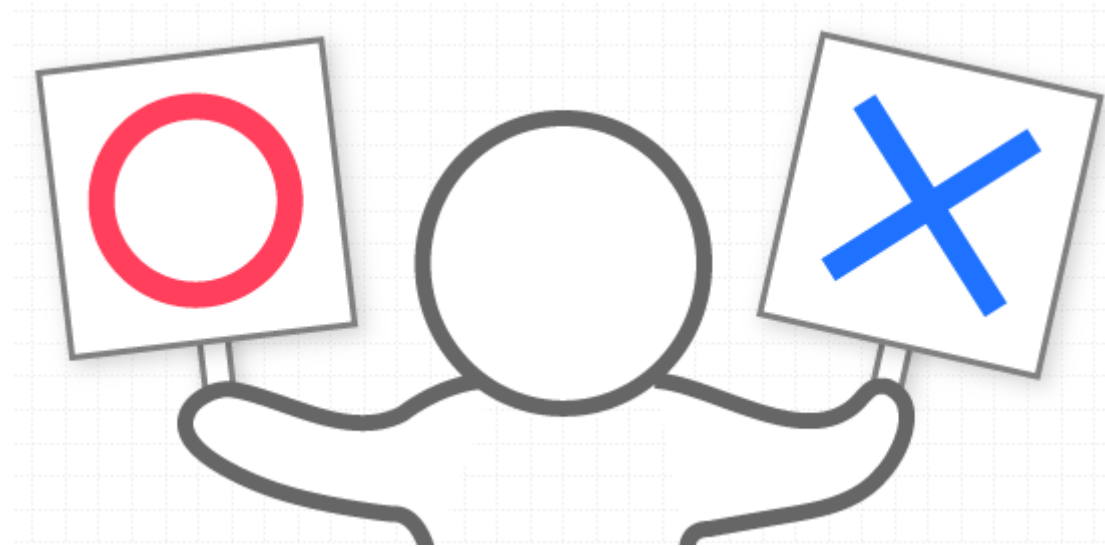




중급 웹 프로그래밍

가천대학교 SW중심대학

- 배열은 자료형이 같은 변수들의 모임이다.



정답은? ○

목차

1. MySQL 시작과 종료
2. 데이터 정의어
3. 데이터 조작어

1. MySQL 시작과 종료

데이터베이스 개념을 이해한다.

MySQL 접속 및 종료 방법을 익힌다.

■ 2차원 테이블에 데이터 저장

■ 관계형 데이터베이스의 구성 요소

• 테이블

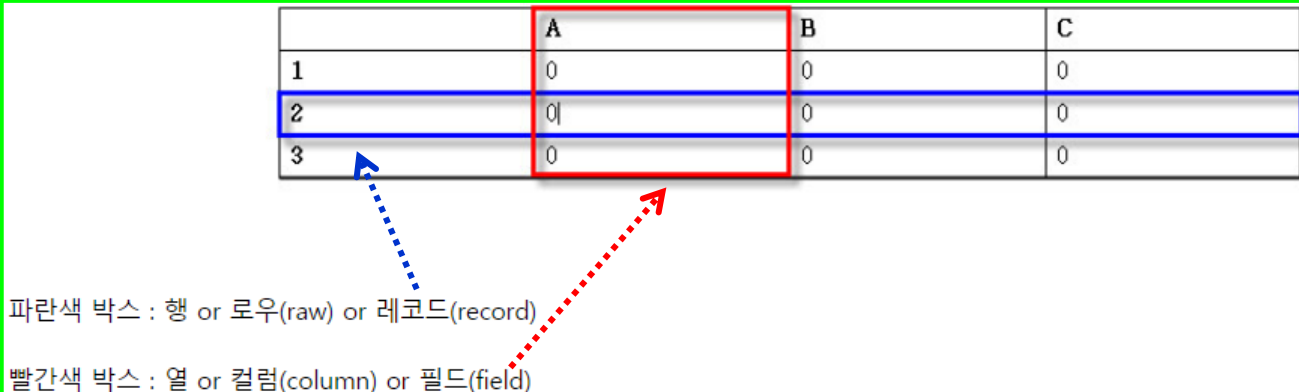
- 데이터를 저장하는 기본 단위
- ex) 웹 사이트에서 자유게시판, 질의응답, 회원 정보 등의 데이터가 각각 1개의 테이블에 저장됨

• 필드(속성)

- 데이터베이스 테이블의 항목 하나하나
- ex) 회원 정보 테이블의 아이디, 이름, 가입일, 주소, 전화번호 같은 항목

• 레코드

- 필드에 저장되는 한 세트의 정보
- ex) 회원 정보 테이블에서 한 명 분의 데이터 집합

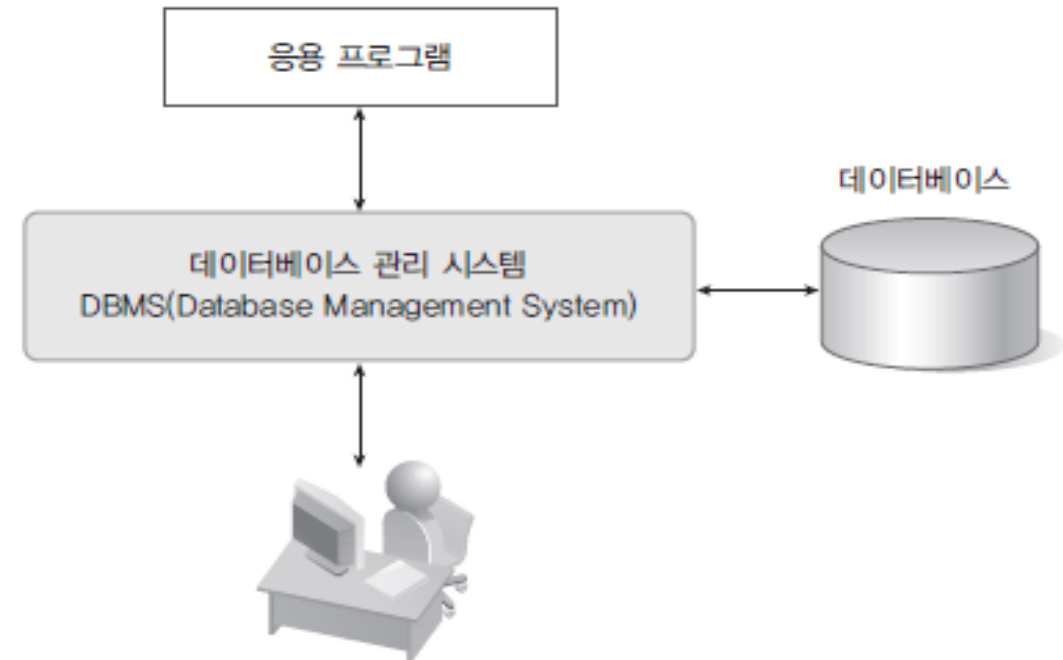


	A	B	C
1	0	0	0
2	0	0	0
3	0	0	0

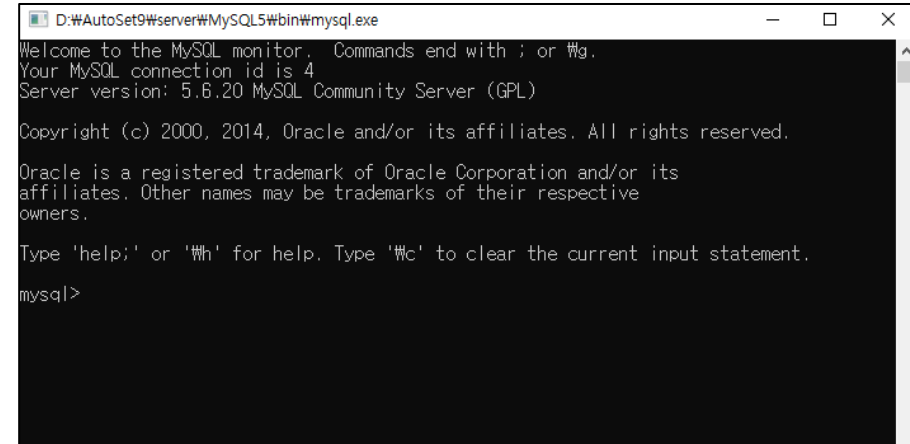
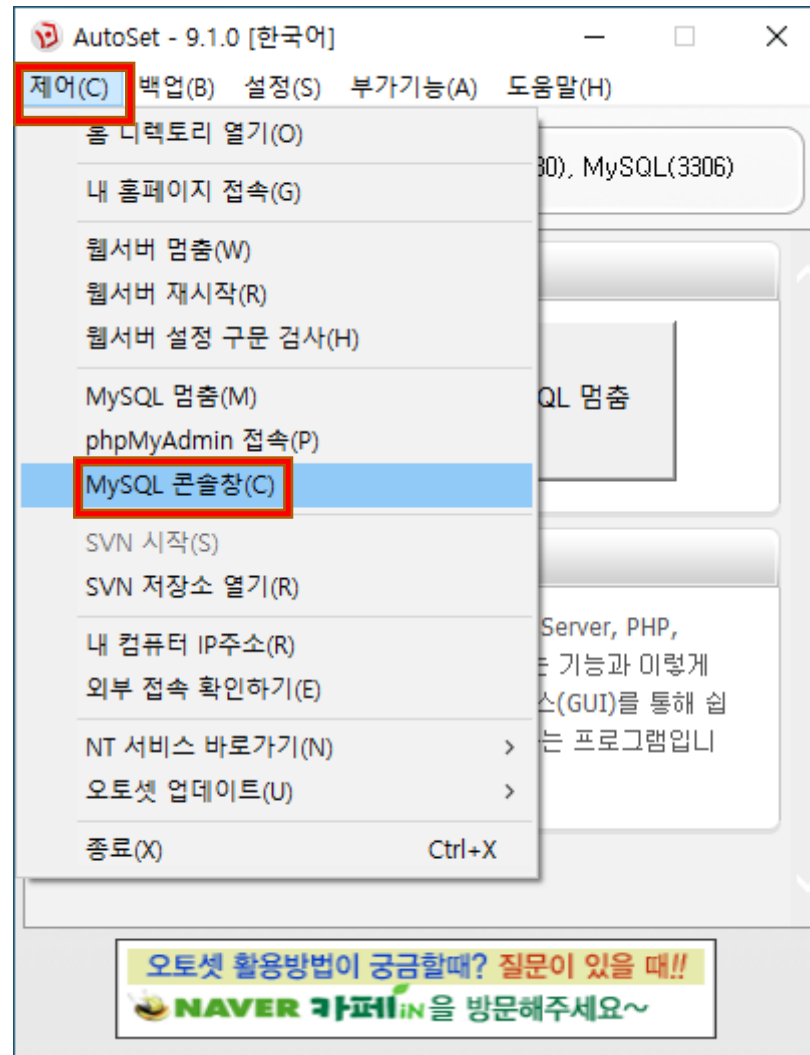
파란색 박스 : 행 or 로우(row) or 레코드(record)

빨간색 박스 : 열 or 컬럼(column) or 필드(field)

- SQL에 기반을 둔 관계형 DBMS 중 하나
- Oracle, IBM, Infomix 등의 데이터베이스는 고가이지만, MySQL 데이터베이스는 교육용은 무료
- 리눅스, 유닉스, 윈도우 등 거의 모든 운영체제에서 사용가능
- 처리 속도가 상당히 빠르고 대용량에 데이터도 처리 용이
- 설치 방법이 쉽고 초보자도 익히기 쉬움
- 보안성이 우수



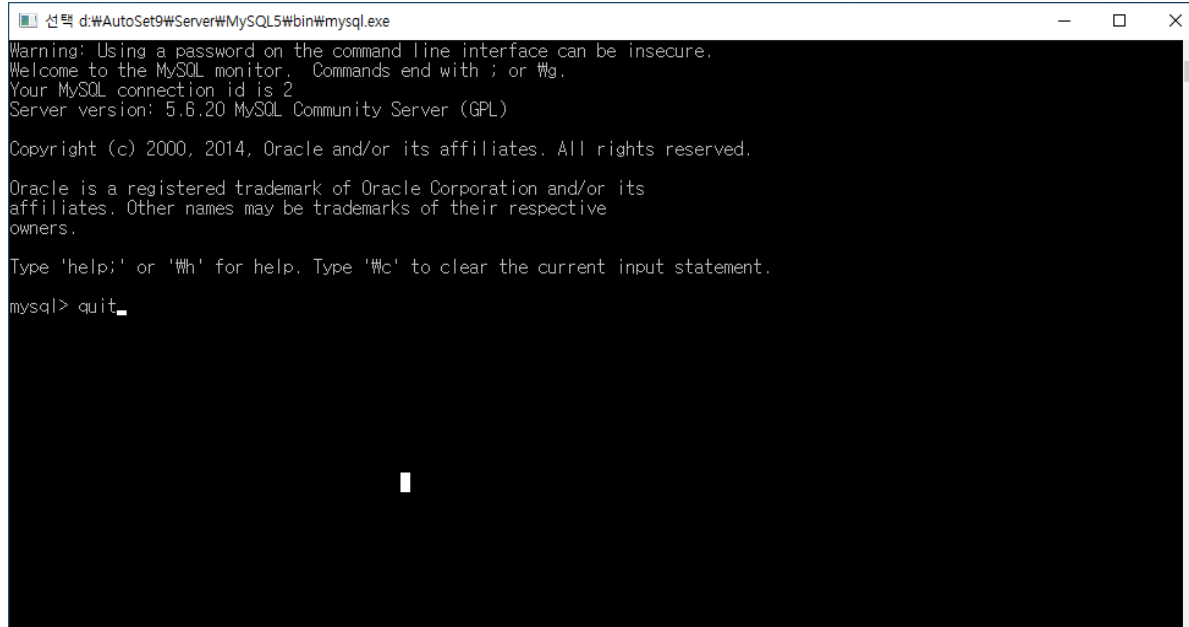
MySQL 시작



■ MySQL 종료

- mysql> quit
- mysql> **W**q

▶ 키보드 엔터키 바로 위에 있어요 ^^



```
선택 d:\AutoSet9\Server\MySQL5\bin\mysql.exe
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.20 MySQL Community Server (GPL)

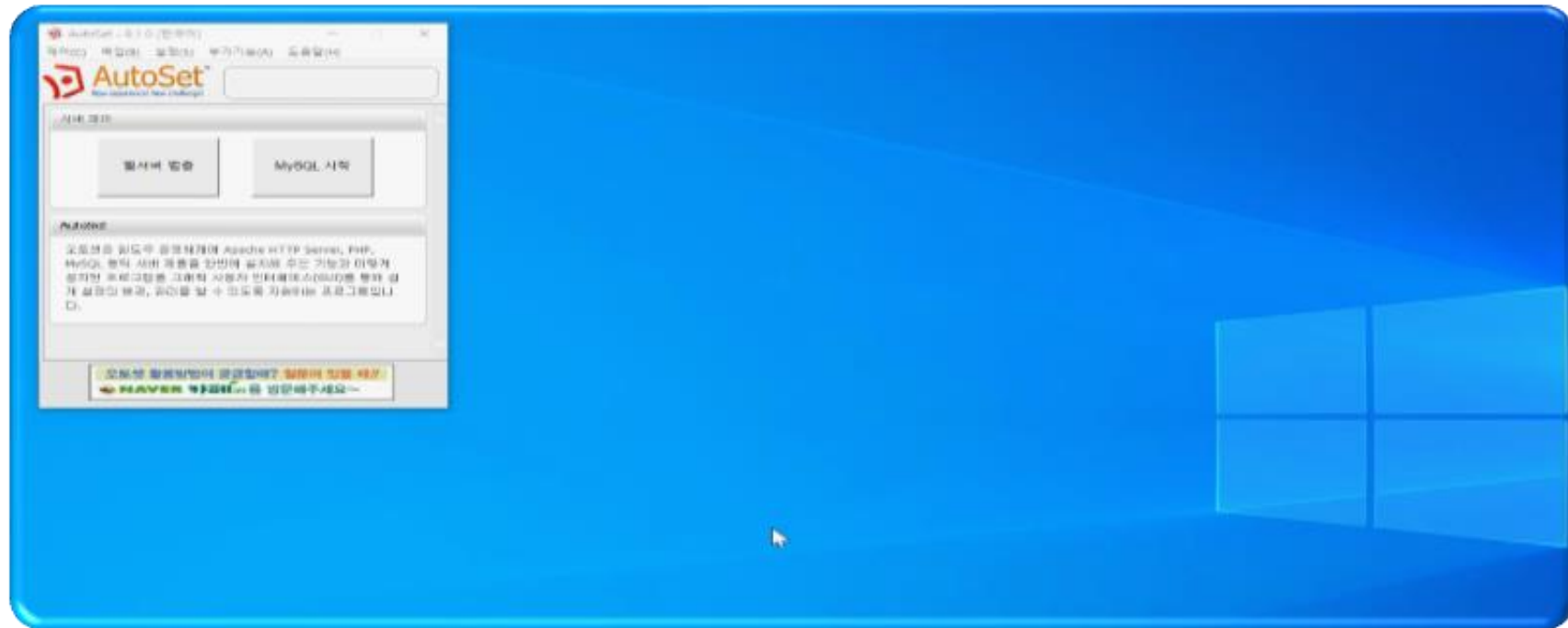
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

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> quit_
```


MySQL 시작과 종료



2. 데이터 정의어

■ 데이터베이스 생성 명령

데이터베이스 생성 명령

```
create database 데이터베이스명;
```

- `mysql> create database sample default CHARACTER SET UTF8;`
- 데이터베이스를 생성하고 한글을 사용할 수 있는 UTF8로 문자열 저장

■ 데이터베이스 목록 보기

데이터베이스 목록 출력 명령

```
show databases;
```

- `mysql> show databases;`

■ 데이터베이스 삭제

데이터베이스 삭제 명령

```
drop database 데이터베이스명;
```

- `mysql> drop database sample;`

■ 테이블 생성

```
CREATE TABLE 테이블_이름 (  
    필드_이름1 데이터_타입 [NOT NULL] [DEFAULT 기본값],  
    필드_이름2 데이터_타입 [NOT NULL] [DEFAULT 기본값],  
    ...  
    필드_이름n 데이터_타입 [NOT NULL] [DEFAULT 기본값],  
    [PRIMARY KEY (필드_이름)]  
);
```

- []의 내용은 생략이 가능
- SQL 질의문은 세미콜론(;)으로 문장의 끝을 표시
- SQL 질의문은 대소문자를 구분하지 않음
- 문자열이나 날짜 데이터는 작은 따옴표로 묶어서 표현(작은 따옴표로 묶여진 문자열은 대소문자를 구분함)

■ 필드(속성)의 정의

- 테이블을 구성하는 각 필드의 데이터 타입을 선택한 다음 NULL 값 허용 여부와 DEFAULT 값 필요 여부를 결정
 - **NOT NULL** : 필드가 널 값을 허용하지 않음을 의미하는 키워드
 - 예) 고객아이디 VARCHAR(20) **NOT NULL**
 - **DEFAULT** : 필드의 기본 값을 지정하는 키워드
 - 예) 적립금 INT **DEFAULT 0**

■ 키의 정의

- **PRIMARY KEY** : 기본키를 지정하는 키워드
 - 예) **PRIMARY KEY**(고객아이디)

■ 필드의 대표적인 데이터 타입

데이터 타입	예제
숫자형	INT , FLOAT, ...
문자형	CHAR , VARCHAR , TEXT, ...
날짜형	DATE, DATETIME, TIME, ...

- **INT** 타입은 앞에 0을 붙여 저장해도 0이 없어지고, 첫 자리는 0이 아닌 숫자가 나올 때까지 0이 없어짐
- **CHAR(n)** 타입은 무조건 n 길이 만큼을 차지하고, 출력시에도 n 길이로 체크됨
- **VARCHAR(n)**은 실제로 들어있는 문자길이만큼 인식됨

- 예) sample 데이터베이스 테이블 설계(테이블명: **friend**)

필드명	타입	설명	속성
id	int	일련번호	AUTO_INCREMENT, PRIMARY KEY
name	varchar(10)	이름	NOT NULL
address	varchar(80)	주소	
tel	varchar(20)	전화번호	DEFAULT '010-0000-0000'
email	varchar(20)	이메일 주소	

■ 예) sample 데이터베이스 테이블 설계(테이블명: *friend*)

- mysql> **create** table *friend* (
 - > **id** int auto_increment not null,
 - > **name** varchar(10) not null,
 - > **address** varchar(80),
 - > **tel** varchar(20) default '010-0000-0000',
 - > **email** varchar(20),
 - > **primary key**(id)
 - >);

```
MySQL Console(root)
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.1.41-community MySQL Community Server (GPL)

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

mysql> use sample;
Database changed
mysql> create table friend (
    -> id int auto_increment not null,
    -> name varchar(10) not null,
    -> address varchar(80),
    -> tel varchar(20) default '010-0000-0000',
    -> email varchar(20),
    -> primary key(id)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> _
```

■ 데이터베이스 테이블의 구조 확인

데이터베이스 테이블의 필드 구조 출력 명령

desc 테이블명;

■ 예) sample 데이터베이스 테이블 설계(테이블명: friend)

- `mysql> desc friend;`

```
MySQL Console(root)
mysql> create table friend (
  -> id int auto_increment not null,
  -> name varchar(10) not null,
  -> address varchar(80),
  -> tel varchar(20) default '010-0000-0000',
  -> email varchar(20),
  -> primary key(id)
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql> desc friend;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(10)	NO		NULL	
address	varchar(80)	YES		NULL	
tel	varchar(20)	YES		010-0000-0000	
email	varchar(20)	YES		NULL	

```
5 rows in set (0.03 sec)

mysql>
```

데이터베이스 테이블에 새로운 필드 추가

■ 새로운 필드 추가 명령

데이터베이스 테이블의 필드 추가 명령

```
alter table 테이블명 add 새로운 필드명 필드타입 [first 또는 after 필드명];
```

■ 예) sample 데이터베이스의 friend 테이블에 나이 필드를 정수형으로 추가

1. `mysql> alter table friend add age int;`
2. `mysql> desc friend;`

```
MySQL Console(root)
+-----+-----+-----+-----+-----+-----+
| tel   | varchar(20) | YES | | 010-0000-0000 |
| email | varchar(20) | YES | | NULL          |
+-----+-----+-----+-----+-----+
5 rows in set (0.03 sec)

mysql> alter table friend add age int;
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc friend;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)   | NO   | PRI | NULL    | auto_increment |
| name  | varchar(10) | NO   |     | NULL    |
| address | varchar(80) | YES  |     | NULL    |
| tel   | varchar(20) | YES  |     | 010-0000-0000 |
| email | varchar(20) | YES  |     | NULL    |
| age   | int(11)   | YES  |     | NULL    |
+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

mysql> _
```

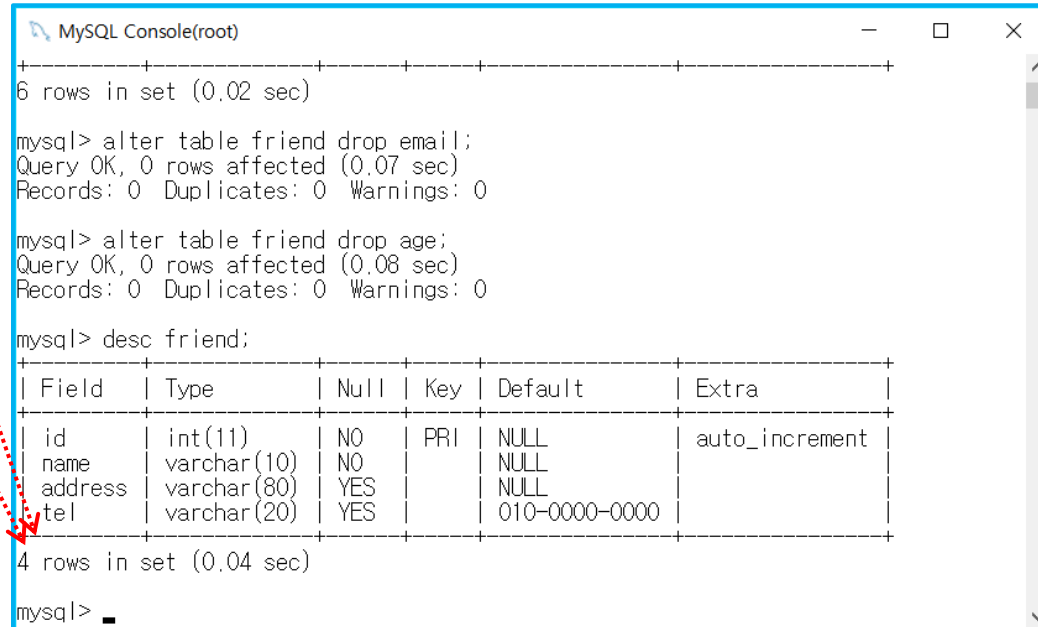
■ 테이블에서 필드 삭제 명령

데이터베이스 테이블의 특정 필드 삭제 명령

`alter table 테이블명 drop 삭제할 필드명1, 삭제할 필드명2;`

■ 예) sample 데이터베이스의 friend 테이블에서 email과 나이 필드 삭제하기

1. `mysql> alter table friend drop email;`
2. `mysql> alter table friend drop age;`
3. `mysql> desc friend;`



```
MySQL Console(root)
6 rows in set (0.02 sec)

mysql> alter table friend drop email;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table friend drop age;
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc friend;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default        | Extra          |
+-----+-----+-----+-----+-----+-----+
| id     | int(11)       | NO   | PRI | NULL           | auto_increment |
| name   | varchar(10)   | NO   |     | NULL           |                |
| address | varchar(80)   | YES  |     | NULL           |                |
| tel    | varchar(20)   | YES  |     | 010-0000-0000  |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> _
```

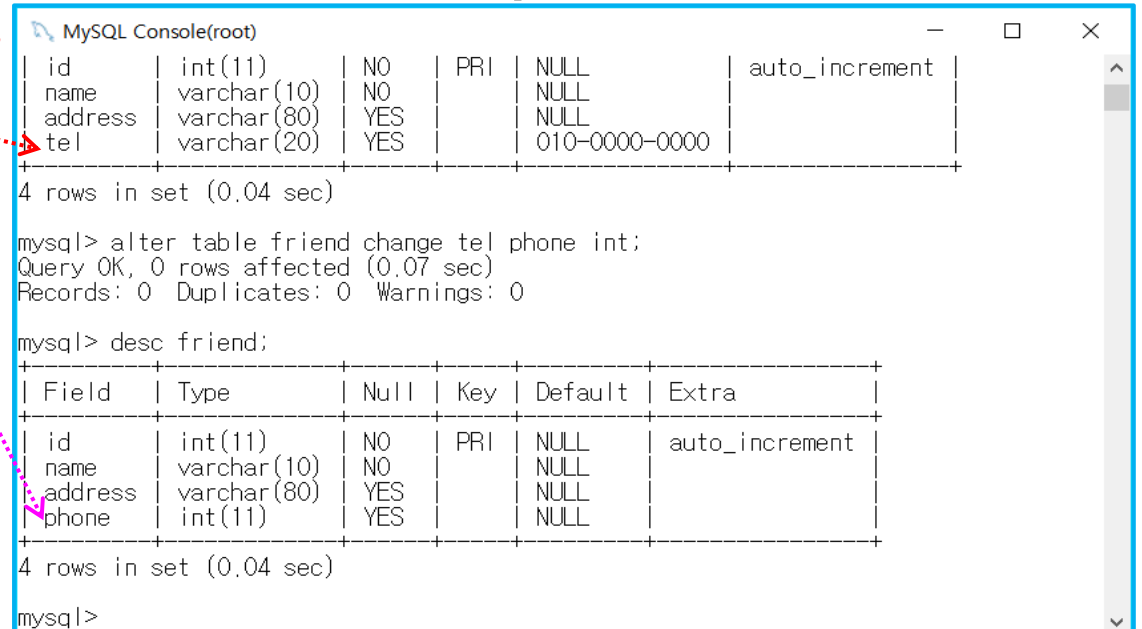
■ 테이블에서 필드 수정 명령

데이터베이스 테이블의 필드 수정 명령

`alter table 테이블명 change 이전 필드명 새로운 필드명 필드 타입;`

■ 예) sample 데이터베이스의 friend 테이블의 필드 중 tel varchar(20)을 phone int로 변경

1. `mysql> alter table friend change tel phone int;`
2. `mysql> desc friend;`



The screenshot shows a MySQL console window with the following content:

```
MySQL Console(root)
+----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| name | varchar(10) | NO | | NULL | |
| address | varchar(80) | YES | | NULL | |
| tel | varchar(20) | YES | | 010-0000-0000 | |
+----+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> alter table friend change tel phone int;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc friend;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| name | varchar(10) | NO | | NULL | |
| address | varchar(80) | YES | | NULL | |
| phone | int(11) | YES | | NULL | |
+----+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql>
```

Two red dotted arrows point from the text in the list to the console output: one from 'change' to the 'change' command, and another from 'tel' to the 'tel' field in the initial table structure.

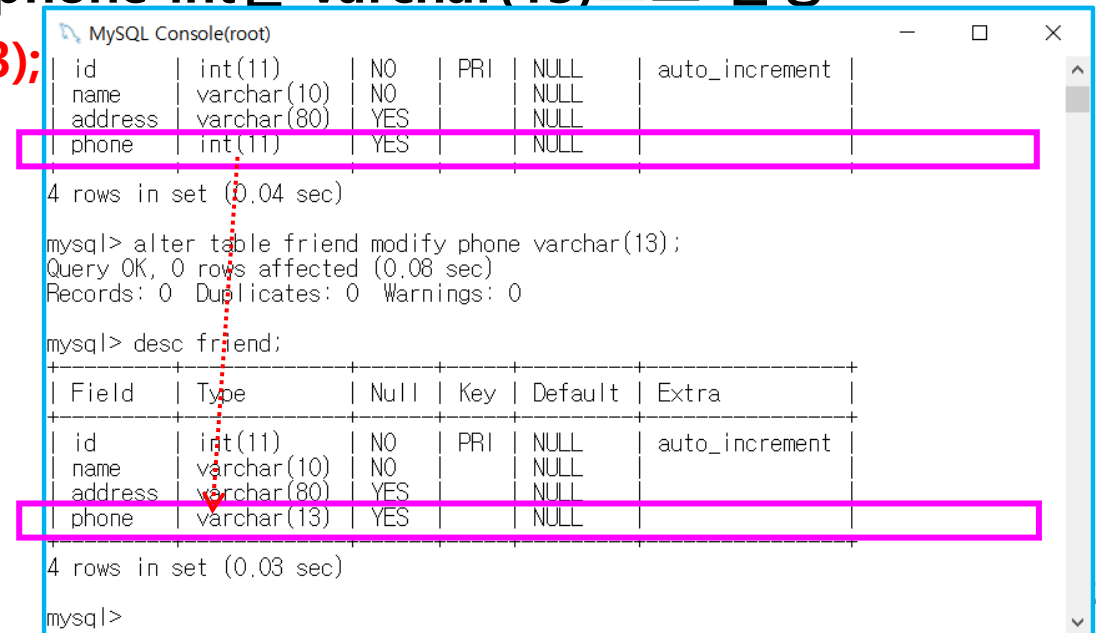
■ 테이블에서 필드 타입 수정 명령

데이터베이스 테이블 필드 타입 수정 명령

`alter table 테이블명 modify 필드명 새로운 타입;`

■ 예) sample 데이터베이스의 friend 테이블의 필드 중 phone int를 varchar(13)으로 변경

1. `mysql> alter table friend modify phone varchar(13);`
2. `mysql> desc friend;`



```
MySQL Console(root)
+----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| name | varchar(10) | NO | | NULL | |
| address | varchar(80) | YES | | NULL | |
| phone | int(11) | YES | | NULL | |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> alter table friend modify phone varchar(13);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc friend;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| name | varchar(10) | NO | | NULL | |
| address | varchar(80) | YES | | NULL | |
| phone | varchar(13) | YES | | NULL | |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)

mysql>
```

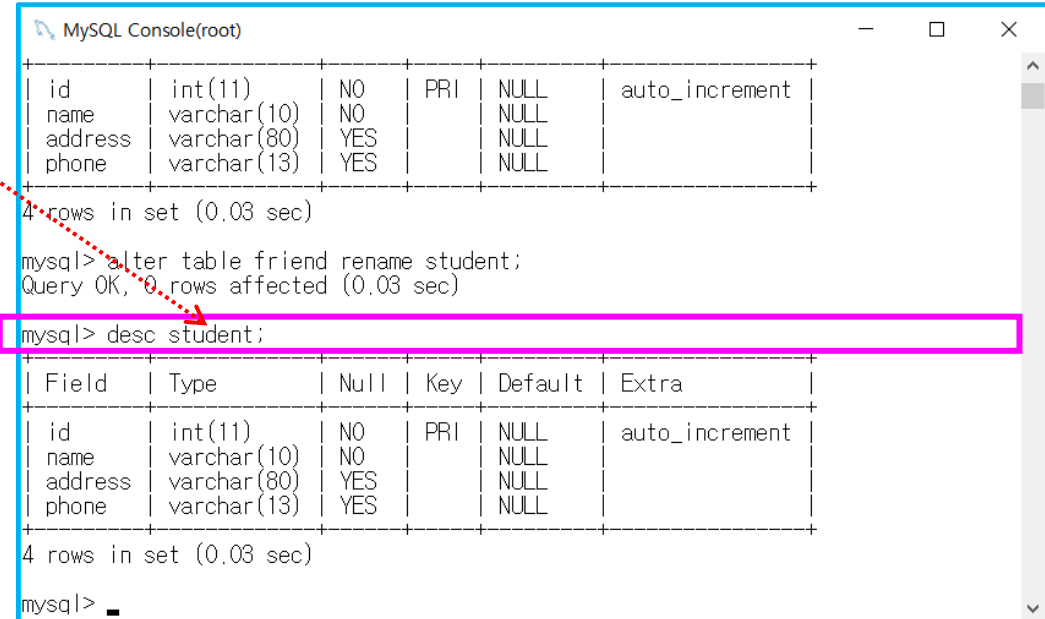
■ 테이블 이름 수정 명령

데이터베이스 테이블명 수정 명령

alter table 이전 테이블명 **rename** 새로운 테이블명;

■ 예) sample 데이터베이스의 friend 테이블 이름을 student로 변경

1. `mysql> alter table friend rename student;`
2. `mysql> desc student;`



```
MySQL Console(root)
+----+-----+-----+-----+-----+-----+
| id | name | int(11) | NO | PRI | NULL | auto_increment |
| name | varchar(10) | NO |  | NULL |  |  |
| address | varchar(80) | YES |  | NULL |  |  |
| phone | varchar(13) | YES |  | NULL |  |  |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)

mysql> alter table friend rename student;
Query OK, 0 rows affected (0.03 sec)

mysql> desc student;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| name | varchar(10) | NO |  | NULL |  |
| address | varchar(80) | YES |  | NULL |  |
| phone | varchar(13) | YES |  | NULL |  |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)

mysql> _
```

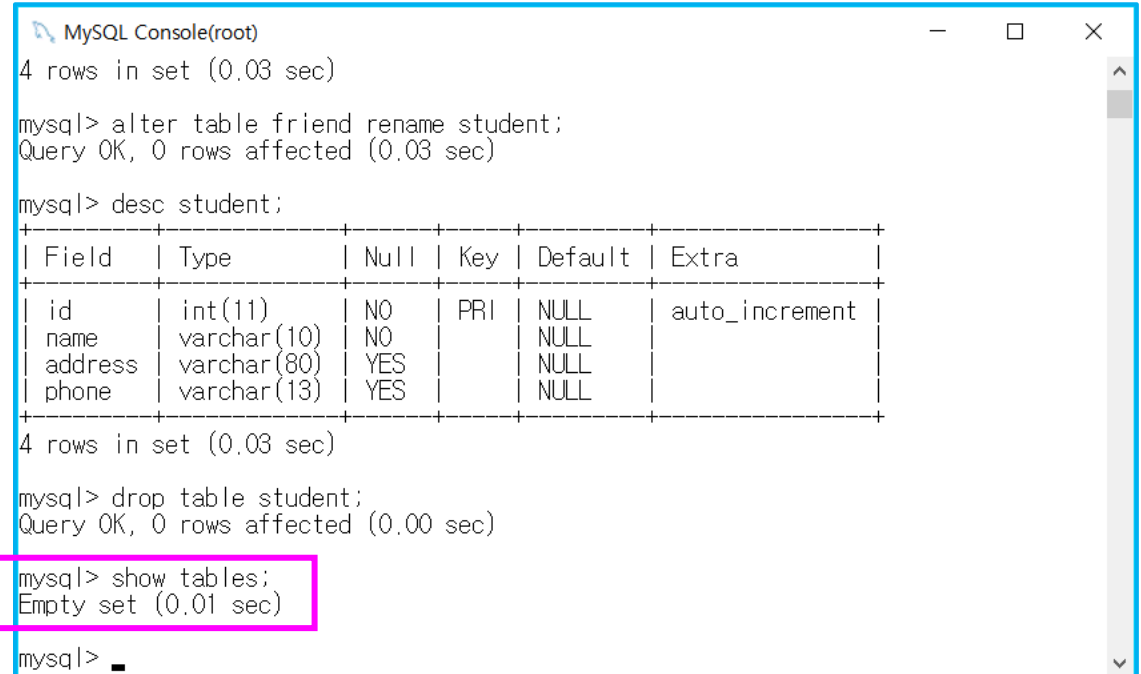
■ 테이블 삭제 명령

데이터베이스 테이블 삭제 명령

drop table 테이블명;

■ 예) sample 데이터베이스의 student 삭제

1. `mysql> drop table student;`
2. `mysql> show tables;`



```
MySQL Console(root)
4 rows in set (0.03 sec)

mysql> alter table friend rename student;
Query OK, 0 rows affected (0.03 sec)

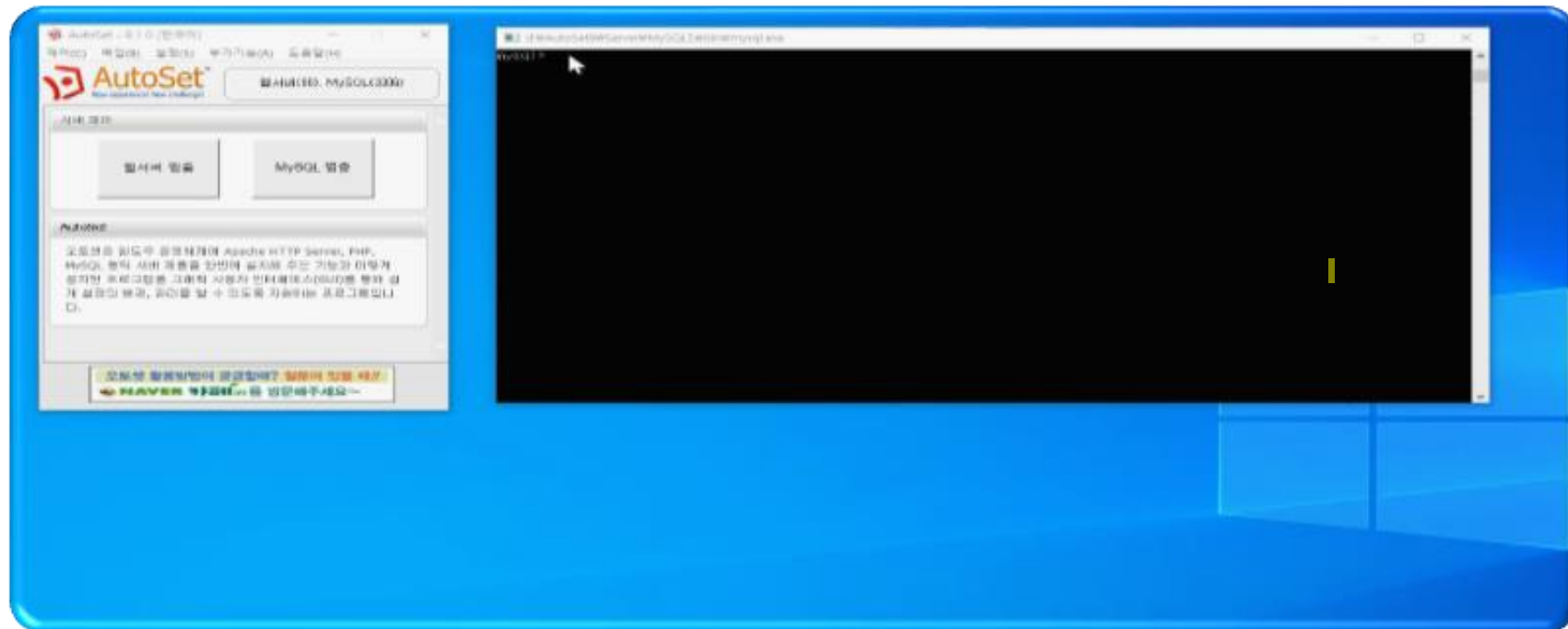
mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id     | int(11)       | NO   | PRI | NULL    | auto_increment |
| name   | varchar(10)   | NO   |     | NULL    |                |
| address| varchar(80)   | YES  |     | NULL    |                |
| phone  | varchar(13)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)

mysql> drop table student;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
Empty set (0.01 sec)

mysql>
```


데이터 정의어



3. 데이터 조작어

■ 테이블에 데이터 삽입

```
INSERT  
INTO 테이블_이름 (필드_이름1, 필드_이름2, ..., 필드_이름n)  
VALUES (필드_이름1_값, 필드_이름2_값, ..., 필드_이름n_값);
```

- INTO 키워드와 함께 레코드를 삽입할 테이블의 이름과 필드의 이름을 나열
- VALUES 키워드와 함께 삽입할 필드 값들을 나열
- INTO 절의 필드 이름과 VALUES 절의 필드 값은 순서대로 일대일 대응되어야 함

▶ 강의동영상 화면을 멈추고
한번씩 실습에 보세요 ^^

■ 테이블에 데이터 삽입 명령

- `mysql> insert into member(id, name, gender, post_num, address, tel, age)`
`values('yjhwan', '황영주', 'm', '10011', '서울시 중구 충무로1가', '010-2334-8879', 22);`
- `mysql> insert into member(id, name, gender, post_num, address, tel, age)`
`values('khshul', '설기형', 'm', '60701', '부산시 동래구 명륜동', '010-7664-3784', 20);`

■ 테이블 member에 저장된 레코드 확인

▶ 강의동영상 화면을 멈추고
한번씩 실습에 보세요 ^^

1. `mysql> select * from member;`

```
mysql> select * from member;
```

num	id	name	gender	post_num	address	tel	age
1	yjhwan	황영주	m	10011	서울시 중구 충무로1가	010-2334-8879	22
2	khshul	설기형	m	60701	부산시 동래구 명륜동	010-7664-3784	20
3	chpark	박철호	m	50320	광주시 남구 지석동	010-2981-9730	25
4	cutecat	김나래	f	17024	경기도 성남시 수정구 북정동	010-9224-0119	23
5	preety	한혜린	f	10021	서울시 동작구 흑석동	010-5214-8787	24
6	mylee	이명연	f	50351	광주시 서구 쌍촌동	010-9837-9432	23
7	kskim	김길수	m	15031	경기도 수원시 장안구 파장동	010-9399-9807	20

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

- 테이블에서 전체 레코드 검색 명령

특정 필드에 입력된 데이터 검색 명령

```
select * from 테이블명;
```

- 예) member_db 데이터베이스의 member 테이블에 저장된 모든 회원의 레코드 검색

1. `mysql> select * from member;`

■ 테이블에서 특정 필드 검색 명령

특정 필드에 입력된 데이터 검색 명령

```
select 필드명1, 필드명2 from 테이블명;
```

■ 예) member_db 데이터베이스의 member 테이블에 저장된 회원의 아이디, 이름, 주소 필드만 검색

1. `mysql> select id, name, address from member;`

▶ 강의동영상 화면을 멈추고
한번씩 실습해 보세요^^

```
mysql> select id, name, address from member;
```

id	name	address
yjhwang	황영주	서울시 중구
khshul	설기철	부산시 남구
chpark	박철호	광주시 동구
cutecat	김나래	경기도 성남시
preety	한혜린	서울시 동작구
mylee	이명연	광주시 서구
kskim	김길수	경기도 수원시

7 rows in set (0.00 sec)

■ 테이블에서 조건에 맞는 레코드 검색

```
SELECT 필드_이름1, 필드_이름2,..., 필드_이름n  
FROM 테이블_이름  
[WHERE 조건];
```

- WHERE 키워드와 함께 비교 연산자와 논리 연산자를 이용한 검색 조건 제시
- 비교 연산자: =, <>, <, >, <=, >=
- 논리 연산자: AND, OR, NOT

테이블에서 조건에 맞는 레코드 검색

- 예) member 테이블에서 여성인 회원의 아이디, 이름, 주소, 전화번호, 성별 필드 검색

- `mysql> select id, name, address, tel, gender from member where gender='f';`

```
mysql> select id, name, address, tel, gender from member where gender = 'f';
```

id	name	address	tel	gender
cutecat	김나래	경기도 성남시 수정구 복정동	010-9224-0119	f
preety	한혜린	서울시 동작구 흑석동	010-5214-8787	f
mylee	이명연	광주시 서구 쌍촌동	010-9837-9432	f

3 rows in set (0.00 sec)

▶ 강의동영상 화면을 멈추고
한번씩 실습에 보세요 ^^

- 예) member 테이블에서 나이가 20~23세인 회원의 이름, 주소, 나이 필드 검색

- `mysql> select name, address, age from member where age>=20 and age<=23;`

```
mysql> select name, address, age from member where age>=20 and age <=23;
```

name	address	age
황영주	서울시 중구 충무로1가	22
설기형	부산시 동래구 명륜동	20
김나래	경기도 성남시 수정구 복정동	23
이명연	광주시 서구 쌍촌동	23
김길수	경기도 수원시 장안구 파장동	20

5 rows in set (0.00 sec)

▶ 강의동영상 화면을 멈추고
한번씩 실습에 보세요 ^^

■ 테이블에서 LIKE를 이용한 "특정 문자열"이 포함된 레코드 검색

- LIKE 키워드를 이용해 부분적으로 일치하는 데이터를 검색
- 문자열을 이용하는 조건에만 LIKE 키워드 사용 가능
- % : 문자의 내용과 개수는 상관 없음
- _ : 한 개의 문자로 문자의 내용은 상관 없음

LIKE '데이터%'	데이터로 시작하는 문자열 (데이터로 시작하기만 하면 길이는 상관 없음)
LIKE '%데이터'	데이터로 끝나는 문자열 (데이터로 끝나기만 하면 길이는 상관 없음)
LIKE '%데이터%'	데이터가 포함된 문자열
LIKE '데이터 _ _ _'	데이터로 시작하는 6자 길이의 문자열
LIKE '_ _ 한%'	세 번째 글자가 '한'인 문자열

테이블에서 특정 문자열이 포함된 레코드 검색

- 테이블에서 LIKE를 이용한 특정 문자열이 포함된 레코드 검색 명령

특정 문자열이 포함된 레코드 검색 명령

```
select 필드명1, 필드명2, from 테이블명 where 검색 필드 like 조건식;
```

- 예) member 테이블에서 성이 김씨인 회원의 이름, 주소, 전화번호 필드 검색

- `mysql> select name, address, tel from member where name LIKE '김%';`

```
mysql> select name, address, tel from member where name like '김%';
```

name	address	tel
김나래	경기도 성남시 수정구 복정동	010-9224-0119
김길수	경기도 수원시 장안구 파장동	010-9399-9807

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

▶ 강의동영상 화면을 멈추고
이번씩 실습에 보세요 ^^

■ 테이블에서 레코드의 정렬 검색

```
SELECT 필드_이름1, 필드_이름2, ... , 필드_이름n  
FROM 테이블_이름  
[WHERE 조건]  
[ORDER BY 속성_리스트 [ASC | DESC] ];
```

- ORDER BY 키워드를 이용해 결과 테이블 내용을 사용자가 원하는 순서로 출력
- ORDER BY 키워드와 함께 정렬 기준이 되는 속성과 정렬 방식을 지정
- 오름차순(기본): **ASC** / 내림차순 : **DESC**

- 예) member 테이블에서 나이순으로 아이디, 이름, 성별, 나이 레코드를 **내림차순**으로 검색

- mysql> select id, name, gender, age from member order by age **DESC**;

```
mysql> select id, name, gender, age from member order by age desc;
```

id	name	gender	age
chpark	박철호	m	25
preety	한혜린	f	24
cutecat	김나래	f	23
mylee	이명연	f	23
yjhwang	황영주	m	22
khshul	설기형	m	20
kskim	김길수	m	20

7 rows in set (0.02 sec)

▶ 강의동영상 화면을 멈추고
한번씩 실습에 보세요^^

- 예) member 테이블에서 광주에 사는 회원의 이름, 주소, 전화번호를 이름으로 **오름차순** 검색

- mysql> select name, address, tel from member where address like '광주%' order by name **ASC**;

```
mysql> select name, address, tel from member where address like '광주%' order by name asc;
```

name	address	tel
이명연	광주시 서구 쌍촌동	010-9837-9432
박철호	광주시 남구 지식동	010-2981-9730

2 rows in set (0.00 sec)

▶ 강의동영상 화면을 멈추고
한번씩 실습에 보세요^^

■ 테이블에 저장된 레코드의 특정 필드 값 수정

```
UPDATE 테이블_이름  
SET 필드_이름1 = 값1, 필드_이름2 = 값2,..., 필드_이름n = 값n)  
[ WHERE 조건 ];
```

- SET 키워드 다음에 필드 값을 어떻게 수정할 것인지를 지정
- WHERE 절에 제시된 조건을 만족하는 레코드에 대해서만 필드 값을 수정
- WHERE 절을 생략하면 테이블에 존재하는 모든 레코드를 대상으로 수정

■ 예) member 테이블에서 한혜린 회원의 아이디를 pretty로 수정

1. `mysql> update member set id='pretty' where name= '한혜린';`
2. `mysql> select id, name from member where name= '한혜린';`

```
mysql> update member set id='pretty' where name = '한혜린';  
Query OK, 1 row affected (0.02 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select id, name from member where name = '한혜린';  
+-----+-----+  
| id      | name    |  
+-----+-----+  
| pretty  | 한혜린  |  
+-----+-----+  
1 row in set (0.00 sec)
```

▶ 강의동영상 화면을 멈추고
한번씩 실습해 보세요^^

■ 테이블에 저장된 레코드 삭제

```
DELETE  
FROM 테이블_이름  
[ WHERE 조건 ];
```

- WHERE 절에 제시된 조건을 만족하는 레코드에 대해서만 삭제
- WHERE 절을 생략하면 테이블에 존재하는 모든 레코드를 삭제

Q&A