

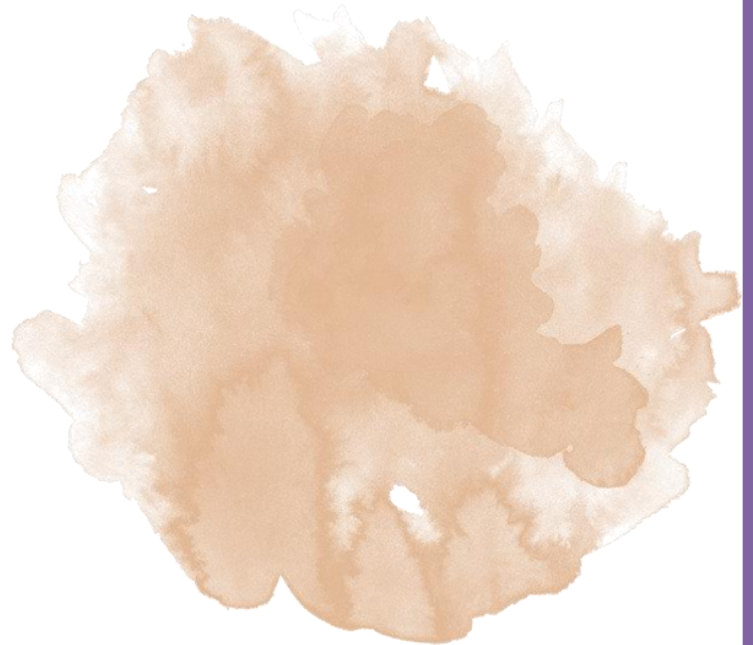


대한민국산업현장교수 김형주

06장 MySQL 명령

학습목표

- ✓ SQL 명령 사용법
- ✓ SQL 명령의 일괄 실행
- ✓ 데이터베이스 백업/복원



목차

1. 데이터베이스 관련 명령
2. 데이터베이스 테이블 관련 명령
3. SQL 명령의 일괄 실행
4. 레코드 관련 명령
5. 데이터베이스 백업 및 복원 명령



1. 데이터베이스 관련 명령

- 데이터베이스 접속 명령

데이터베이스 접속 명령

```
mysql -u계정 -p비밀번호 데이터베이스명
```

- C:₩> mysql - ukhj - pquer khj_db

- 데이터베이스 생성 명령

데이터베이스 생성 명령

```
create database 데이터베이스명;
```

- mysql> create database sample1;

[참고]여기에서는 5장에서 만든 비밀번호가 1234인 kdhong 계정과 kdhong_db 데이터베이스를 사용한다

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

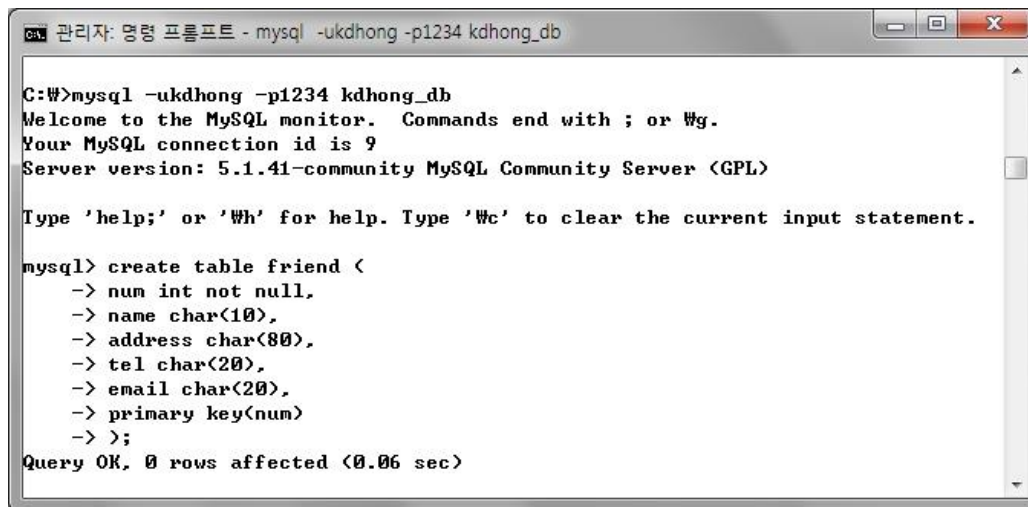
- 테이블 생성 명령

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)
 - C:\W> mysql -ukhj -pqrwer khj_db
 - mysql> create table friend (
 - > num int not null,
 - > name char(10),
 - > address char(80),
 - > tel char(20),
 - > email char(20),
 - > primary key(num)
 - >);



```
C:\W>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

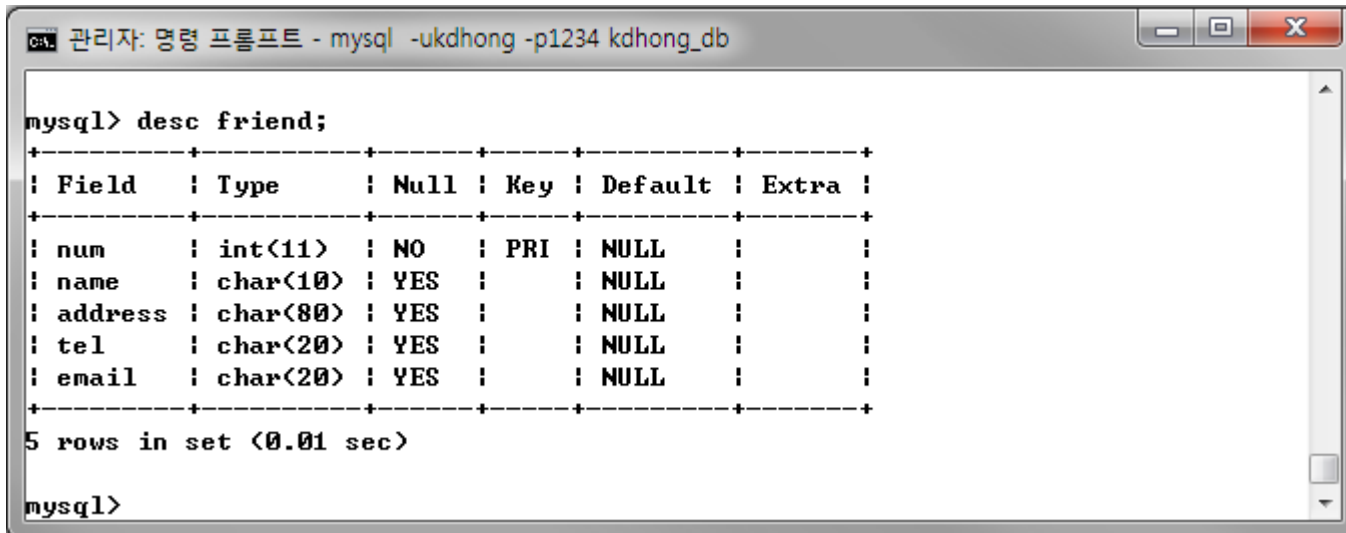
2. 데이터베이스 테이블 관련 명령

- 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

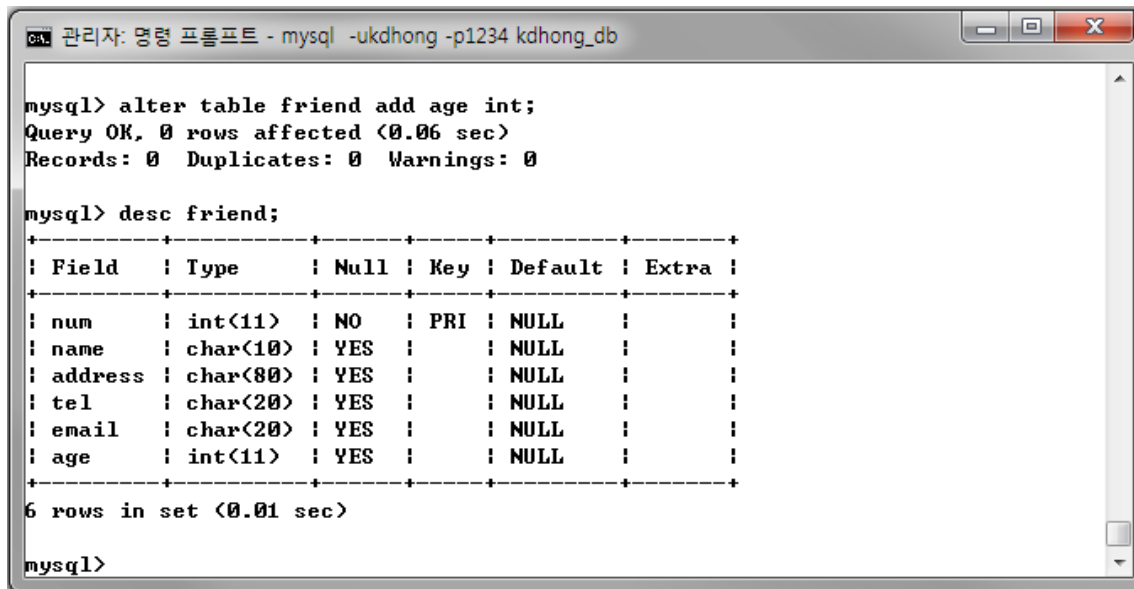
2. 데이터베이스 테이블 관련 명령

- 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

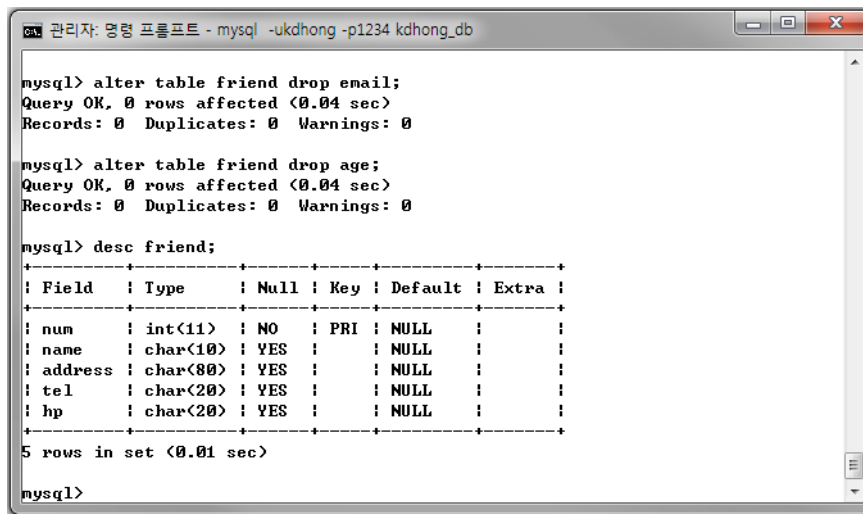
- 필드 삭제

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

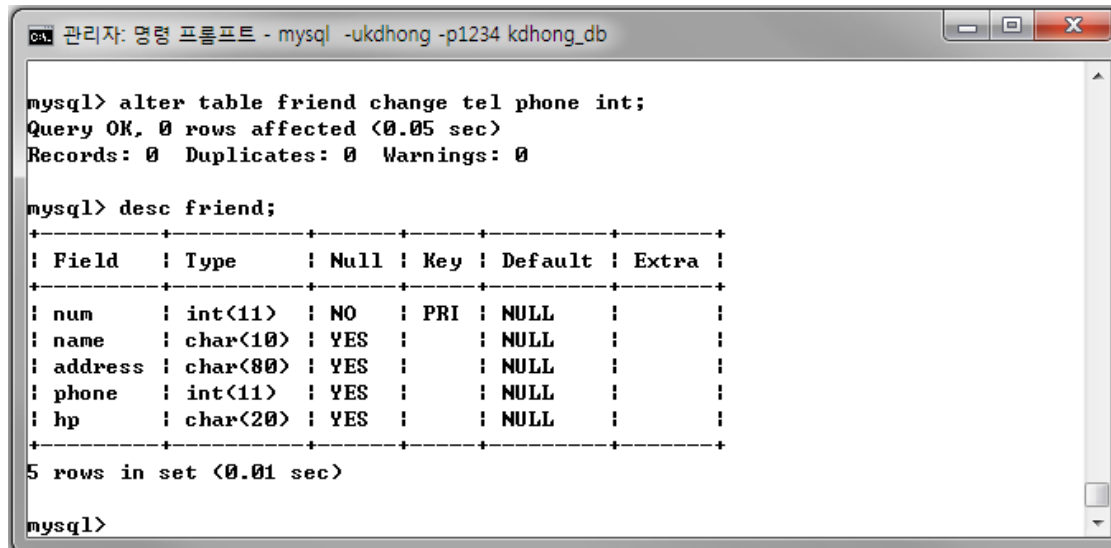
2. 데이터베이스 테이블 관련 명령

- 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

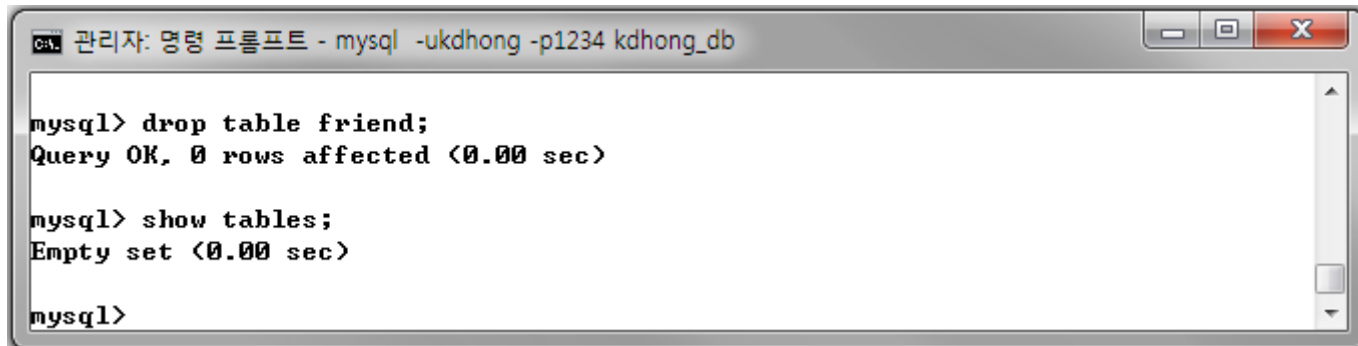
- 테이블 삭제

데이터베이스 테이블 삭제 명령

```
drop table 테이블명;
```

- ex) friend 테이블 삭제

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



The screenshot shows a Windows-style command prompt window titled "관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db". The window contains the following text:

```
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. C:\W>APM_Setup>Server>MySQL>bin> 폴더로 이동
 3. SQL 명령 일괄 실행 및 실행 확인

3. SQL 명령의 일괄 실행

1. 텍스트 파일에 데이터베이스 테이블 생성 명령 저장

- C 드라이브에 [C:\₩>APM_Setup>Server>MySQL>bin>]폴더 생성
- 메모장 “friend.sql” 이름으로 저장

friend2 데이터베이스 테이블 생성

friend.sql

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


3. SQL 명령의 일괄 실행

2. bin 폴더로 이동

1. `C:\W>APM_Setup>Server>MySQL>bin>dir`
2. `bin> dir fri*`

3. SQL 명령 일괄 실행 및 실행 확인

1. `C:\Wbin> mysql -ukhj -pquer khj_db < friend.sql`
2. `C:\Wbin> mysql -ukhj -pquer khj_db`
3. `mysql> show tables;`

4. friend 테이블의 구조 점검

1. `mysql> desc friend;`

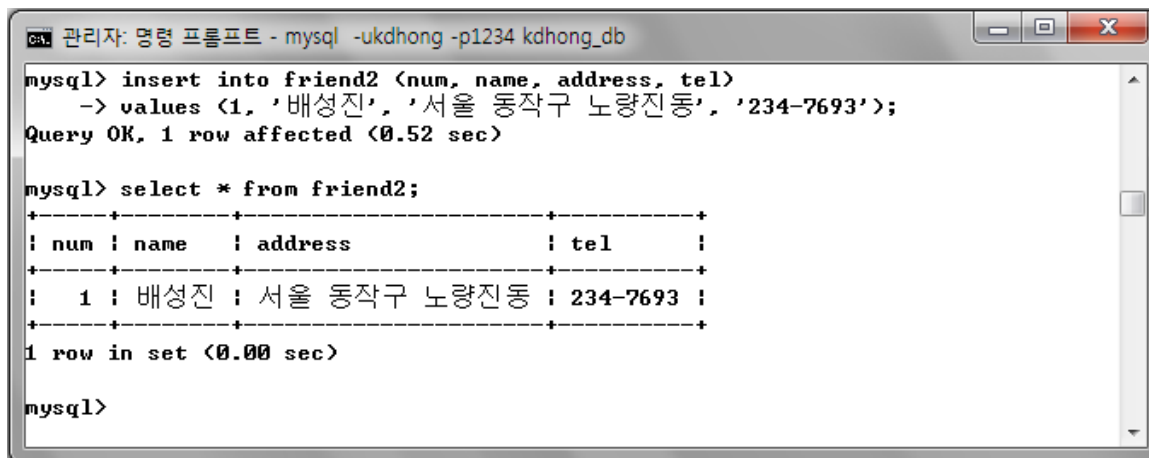
4. 레코드 관련 명령

- 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 friend (num, name, address, tel, email)
-> values (1, '김형주', '인천 미추홀구 인하로',
-> '123-4567', 'kh4444@naver.com');
 - mysql> select * from friend;



```
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] friend
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 (num, id, name, gender, post_num, address, tel, age) values (1  
, "kh4444", "김형주", "M", "607-010", "부산시 동래구 명륜동", "764-3784", 33);  
insert into mem (num, id, name, gender, post_num, address, tel, age) values (2  
, "chpark", "박철호", "M", "503-200", "광주시 남구 지석동", "298-9730", 34);  
insert into mem (num, id, name, gender, post_num, address, tel, age) values (3  
, "parkyh", "박요한", "M", "411-300", "경기도 성남시 남구 승학동", "856-2730",  
24);  
insert into mem (num, id, name, gender, post_num, address, tel, age) values (4  
, "sjjang", "장수진", "W", "333-111", "서울시 중구 인사동", "324-1243", 21);  
insert into mem (num, id, name, gender, post_num, address, tel, age) values (5  
, "jwchoi", "최정원", "W", "432-121", "서울시 용산구 구미동", "987-1234", 23);
```

4. 레코드 관련 명령

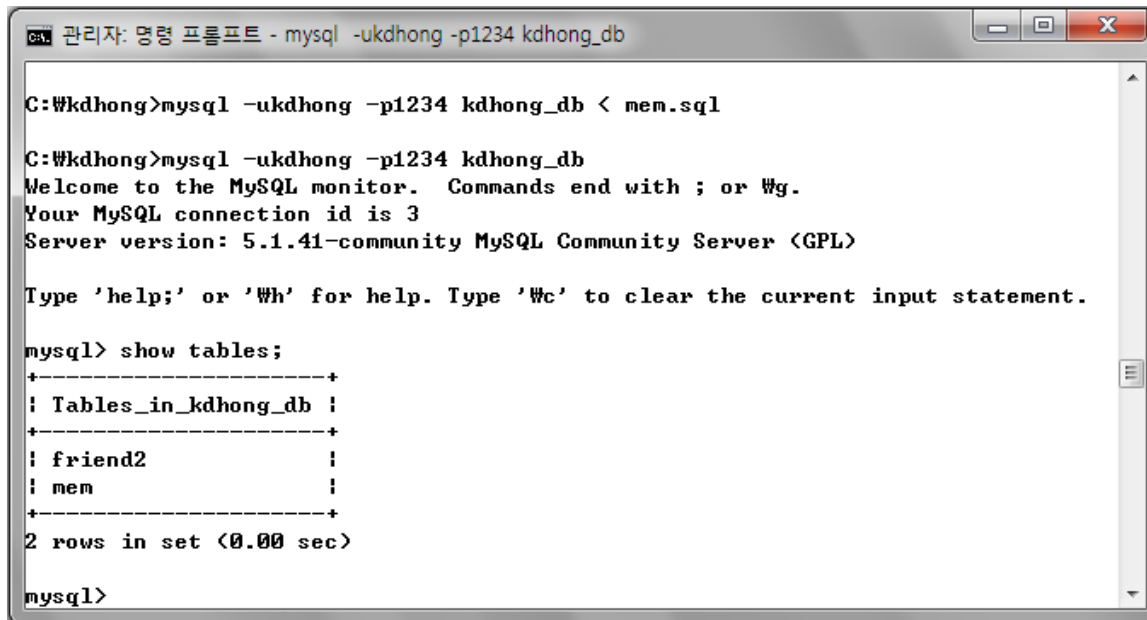
2. mem.sql의 일괄 실행

1. C:\Wbin> mysql -ukhj -pqrwer khj_db < mem.sql

3. 생성된 데이터베이스 테이블 및 테이블 구조 확인

2. C:\W> mysql -ukhj -pqrwer khj_db

3. mysql> show tables;



The screenshot shows a Windows command prompt window titled "관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db". The command prompt shows the following sequence of commands and output:

```
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

3. 생성된 데이터베이스 테이블 및 테이블 구조 확인

1. `mysql> desc mem;`

```
mysql> desc mem;
```

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

4. 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;

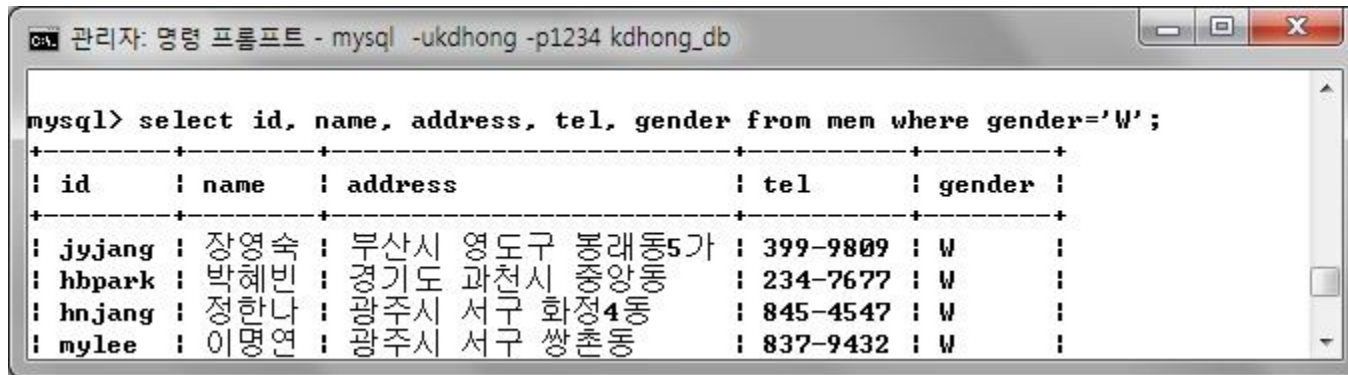
- 조건에 맞는 레코드 검색 명령

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 ' ;



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

mysql> select id, name, address, tel, gender from mem where gender='W';

+----+-----+-----+-----+-----+
| id  | name  | address          | tel    | gender |
+----+-----+-----+-----+-----+
| jyjang | 장영숙 | 부산시 영도구 봉래동5가 | 399-9809 | W      |
| hbpark | 박혜빈 | 경기도 과천시 중앙동   | 234-7677 | W      |
| hnjang | 정한나 | 광주시 서구 화정4동     | 845-4547 | W      |
| mylee  | 이명연 | 광주시 서구 쌍촌동     | 837-9432 | W      |
+----+-----+-----+-----+-----+
```

- ex) 30세 이상인 레코드의 전체 필드 보기
 - mysql> select * from mem where age>=30;



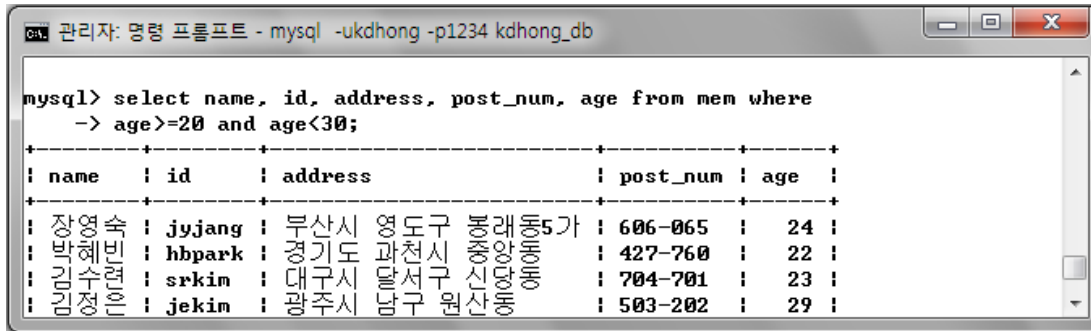
```
CA. 관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

mysql> select * from mem where age>=30;

+----+-----+-----+-----+-----+-----+-----+
| num | id    | name  | gender | post_num | address          | tel    |
+----+-----+-----+-----+-----+-----+-----+
| 8   | mskim | 김문수 | M      | 429-020  | 경기도 시흥시 신천동 | 370-6003 |
| 10  | kskim | 김길수 | M      | 440-747  | 경기도 수원시 장안구 파장동 | 324-5875 |
| 13  | hnjang | 정한나 | W      | 502-763  | 광주시 서구 화정4동   | 845-4547 |
+----+-----+-----+-----+-----+-----+-----+
```

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;

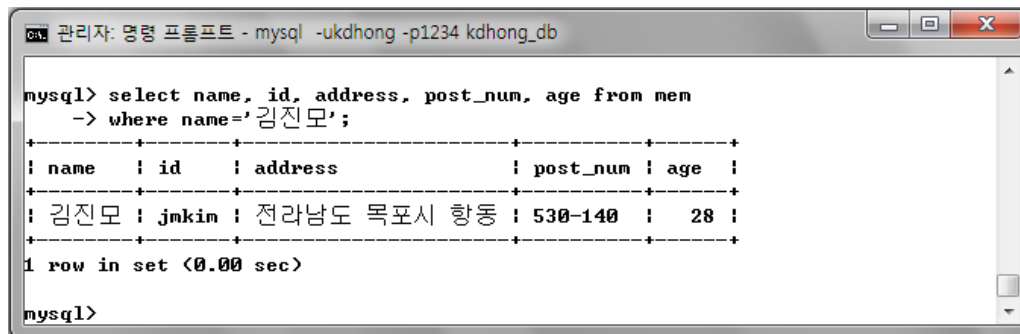


관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

```
mysql> select name, id, address, post_num, age from mem where  
-> age>=20 and age<30;
```

name	id	address	post_num	age
장영숙	jjjang	부산시 영도구 봉래동5가	606-065	24
박혜빈	hbpark	경기도 과천시 중앙동	427-760	22
김수련	srkin	대구시 달서구 신명동	704-701	23
김정은	jekin	광주시 남구 원산동	503-202	29

- ex) 김형주 레코드의 아이디, 주소, 우편번호, 나이 출력
 - mysql> select name, id, address, post_num, age
-> from mem where name='김형주';



관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

```
mysql> select name, id, address, post_num, age from mem  
-> where name='김진모';
```

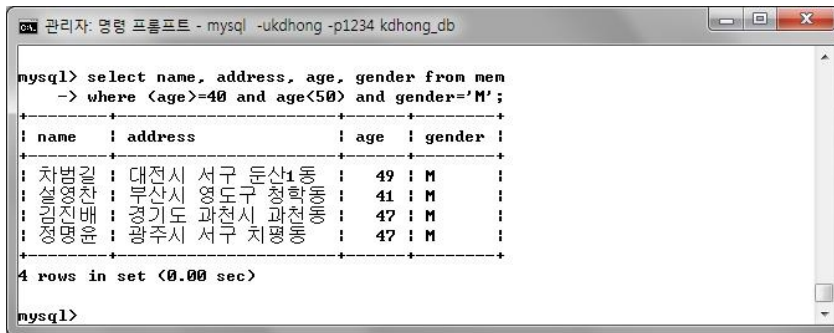
name	id	address	post_num	age
김진모	jmkin	전라남도 목포시 항동	530-140	28

1 row in set (0.00 sec)

```
mysql>
```

4.3 레코드 검색 명령

- ex) 30대 남성의 이름, 주소, 나이 보기
 - mysql> select name, address, age from mem
-> where (age>=30 and age<40) and gender='M';



관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

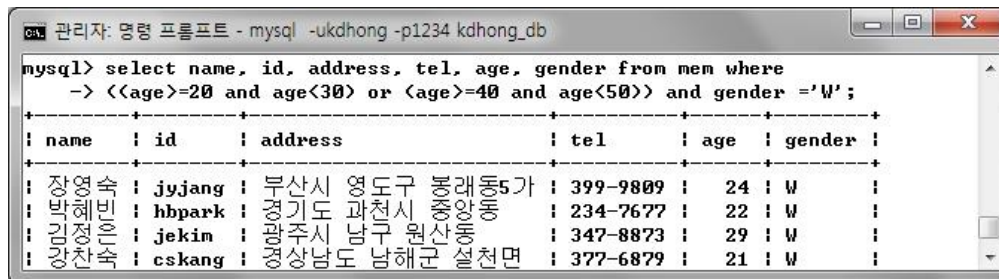
```
mysql> select name, address, age, gender from mem  
-> where (age>=30 and age<40) and gender='M';
```

name	address	age	gender
차범길	대전시 서구 둔산1동	49	M
설영찬	부산시 영도구 청학동	41	M
김진배	경기도 과천시 과천동	47	M
정영훈	광주시 서구 치평동	47	M

4 rows in set (0.00 sec)

mysql>

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



관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

```
mysql> select name, id, address, tel, age, gender from mem where  
-> ((age>=20 and age<30) or (age>=40 and age<50)) and gender = 'W';
```

name	id	address	tel	age	gender
장영숙	jjjang	부산시 영도구 봉래동5가	399-9809	24	W
박혜빈	hhpark	경기도 과천시 중양동	234-7677	22	W
김정은	jekim	광주시 남구 원산동	347-8873	29	W
강찬숙	cskang	경상남도 남해군 설천면	377-6879	21	W

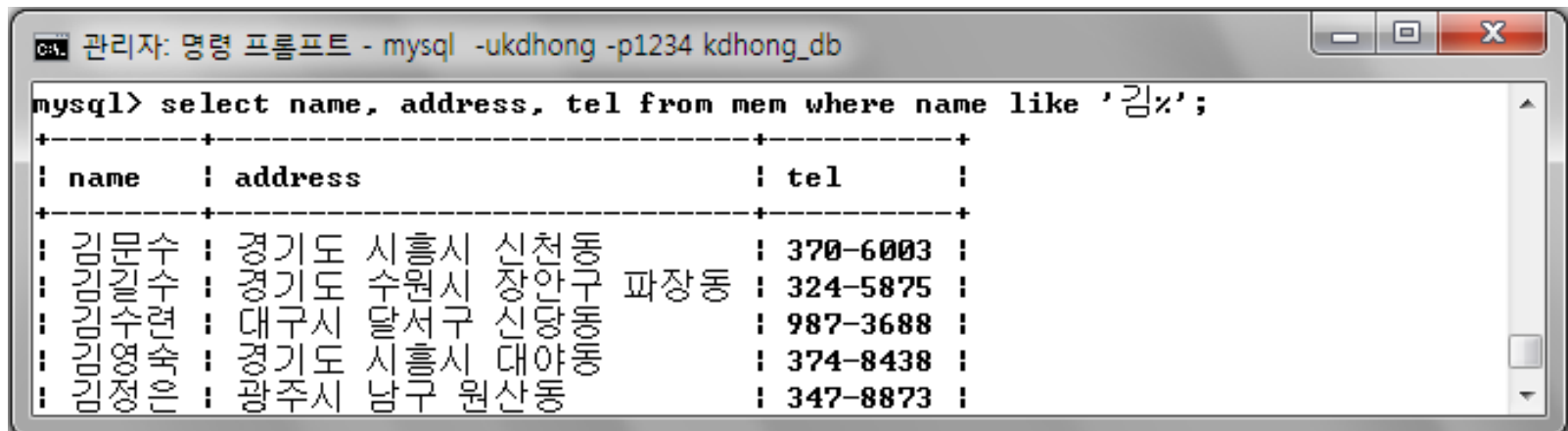
4.3 레코드 검색 명령

- 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%';



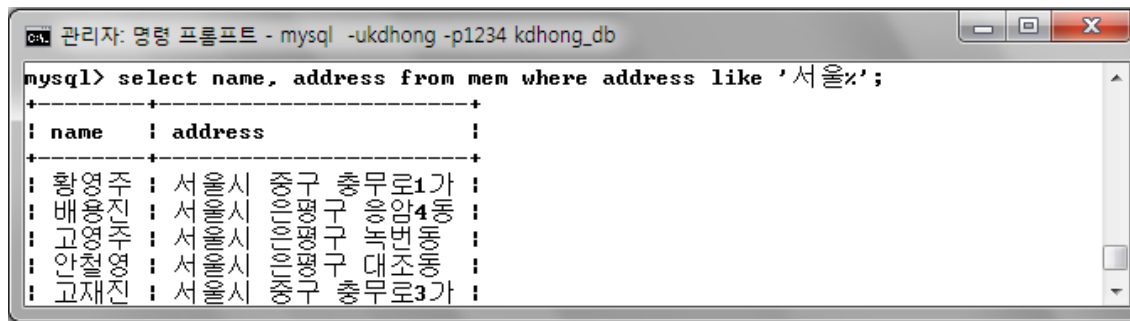
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

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

name	address	tel
김문수	경기도 시흥시 신천동	370-6003
김길수	경기도 수원시 장안구 파장동	324-5875
김수련	대구시 달서구 신당동	987-3688
김영숙	경기도 시흥시 대야동	374-8438
김정은	광주시 남구 원산동	347-8873

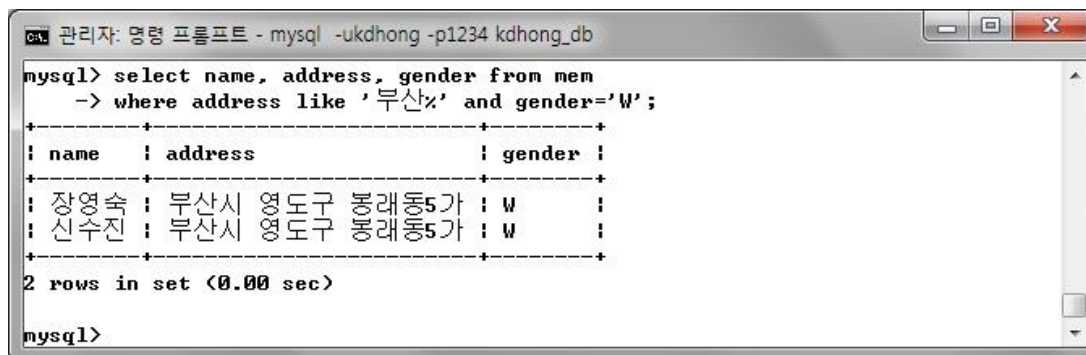
4.3 레코드 검색 명령

- ex) 서울에 사는 사람의 이름, 주소 보기
 - mysql> select name, address from mem where address
-> like '서울%';



```
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db
mysql> select name, address from mem where address like '서울%';
+-----+-----+
| name | address |
+-----+-----+
| 황영주 | 서울시 중구 충무로1가 |
| 배용진 | 서울시 중구 용암4동 |
| 고영주 | 서울시 중구 녹번동 |
| 안철영 | 서울시 중구 대조동 |
| 고재진 | 서울시 중구 충무로3가 |
+-----+-----+
```

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

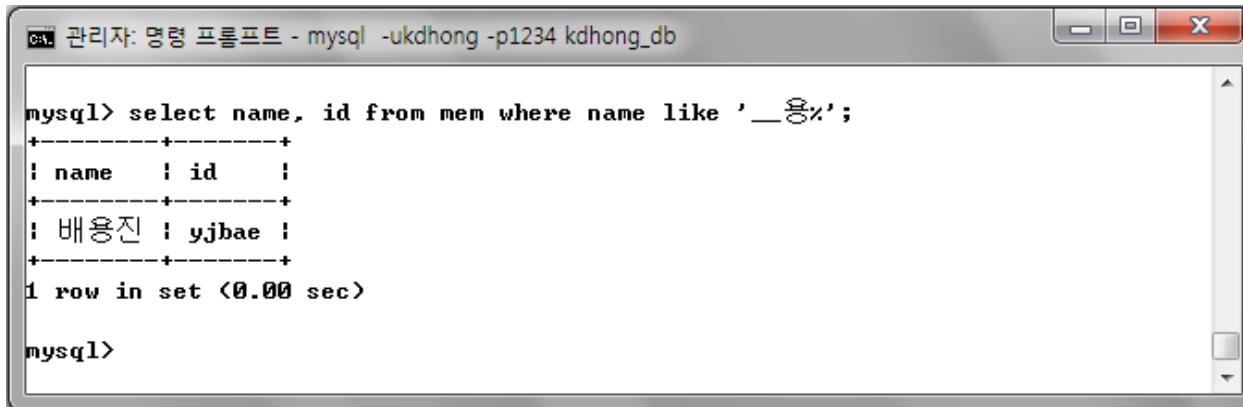


```
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db
mysql> select name, address, gender from mem
-> where address like '부산%' and gender='W' ;
+-----+-----+-----+
| name | address | gender |
+-----+-----+-----+
| 장영숙 | 부산시 영도구 봉래동5가 | W |
| 신수진 | 부산시 영도구 봉래동5가 | W |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

4.3 레코드 검색 명령

- ex) 가운데 이름이 '요'인 사람의 이름, 아이디 보기
 - mysql> select name, id from mem where name like '__요%' ;

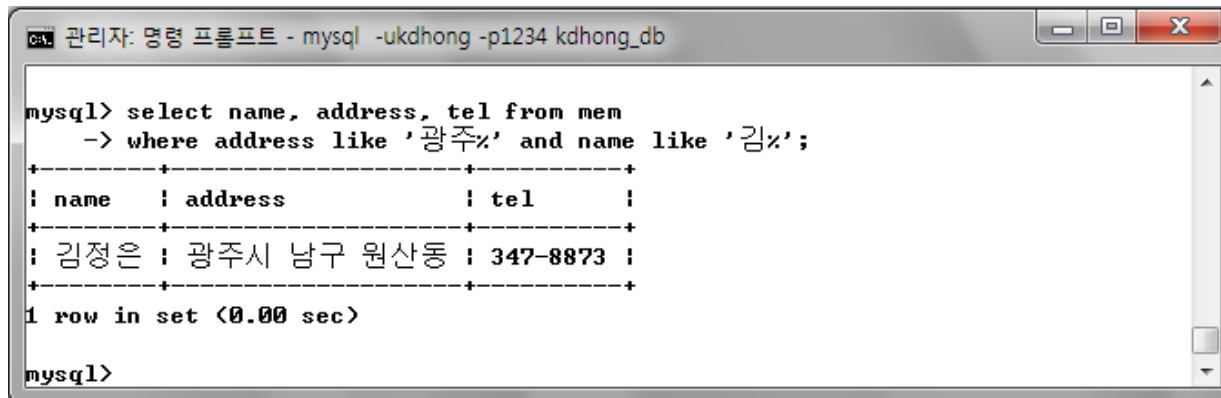


The screenshot shows a MySQL command prompt window titled "관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db". The command entered is "mysql> select name, id from mem where name like '__요%' ;". The output is a table with two columns: "name" and "id". The first row contains the values "배용진" and "yjbae". Below the table, it says "1 row in set (0.00 sec)".

```
mysql> select name, id from mem where name like '__요%';
+-----+-----+
| name  | id    |
+-----+-----+
| 배용진 | yjbae |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

- ex) 광주에 사는 박씨의 이름, 주소, 전화번호 출력
 - mysql> select name, address, tel from mem
-> where address like '광주%' and name like '박%';



The screenshot shows a MySQL command prompt window titled "관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db". The command entered is "mysql> select name, address, tel from mem -> where address like '광주%' and name like '박%';". The output is a table with three columns: "name", "address", and "tel". The first row contains the values "김정은", "광주시 남구 원산동", and "347-8873". Below the table, it says "1 row in set (0.00 sec)".

```
mysql> select name, address, tel from mem
-> where address like '광주%' and name like '박%';
+-----+-----+-----+
| name  | address          | tel    |
+-----+-----+-----+
| 김정은 | 광주시 남구 원산동 | 347-8873 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

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;
```

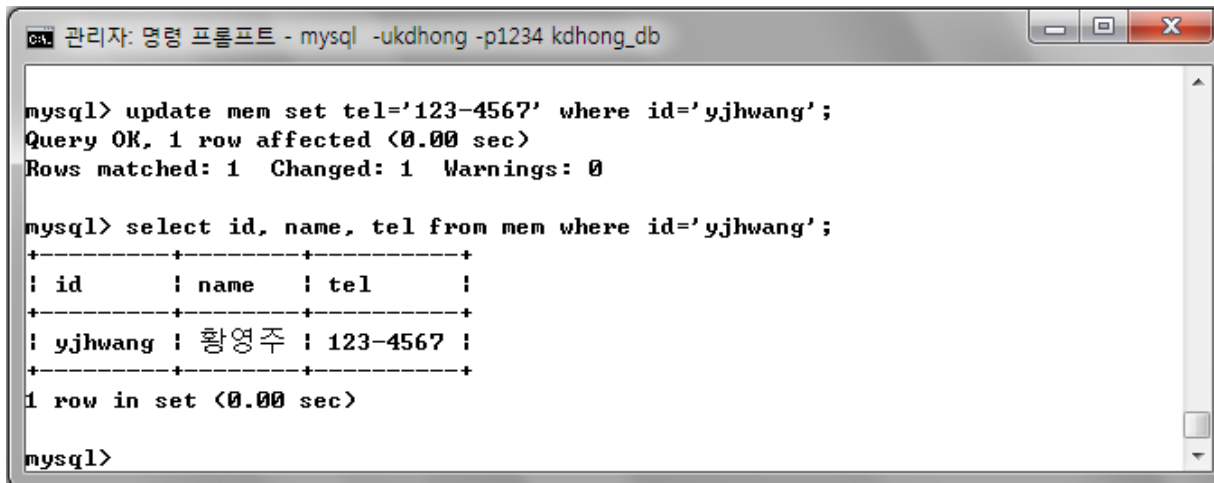
4.5 레코드 수정 명령

- 레코드 수정 명령

레코드 수정 명령

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

- ex) mem 테이블에서 아이디가 sjjang인 레코드의 전화번호를 123-4567로 변경
 - mysql> update mem set tel='123-4567' where id='sjjang';
 - mysql> select id, name, tel from mem where id='sjjang';



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

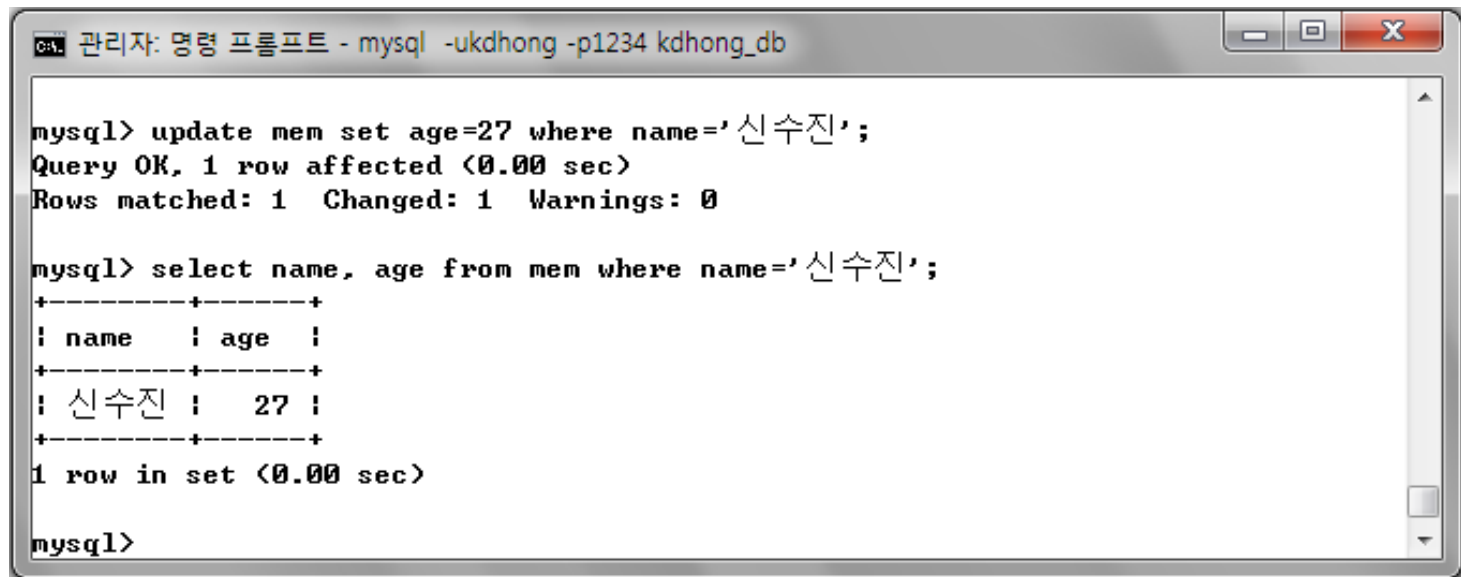
mysql> update mem set tel='123-4567' where id='yjhwan';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select id, name, tel from mem where id='yjhwan';
+----+-----+-----+
| id   | name  | tel   |
+----+-----+-----+
| yjhwan | 황영주 | 123-4567 |
+----+-----+-----+
1 row in set (0.00 sec)

mysql>
```


4.5 레코드 수정 명령

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



The screenshot shows a MySQL command prompt window titled "관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db". The window contains the following text:

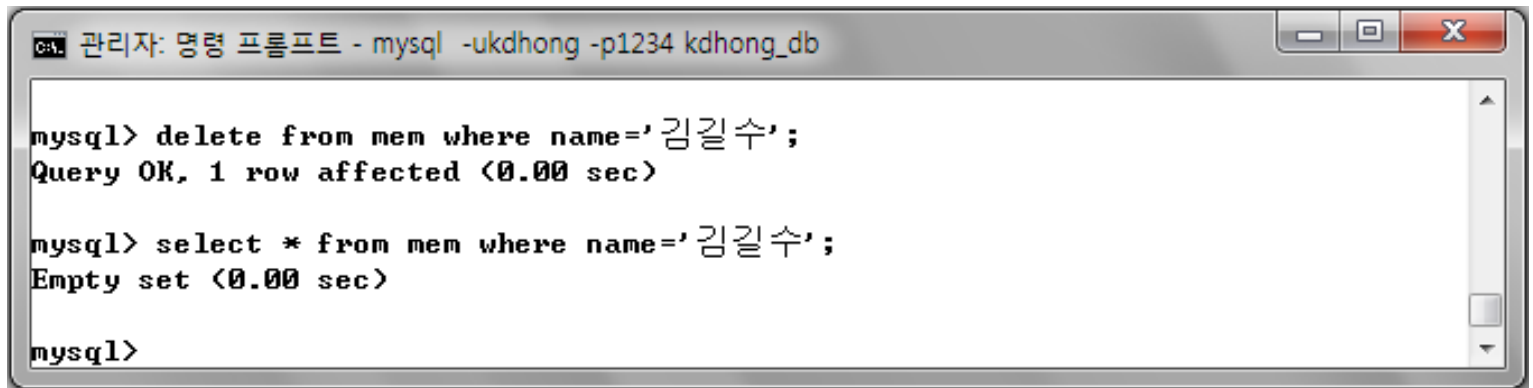
```
mysql> update mem set age=27 where name='신수진';  
Query OK, 1 row affected (0.00 sec)  
Rows matched: 1  Changed: 1  Warnings: 0  
  
mysql> select name, age from mem where name='신수진';  
+-----+-----+  
| name  | age  |  
+-----+-----+  
| 신수진 | 27   |  
+-----+-----+  
1 row in set (0.00 sec)  
  
mysql>
```

4.6 레코드 삭제 명령

조건식을 만족하는 특정 레코드 삭제 명령

```
delete from 테이블명 where 조건식
```

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



```
C:\> 관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

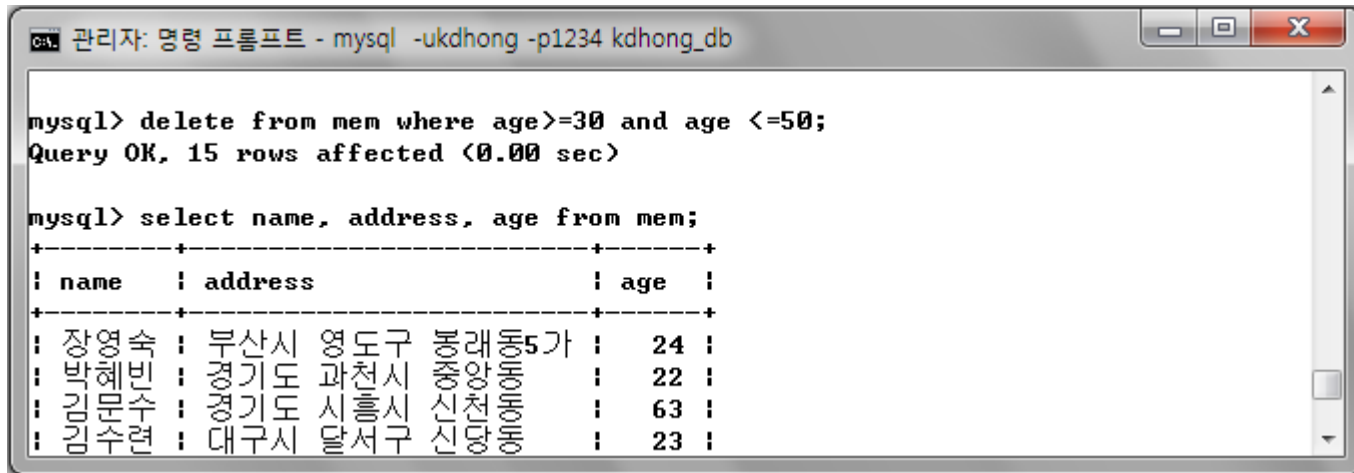
mysql> delete from mem where name='김길수';
Query OK, 1 row affected (0.00 sec)

mysql> select * from mem where name='김길수';
Empty set (0.00 sec)

mysql>
```

4.6 레코드 삭제 명령

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



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

mysql> delete from mem where age>=30 and age <=50;
Query OK, 15 rows affected (0.00 sec)

mysql> select name, address, age from mem;
+-----+-----+-----+
| name  | address                | age  |
+-----+-----+-----+
| 장영숙 | 부산시 영도구 봉래동5가 | 24   |
| 박혜빈 | 경기도 과천시 중앙동   | 22   |
| 김문수 | 경기도 시흥시 신천동   | 63   |
| 김수현 | 대구시 달서구 신당동   | 23   |
```

- 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 khj_db 를 khj_db.sql에 백업

- C:\Wkdhong> mysqldump -ukhj -ppower hhj_db
-> kdhong_db.sql

- 백업 파일 복원

데이터베이스 백업 파일 복원 명령

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

- khj_db.sql을 다른 사용자 계정에 복원

- 계정 : kim, 비밀번호 : 1234, 데이터베이스명 : kim_db
- C:\Wkdhong>mysql -ukim -p1234 kim_db < khj_db.sql