

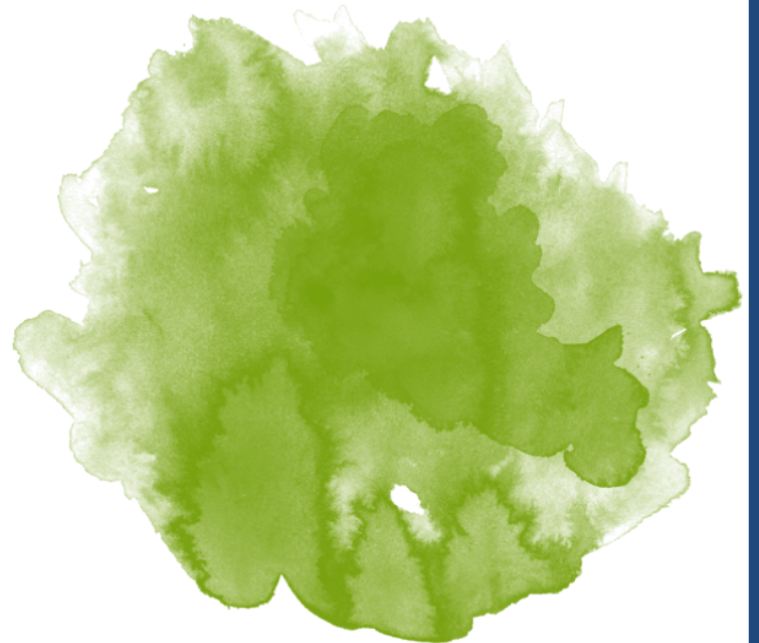


# 06장 MySQL 명령

---

# 목차

1. 데이터베이스 관련 명령
2. 데이터베이스 테이블 관련 명령
3. SQL 명령의 일괄 실행
4. 레코드 관련 명령



# 1. 데이터베이스 관련 명령

---

- 데이터베이스 접속 명령
  - 시작메뉴- APMSetup 7 - MySQL Console(root)
  - Root 계정 password : apmsetup

- 데이터베이스 생성 명령

데이터베이스 생성 명령

```
create database 데이터베이스명;
```

- mysql> **create database** sample1;

# 1. 데이터베이스 관련 명령

---

- 데이터베이스 목록 보기

데이터베이스 목록 출력 명령

```
show databases;
```

- `mysql> show databases;`

- 데이터베이스 삭제

데이터베이스 삭제 명령

```
drop database 데이터베이스명;
```

- `mysql> drop database sample1;`

## 2. 데이터베이스 테이블 관련 명령

- 데이터베이스 테이블 설계

[표 6-1] 주소록 데이터베이스 테이블 설계(테이블명: friend)

필드명	타입	설명
num	int	일련번호
name	char(10)	이름
address	varchar(80) 또는 char(80)	주소
tel	char(20)	전화번호
email	char(20)	이메일 주소

## 2. 데이터베이스 테이블 관련 명령

- 테이블 생성 명령

### 데이터베이스 테이블 생성 명령

```
mysql> create table 테이블명(  
    필드명1 타입,  
    필드명2 타입,  
    필드명3 타입,  
    .....  
    PRIMARY KEY(필드명)  
);
```

## 2. 데이터베이스 테이블 관련 명령

- 주소록 테이블(friend) 만들기 (표 6-1)
  - kdhong\_db 데이터베이스 이용
  - mysql> create table friend (
    - > num int not null,
    - > name char(10),
    - > address char(80),
    - > tel char(20),
    - > email char(20),
    - > primary key(num)
    - > );



```
C:\>mysql -ukdhong -p1234 kdhong_db
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
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> create table friend (
-> num int not null,
-> name char(10),
-> address char(80),
-> tel char(20),
-> email char(20),
-> primary key(num)
-> );
Query OK, 0 rows affected (0.06 sec)
```

[그림 6-9] friend  
테이블 생성

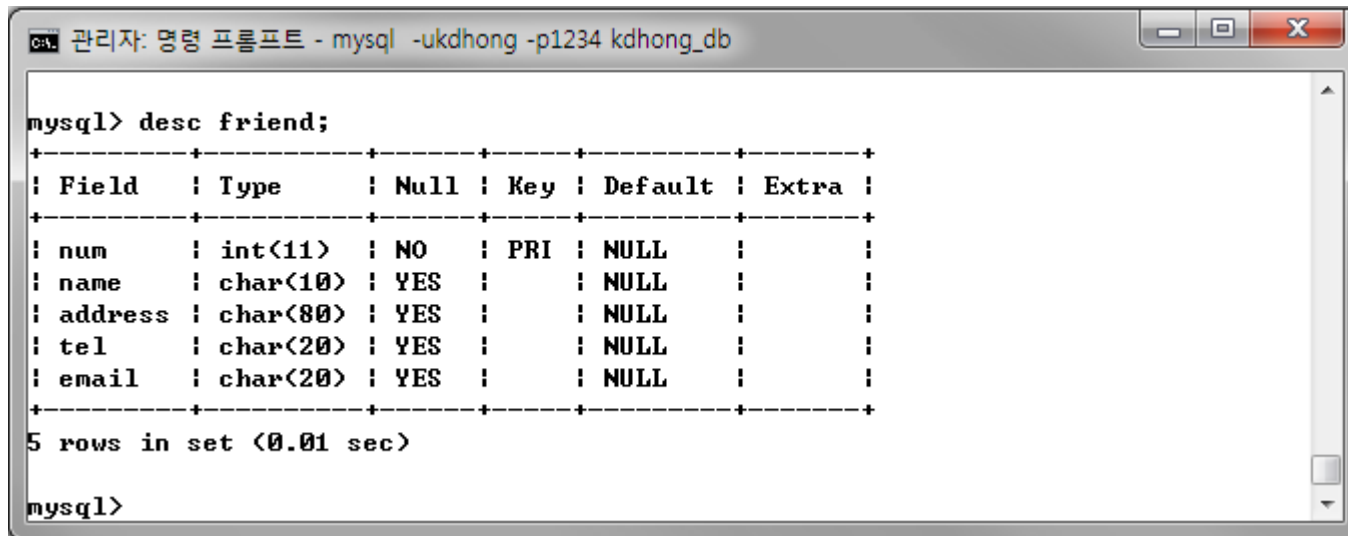
## 2. 데이터베이스 테이블 관련 명령

- 데이터베이스 테이블의 구조 확인

데이터베이스 테이블의 필드 구조 출력 명령

```
desc 테이블명;
```

- mysql> desc friend;



```
mysql> desc friend;
```

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	
name	char(10)	YES		NULL	
address	char(80)	YES		NULL	
tel	char(20)	YES		NULL	
email	char(20)	YES		NULL	

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

```
mysql>
```

[그림 6-11] friend 테이블의 구조 확인



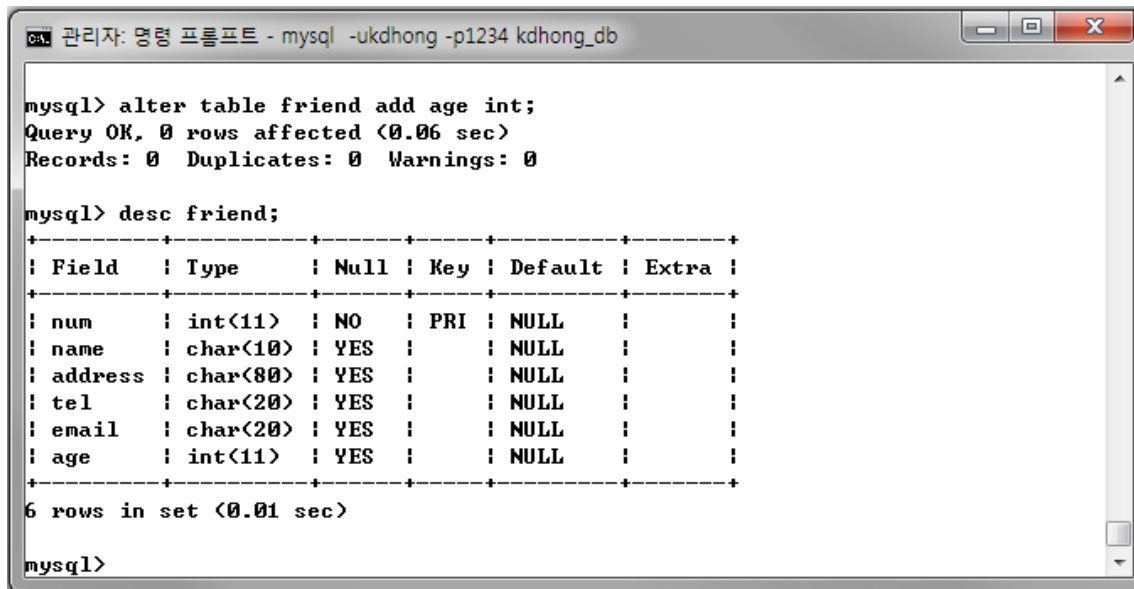
## 2. 데이터베이스 테이블 관련 명령

- 새로운 필드 추가 명령

데이터베이스 테이블의 필드 추가 명령

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

- ex) 앞서 만든 friend 테이블 나이 필드를 정수형으로 추가
  - mysql> alter table friend add age int;
  - mysql> desc friend;



```
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

mysql> alter table friend add age int;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc friend;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| num   | int(11) | NO   | PRI | NULL    |       |
| name  | char(10) | YES  |     | NULL    |       |
| address | char(80) | YES  |     | NULL    |       |
| tel   | char(20) | YES  |     | NULL    |       |
| email | char(20) | YES  |     | NULL    |       |
| age   | int(11) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

mysql>
```

[그림 6-12] friend  
테이블에 age 필드  
추가

## 2. 데이터베이스 테이블 관련 명령

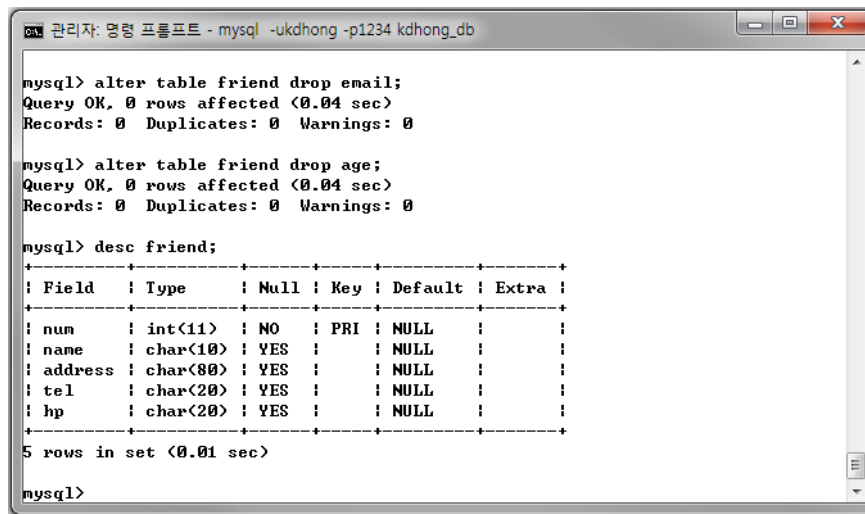
- 필드 삭제

데이터베이스 테이블의 특정 필드 삭제 명령

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

- ex) friend 테이블에서 email과 age 필드 삭제

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



```
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

mysql> alter table friend drop email;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table friend drop age;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc friend;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| num   | int(11) | NO   | PRI | NULL    |       |
| name  | char(10) | YES  |     | NULL    |       |
| address | char(80) | YES  |     | NULL    |       |
| tel   | char(20) | YES  |     | NULL    |       |
| hp    | char(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql>
```

[그림 6-14] friend 테이블에서 age와 email 필드 삭제

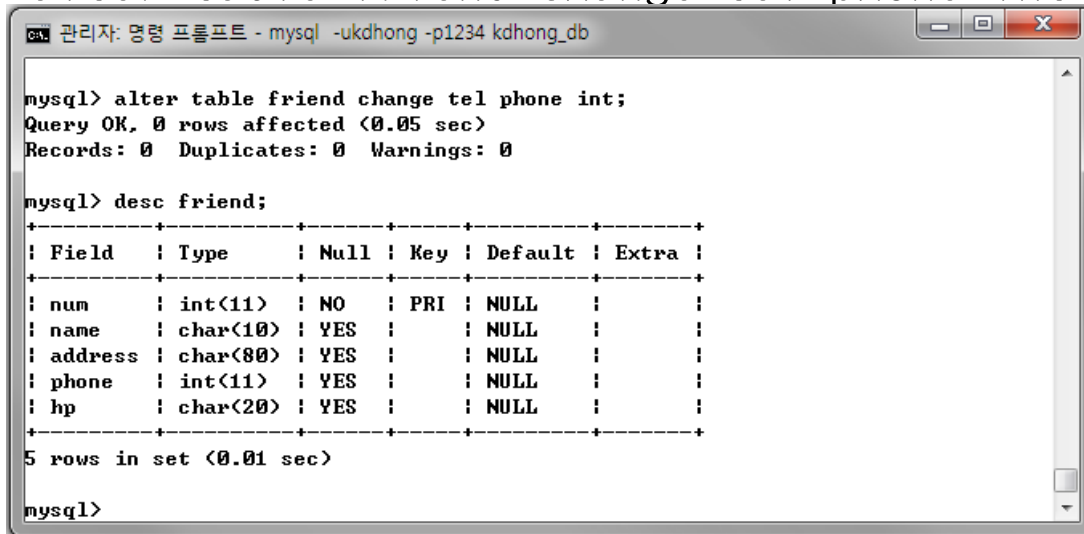
## 2. 데이터베이스 테이블 관련 명령

- 필드 수정 명령

데이터베이스 테이블의 필드 수정 명령

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

- ex) friend 테이블 필드 중 tel char(20)을 phone int로 변경
  - mysql> alter table friend change tel phone int;



```
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

mysql> alter table friend change tel phone int;
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc friend;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| num   | int(11)| NO   | PRI | NULL    |       |
| name  | char(10)| YES  |     | NULL    |       |
| address | char(80)| YES  |     | NULL    |       |
| phone | int(11)| YES  |     | NULL    |       |
| hp    | char(20)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql>
```

[그림 6-15] tel char(20)을 phone int로 변경

## 2. 데이터베이스 테이블 관련 명령

- 필드 타입 수정 명령

데이터베이스 테이블 필드 타입 수정 명령

```
alter table 테이블명 modify 필드명 새로운 타입;
```

- ex) name 필드의 타입을 int로 변경
  - mysql> alter table friend modify name int;

- 데이터베이스 테이블명 수정 명령

데이터베이스 테이블명 수정 명령

```
alter table 이전 테이블명 rename 새로운 테이블명;
```

- ex) 테이블명 friend에서 student로 변경
  - mysql> alter table friend rename student;

## 2. 데이터베이스 테이블 관련 명령

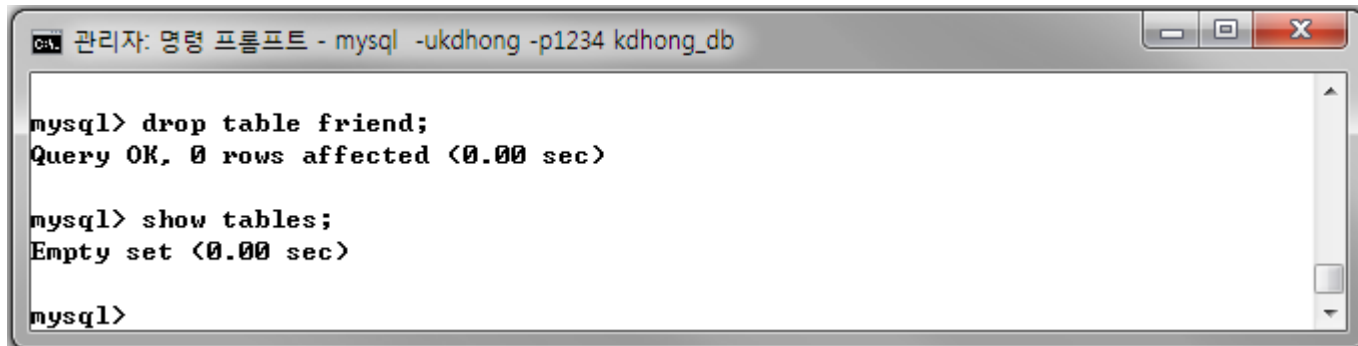
- 테이블 삭제

데이터베이스 테이블 삭제 명령

```
drop table 테이블명;
```

- ex) friend 테이블 삭제

1. mysql> drop table friend;
2. mysql> show tables;



```
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

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

mysql> show tables;
Empty set (0.00 sec)

mysql>
```

[그림 6-21] friend 테이블의 삭제

# 3. SQL 명령의 일괄 실행

---

- 명령 일괄 실행 과정
  1. 텍스트 파일에 데이터베이스 테이블 생성 명령 저장
  2. SQL 명령 일괄 실행 및 실행 확인

### 3. SQL 명령의 일괄 실행

---

1. 텍스트 파일에 데이터베이스 테이블 생성 명령 저장
  - 메모장 “friend2.sql” 이름으로 저장

friend2 데이터베이스 테이블 생성

friend2.sql

```
create table friend2(  
  num int not null,  
  name char(10),  
  address char(80),  
  tel char(20),  
  primary key(num)  
);
```

### 3. SQL 명령의 일괄 실행

---

2. 명령창에서 파일 내용을 copy & paste후 실행

3. `mysql> show tables;`

4. friend 테이블의 구조 점검

1. `mysql> desc friend2;`



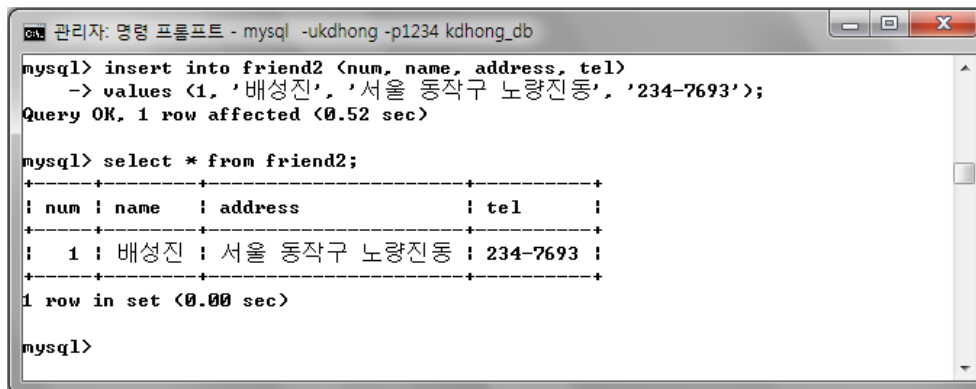
## 4. 레코드 관련 명령

- 테이블에 데이터 삽입

### 레코드 삽입 명령

```
insert into 테이블명 (필드명1, 필드명2, ....) values (필드값1, 필드값2, ...);
```

- ex) '배성진'과 관련된 데이터를 삽입 후 select 명령으로 확인
  - mysql> insert into friend2 (num, name, address, tel)  
-> values (1, '배성진', '서울 동작구 노량진동',  
-> '234-7693');
  - mysql> select \* from friend2;



```
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db
mysql> insert into friend2 (num, name, address, tel)
-> values (1, '배성진', '서울 동작구 노량진동', '234-7693');
Query OK, 1 row affected (0.52 sec)

mysql> select * from friend2;
+----+-----+-----+-----+
| num | name  | address          | tel    |
+----+-----+-----+-----+
| 1   | 배성진 | 서울 동작구 노량진동 | 234-7693 |
+----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

[그림 6-25] friend2  
테이블에 레코드 삽입

## 4. 레코드 관련 명령

---

- 데이터베이스 테이블 생성 및 다수의 레코드 삽입
  1. 데이터베이스 테이블 설계
  2. mem.sql 일괄 실행
  3. 생성된 데이터베이스 테이블 및 테이블 구조 확인
  4. 데이터베이스 mem에 저장된 레코드 확인

## 4. 레코드 관련 명령

### 1. 데이터베이스 테이블 설계

[표 6-2] 회원관리 데이터베이스 테이블 설계(테이블명: mem)

필드명	타입	NULL	설명
num	int	no	일련번호
id	char(15)	no	아이디
name	char(10)	no	이름
gender	char(1)	yes	성별(남성:'M',여성:'W')
post_num	char(8)	yes	우편번호
address	char(80)	yes	주소
tel	char(20)	yes	전화번호
age	int	yes	나이

# 4. 레코드 관련 명령

## 2. mem.sql 일괄 실행

mem 데이터베이스 테이블 생성

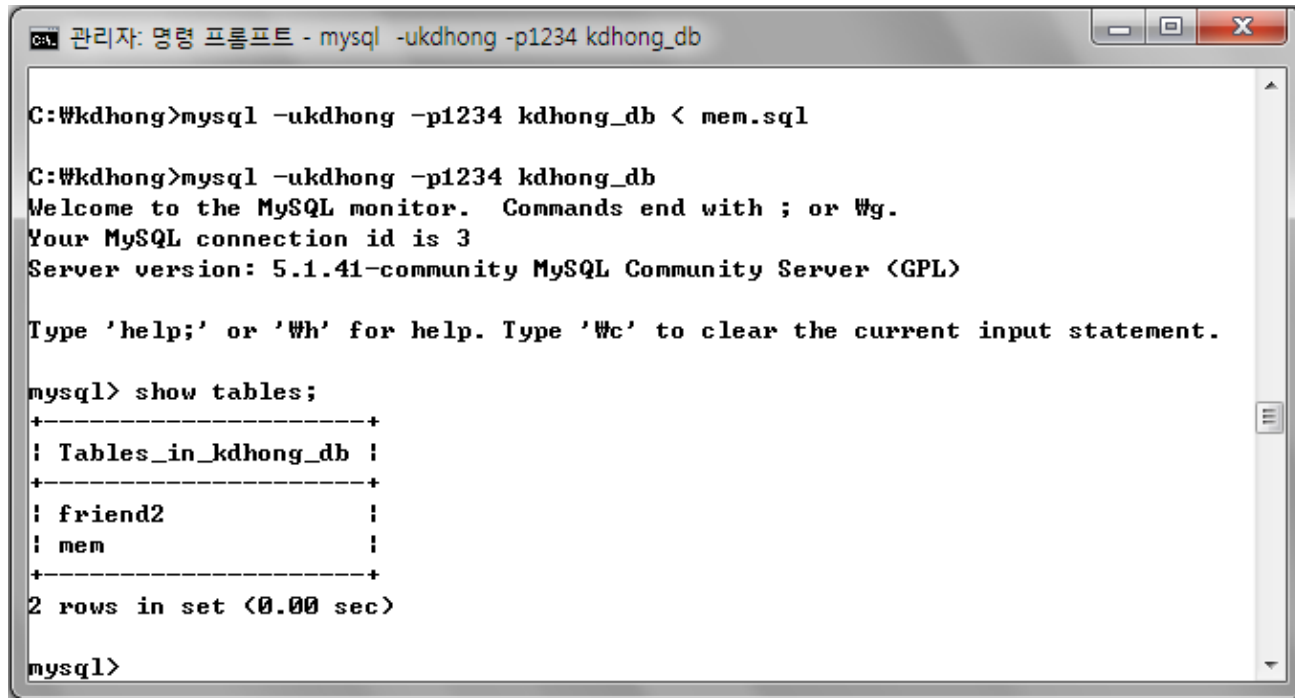
mem.sql

```
create table mem (  
  num int not null,  
  id char(15) not null,  
  name char(10) not null,  
  sex char(1),  
  post_num char(8),  
  address char(80),  
  tel char(20),  
  age int,  
  primary key(num)  
);  
  
insert into mem values (1, 'yjhwang', '황영주', 'M', '100-011', '서울시 중구  
충무로1가', '234-8879', 35);  
insert into mem values (2, 'khshul', '설기형', 'M', '607-010', '부산시 동래구  
명륜동', '764-3784', 33);  
insert into mem values (3, 'chpark', '박철호', 'M', '503-200', '광주시 남구  
지석동', '298-9730', 34);  
.....
```

## 4. 레코드 관련 명령

### 3. 생성된 데이터베이스 테이블 및 테이블 구조 확인

- mysql> show tables;



```
C:\Wkdhong>mysql -ukdhong -p1234 kdhong_db < mem.sql

C:\Wkdhong>mysql -ukdhong -p1234 kdhong_db
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
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> show tables;
+-----+
| Tables_in_kdhong_db |
+-----+
| friend2              |
| mem                  |
+-----+
2 rows in set (0.00 sec)

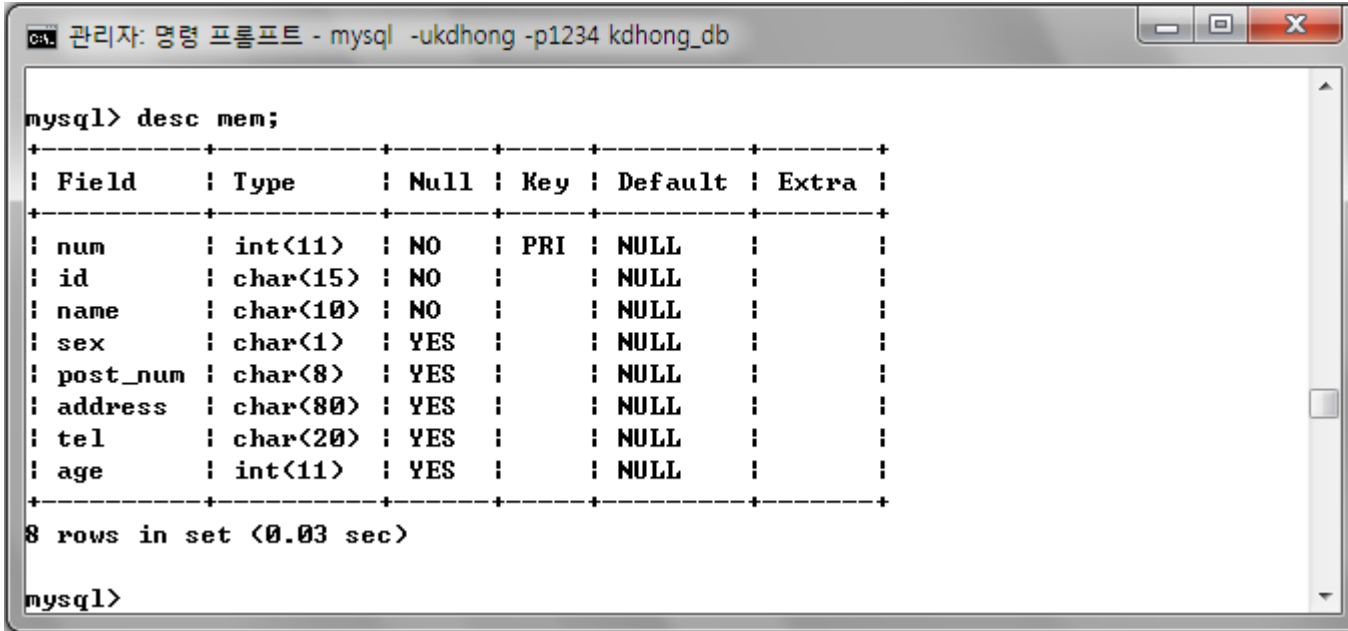
mysql>
```

[그림 6-27] mem.sql 일괄 실행 후, mem 테이블의 존재 확인

## 4. 레코드 관련 명령

### 4. 생성된 데이터베이스 테이블 및 테이블 구조 확인

1. mysql> desc mem;



The screenshot shows a MySQL command prompt window titled "관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong\_db". The command "mysql> desc mem;" has been entered, and the output is a table describing the structure of the 'mem' table. The table has 8 rows and 6 columns: Field, Type, Null, Key, Default, and Extra. The fields are num (int(11), NO, PRI, NULL), id (char(15), NO, NULL), name (char(10), NO, NULL), sex (char(1), YES, NULL), post\_num (char(8), YES, NULL), address (char(80), YES, NULL), tel (char(20), YES, NULL), and age (int(11), YES, NULL). The output also indicates "8 rows in set (0.03 sec)".

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	
id	char(15)	NO		NULL	
name	char(10)	NO		NULL	
sex	char(1)	YES		NULL	
post_num	char(8)	YES		NULL	
address	char(80)	YES		NULL	
tel	char(20)	YES		NULL	
age	int(11)	YES		NULL	

8 rows in set (0.03 sec)

mysql>

[그림 6-28] mem 테이블의 구조 확인

### 5. 데이터베이스 mem에 저장된 레코드 확인

1. mysql> select \* from mem;

## 4.3 레코드 검색 명령

---

- 특정 필드 데이터 검색 명령

특정 필드에 입력된 데이터 검색 명령

```
select 필드명1, 필드명2 from 테이블명;
```

- mysql> select id, name, address from mem;

- 전체 필드 데이터 검색 명령

- mysql> select \*from mem;

- 조건에 맞는 레코드 검색 명령

조건에 맞는 레코드 검색 명령

```
select 필드명1, 필드명2 from 테이블명 where 조건식;
```

## 4.3 레코드 검색 명령

---

- ex) 여성의 아이디, 이름, 주소, 전화번호, 성별 보기
  - `mysql> select id, name, address, tel,  
    -> gender from mem where gender = 'W' ;`
- ex) 50세 이상인 레코드의 전체 필드 보기
  - `mysql> select * from mem where age>=50;`



## 4.3 레코드 검색 명령

---

- ex) 20대의 이름, 아이디, 주소, 우편번호 보기
  - `mysql> select name, id, address, post_num from mem`  
    `-> where age>=20 and age<30;`
- ex) 김진모 레코드의 아이디, 주소, 우편번호, 나이 출력
  - `mysql> select name, id, address, post_num, age`  
    `-> from mem where name='김진모';`

## 4.3 레코드 검색 명령

---

- ex) 40대 남성의 이름, 주소, 나이 보기
  - `mysql> select name, address, age from mem`  
→ `where (age>=40 and age<50) and gender='M';`
- ex) 20대 또는 40대 여성의 이름, 아이디, 주소, 전화번호, 나이 성별 보기
  - `mysql> select name, id, address, tel, age from mem`  
→ `where ((age>=20 and age<30) or (age>=40`  
→ `and age<50)) and gender = 'W';`

## 4.3 레코드 검색 명령

---

- 특정 문자열이 포함된 레코드 검색 명령

특정 문자열이 포함된 레코드 검색 명령

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

- 성이 김씨인 사람의 이름, 주소, 전화번호 보기
  - mysql> select name, address, tel from mem where  
-> name like '김%';

## 4.3 레코드 검색 명령

---

- ex) 서울에 사는 사람의 이름, 주소 보기
  - `mysql> select name, address from mem where address  
-> like '서울%';`
- ex) 부산 사는 여성의 이름, 주소, 성별 보기
  - `mysql> select name, address, gender from mem  
-> where address like '부산%' and gender ='W ';`

## 4.3 레코드 검색 명령

---

- ex) 가운데 이름이 '용'인 사람의 이름, 아이디 보기
  - `mysql> select name, id from mem where name like '__용%' ;`
- ex) 광주에 사는 김씨의 이름, 주소, 전화번호 출력
  - `mysql> select name, address, tel from mem`  
    `-> where address like '광주%' and name like '김%';`

## 4.4 레코드 정렬 명령

### 레코드 정렬 명령

```
select 필드명1, 필드명2 from 테이블 명 order by 필드명;
```

- ex) 나이순 정렬, age, id, name, tel 필드 검색
  - 오름차순 정렬 명령

```
mysql> select age, id, name, gender, tel from mem  
-> order by age;
```
  - 내림차순 정렬 명령

```
mysql> select age, name, address from mem  
-> where address like '서울%' order by age desc;
```

## 4.5 레코드 수정 명령

---

- 레코드 수정 명령

레코드 수정 명령

```
update 테이블명 set 필드명=필드값 [where 조건식]
```

- ex) mem 테이블에서 아이디가 yjhwang인 레코드의 전화번호를 123-4567로 변경
  1. mysql> update mem set tel='123-4567' where id='yjhwang';
  2. mysql> select id, name, tel from mem where id='yjhwang';

## 4.5 레코드 수정 명령

---

- ex) mem 테이블에 속한 신수진의 나이를 27세로 변경
  1. `mysql> update mem set age=27 where name='신수진';`
  2. `mysql> select name, age from mem where name='신수진';`



## 4.6 레코드 삭제 명령

---

조건식을 만족하는 특정 레코드 삭제 명령

```
delete from 테이블명 where 조건식
```

- ex) mem 테이블에서 이름이 김길수인 레코드를 삭제
  1. mysql> delete from mem where name= '김길수';
  2. mysql> select \* from mem where name= '김길수';

## 4.6 레코드 삭제 명령

---

- 30~50세 레코드 삭제
  - 나이가 30~50세인 레코드를 삭제
    1. `mysql> delete from mem where age>=30 and age<=50;`
    2. `mysql> select name, address, age from mem;`
- 전체 레코드 삭제
  - `mysql> delete from mem;`

# 5. 데이터베이스 백업 및 복원 명령

- 데이터베이스 백업

## 데이터베이스 백업 명령

```
mysqldump -u계정 -p비밀번호 데이터베이스명 > 백업파일명
```

- kdhong 계정의 kdhong\_db 를 kdhong\_db.sql에 백업
  - C:\Wkdhong> mysqldump -ukdhong -p1234 kdhong\_db  
-> kdhong\_db.sql

- 백업 파일 복원

## 데이터베이스 백업 파일 복원 명령

```
mysql -u계정 -p비밀번호 데이터베이스명 < 백업 파일명
```

- kdhong\_db.sql을 다른 사용자 계정에 복원
  - 계정 : kim, 비밀번호 : 1234, 데이터베이스명 : kim\_db
  - C:\Wkdhong>mysql -ukim -p1234 kim\_db < kdhong\_db.sql