, 1 row affected (0.08 sec)
```

Num은 auto\_increment속성으로 자동으로 값이 1씩 증가

# 테이블 내 데이터 확인

Select [필드명] from [테이블 이름];

```
mysql> select num, name from student;
```

```
+-----+-----+
```

```
| num | name |
```

```
+-----+-----+
```

```
| 1 | 홍길동 |
```

```
+-----+-----+
```

```
1 row in set (0.00 sec)
```

\* 모든 필드를 보여달라는 의미

```
mysql> select * from student;
```

```
+-----+-----+-----+-----+-----+
```

```
| num | name | math | kor | eng |
```

```
+-----+-----+-----+-----+-----+
```

```
| 1 | 홍길동 | 100 | 100 | 100 |
```

```
+-----+-----+-----+-----+-----+
```

```
1 row in set (0.00 sec)
```

# 실습

다음과 같은 값을 student 테이블에 입력

```
mysql> select * from student;
```

num	name	math	kor	eng
1	홍길동	100	100	100
2	홍상진	98	70	55
3	홍인형	77	80	65
4	백은도사	90	90	90
5	차돌바위	80	85	10
6	전석부리	50	50	50
7	전단이	60	65	40
8	문순이	90	95	80

8 rows in set (0.00 sec)

키보드 ↑↓를 통하여 이전에 작성한 문장을 불러올 수 있다

# 테이블 내 데이터 검색

Select [필드명] from [테이블 이름]  
Where [조건];

```
mysql> select * from student where kor>90;
```

num	name	math	kor	eng
1	홍길동	100	100	100
8	돌순이	90	95	80

```
2 rows in set (0.00 sec)
```

국어점수가 90점보다 큰 학생을 검색하는 쿼리문

# 테이블 내 데이터 삭제

`delete from [테이블 이름] Where [조건];`

```
mysql> delete from student where name="돌순이";  
Query OK, 1 row affected (0.09 sec)
```

```
mysql> select * from student;
```

num	name	math	kor	eng
1	홍길동	100	100	100
2	홍상직	98	70	55
3	홍인형	77	80	65
4	백운도사	90	90	90
5	차돌바위	80	85	10
6	덥석부리	50	50	50
7	곱단이	60	65	40

```
7 rows in set (0.00 sec)
```

돌순이가 사라진것을 확인

# 테이블 내 데이터 정렬

Select [필드명] from [테이블명]  
where [조건] order by [기준];

```
mysql> select * from student where math>70 order by math asc;
```

num	name	math	kor	eng
3	홍인형	77	80	65
5	차돌바위	80	85	10
4	백운도사	90	90	90
2	홍상직	98	70	55
1	홍길동	100	100	100

```
5 rows in set (0.00 sec)
```

Asc : 기준되는 필드를 오름차순으로 표현

Desc : 기준되는 필드를 내림차순으로 표현

# 실습

다음과 같이 student 테이블을 검색

num	name	math	kor	eng
1	홍길동	100	100	100
4	백운도사	90	90	90
3	홍인형	77	80	65
2	홍상직	98	70	55
5	차돌바위	80	85	10

수학이 70점 이상인 사람을 기준으로 검색

# 값 변화

---

**update [테이블 이름] set [필드명]=[값]  
Where [조건];**

```
mysql> update student set math=1 where name="홍길동";  
Query OK, 1 row affected (0.03 sec)  
Rows matched: 1    Changed: 1    Warnings: 0
```

**matched :** 매칭되는 결과값이 몇 개인지를 확인

**Changed :** 변경된 결과값이 몇 개인지를 확인

**Warnings :** 오류가 발생할 수 있는 경우가 몇가지인지를 확인