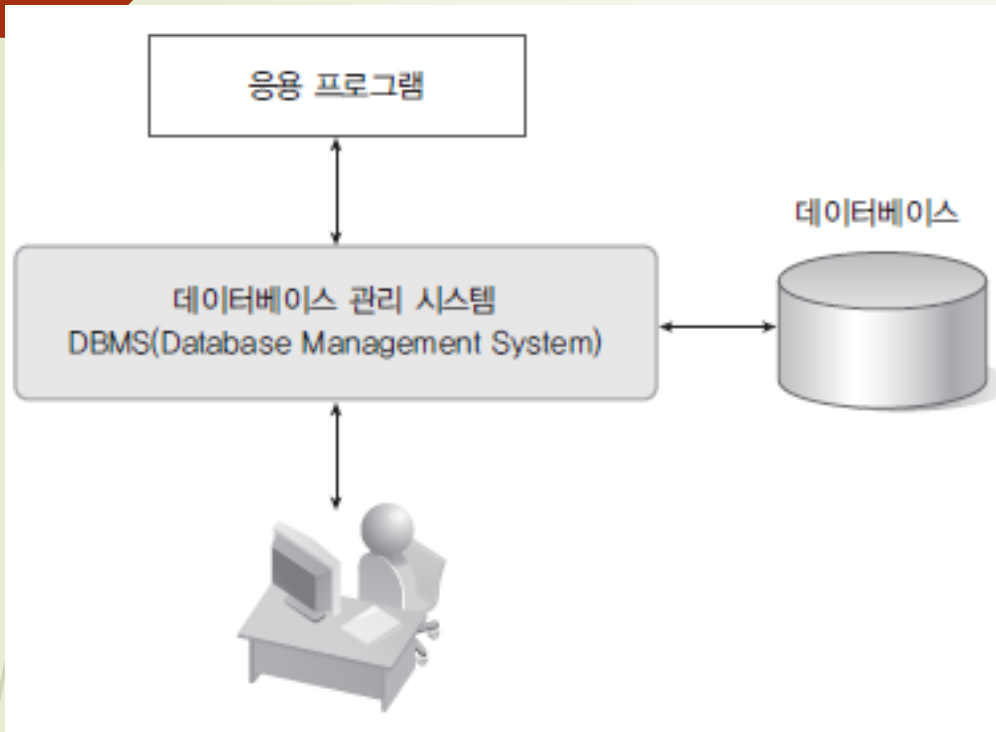


MySQL 데이터베이스

1.1 데이터베이스와 DBMS



[그림 5-1] DBMS와 다른 요소와의 관계

[참고] 데이터베이스 관리 시스템을 통해 데이터베이스에 접근할 수 있으므로, 두 개념을 나누지 않고 합쳐서 데이터베이스라고 부른다.

1.2 관계형 데이터베이스의 구조

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

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

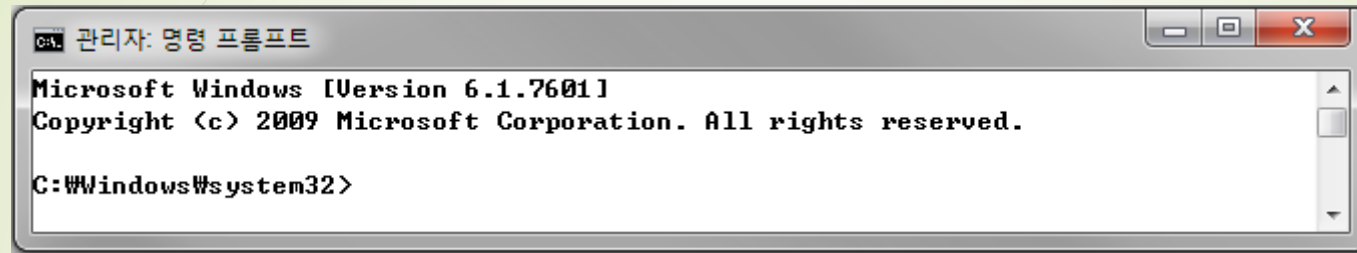
■ 레코드

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

1.3 MySQL의 특징

- SQL에 기반을 둔 관계형 DBMS 중 하나
- Oracle, IBM, Infomix 등의 데이터 베이스는 고가이지만, MySQL 데이터 베이스는 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

2.1 MySQL의 시작



[그림 5-4] 명령 프롬프트 실행

[참고][그림 5-4]에서 시작 위치가 C:\Users\Administrator으로 출력되었지만, 실습하는 컴퓨터 환경에 따라 다르게 나타날 수 있다. 실습하는데 전혀 지장을 주지 않으므로 신경 쓰지 않아도 된다.

➤ 루트 디렉토리로 이동

➤ C:\Windows\Administrator>cd\

2.1 MySQL의 시작

MySQL 사용자 정보

- 계정
- 비밀번호
- 데이터베이스명

MySQL 접속 명령

MySQL 접속 명령 1

```
C:\W> mysql -u계정 -p비밀번호  
mysql> use 데이터베이스명
```

MySQL 접속 명령 2

```
C:\W> mysql -u계정 -p비밀번호 데이터베이스명
```

ex)

계정 : kdhong, 비밀번호:1234, DB명:yangjung 라면,
C:\W> mysql -ukdhong -p1234 yangjung

2.1 MySQL의 시작

관리자 접속

➤ C:\W> mysql -uroot -p1234 mysql

```
C:\Users\kim>mysql -uroot -p1234 mysql
```

```
mysql: [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 154
```

```
Server version: 8.0.13 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>
```

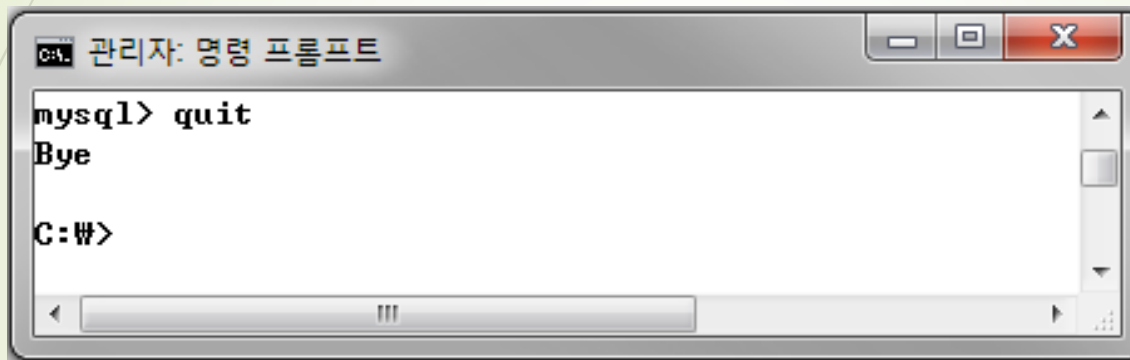
[그림 5-7] 관리자 계정으로 MySQL에 접속

2.2 MySQL 종료

MySQL 종료

➤ mysql> quit

➤ mysql> \q



[그림 5-8] MySQL 접속 종료

3. 새로운 계정 생성

- ▶ 데이터베이스에 관리자 계정으로 접속

- ▶ C:\W> mysql -uroot -p1234

```
C:\Users\kim>mysql -uroot -p1234
mysql: [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 25
Server version: 5.7.20-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, 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>
```


- ▶ 존재하는 데이터베이스 목록보기

- ▶ mysql> show databases;

[그림 5-9] 관리자 계정으로 접속

3. 새로운 계정 생성

- ▶ 테이블 목록보기
 - ▶ `mysql> show tables;`



```
관리자: 명령 프롬프트 - mysql -ur...
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| event           |
| func            |
| general_log     |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| ndb_binlog_index|
| plugin          |
| proc            |
| procs_priv      |
| servers         |
| slow_log        |
| tables_priv     |
| time_zone       |
| time_zone_leap_second|
| time_zone_name  |
| time_zone_transition|
| time_zone_transition_type|
| user            |
+-----+
23 rows in set (0.20 sec)
```

[그림 5-10] mysql 데이터베이스의 테이블

3.2 개인 계정 등록

- 생성하는 계정
 - 계정 : root
 - 비밀번호 : 1234
 - 데이터베이스명 : javadb

① 새로운 데이터베이스 생성

데이터베이스 생성 명령

```
mysql> create database 데이터베이스명;
```

- 이름이 yangjung인 새로운 데이터베이스 생성
 - mysql> create database javadb;

3.2 개인 계정 등록

데이터베이스 목록보기

➡ mysql> show databases;

```
mysql> create database javadb;  
Query OK, 1 row affected (0.04 sec)
```

```
mysql> show databases;
```

Database
information_schema
javadb
mysql
performance_schema
sakila
sampledb
sys
world
yang
yangjung
yangjung1
yangjung2

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

```
mysql>
```

[그림 5-12] javadb 데이터베이스의 존재 확인

3.2 개인 계정 등록

② user 테이블에 계정과 비밀번호 등록

테이블 구조 출력 명령

```
mysql> desc 테이블명;
```

➡ user 테이블의 구조 확인

➡ mysql> desc user;

```
mysql> desc user;
```

Field	Type	Null	Key	Default	Extra
Host	char(60)	NO	PRI		
User	char(32)	NO	PRI		
Select_priv	enum('N','Y')	NO		N	
Insert_priv	enum('N','Y')	NO		N	
Update_priv	enum('N','Y')	NO		N	
Delete_priv	enum('N','Y')	NO		N	
Create_priv	enum('N','Y')	NO		N	
Drop_priv	enum('N','Y')	NO		N	
Reload_priv	enum('N','Y')	NO		N	
Shutdown_priv	enum('N','Y')	NO		N	
Process_priv	enum('N','Y')	NO		N	
File_priv	enum('N','Y')	NO		N	
Grant_priv	enum('N','Y')	NO		N	
References_priv	enum('N','Y')	NO		N	
Index_priv	enum('N','Y')	NO		N	
Alter_priv	enum('N','Y')	NO		N	
Show_db_priv	enum('N','Y')	NO		N	

[그림 5-13] user 테이블의 구조 확인

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 javadb;
```

4. 레코드 관련 명령

[표 6-2] 회원관리 데이터베이스 테이블 설계 (테이블명: mem)

1. 데이터베이스 테이블 설계

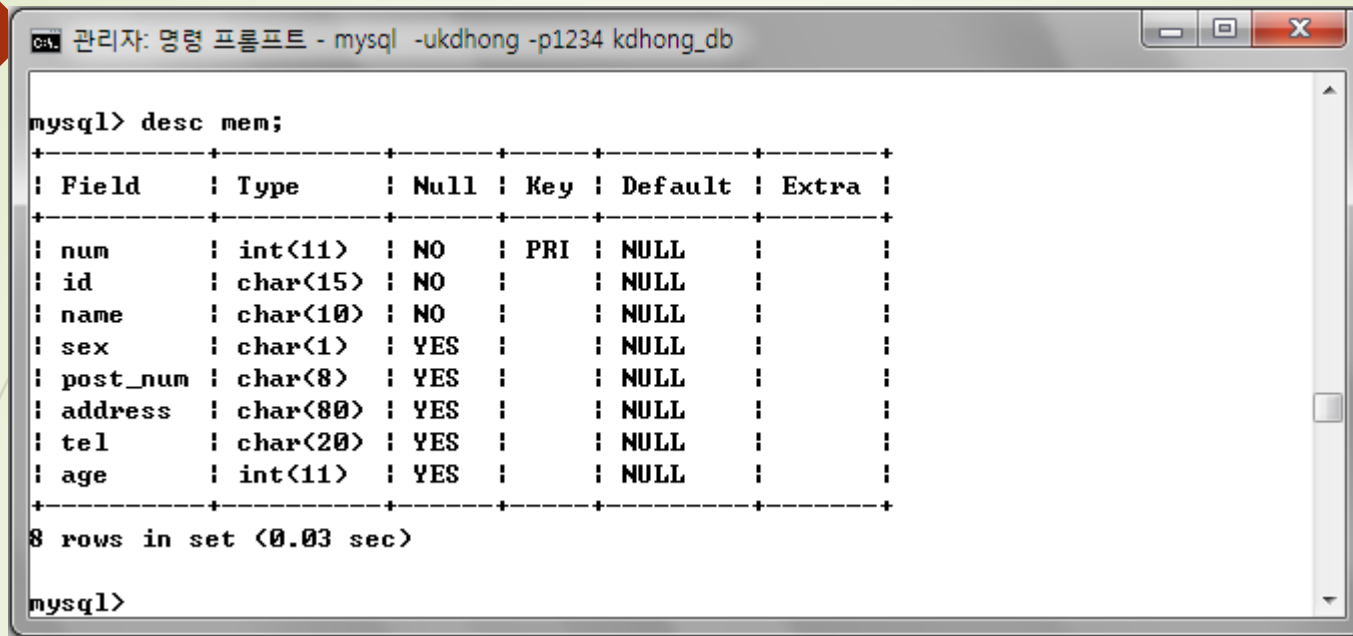
필드명	타입	NULL	설명
num	int	no	일련번호
id	char(15)	no	아이디
name	char(10)	no	이름
gender	char(1)	yes	성별 (남성: 'M', 여성: 'W')
post_num	char(8)	yes	우편번호
address	char(80)	yes	주소
tel	char(20)	yes	전화번호
age	int	yes	나이

4. 레코드 관련 명령

mem 데이터베이스 테이블 생성

```
create table mem (  
  num int not null primary key,  
  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);  
  
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



The screenshot shows a MySQL command prompt window titled "관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db". The user has entered the command "mysql> desc mem;". The output displays the table structure for 'mem' with 8 rows. The columns are: num (int(11), primary key, not null), id (char(15), not null), name (char(10), not null), sex (char(1), yes), post_num (char(8), yes), address (char(80), yes), tel (char(20), yes), and age (int(11), yes). All default values are NULL.

```
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;

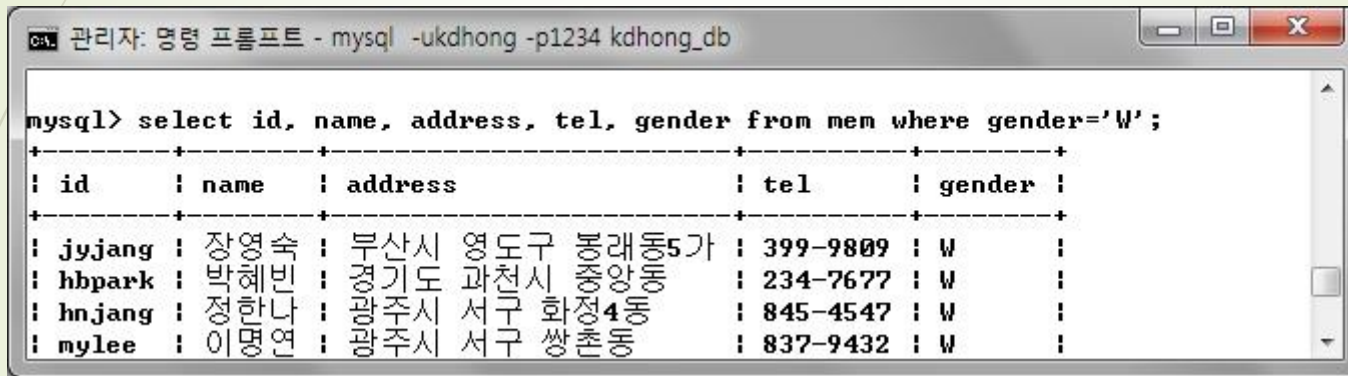
조건에 맞는 레코드 검색 명령

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,
-> sex from mem where sex = 'W';

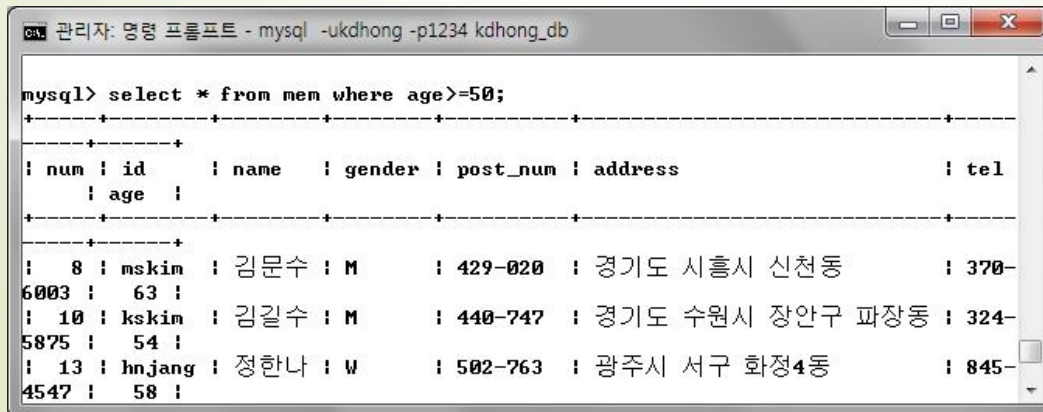


```
CA. 관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

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) 50세 이상인 레코드의 전체 필드 보기
 - mysql> select * from mem where age>=50;



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

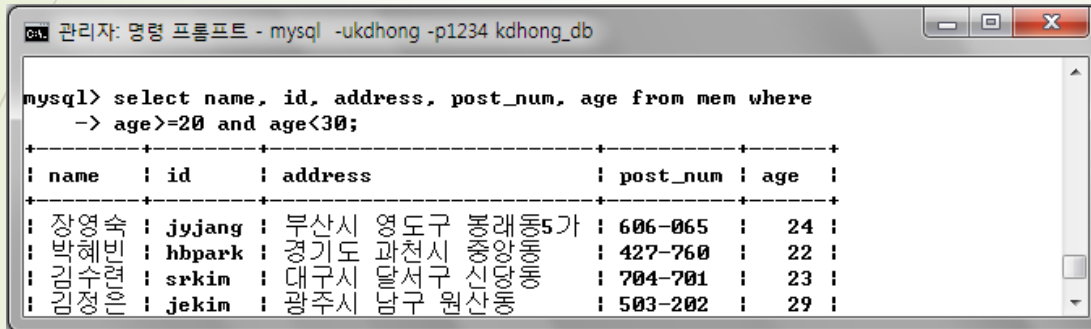
mysql> select * from mem where age>=50;
```

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;



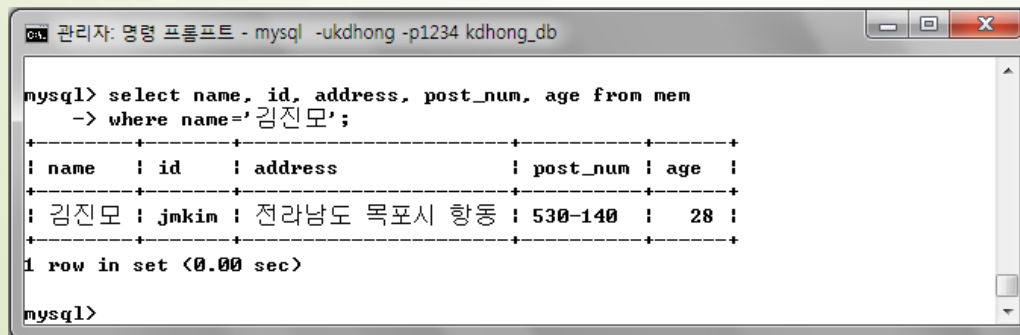
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

```
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	hhpark	경기도 과천시 중앙동	427-760	22
김수현	srkin	대구시 달서구 신당동	704-701	23
김정은	jekin	광주시 남구 원산동	503-202	29

- ex) 김진모 레코드의 아이디, 주소, 우편번호, 나이 출력

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



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 향동	530-140	28

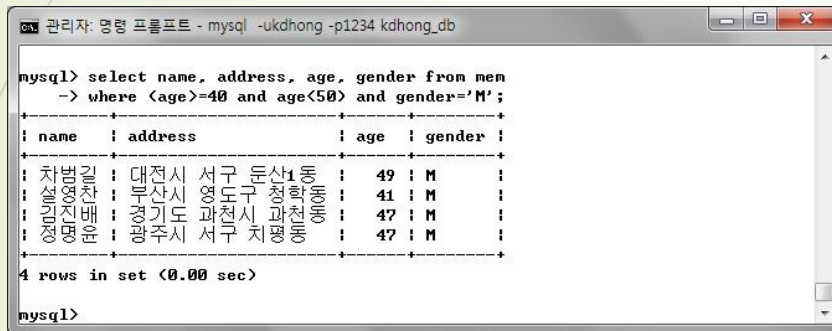
1 row in set (0.00 sec)

```
mysql>
```

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 sex='M';



관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

```
mysql> select name, address, age, gender from mem  
-> where (age>=40 and age<50) and gender='M';
```

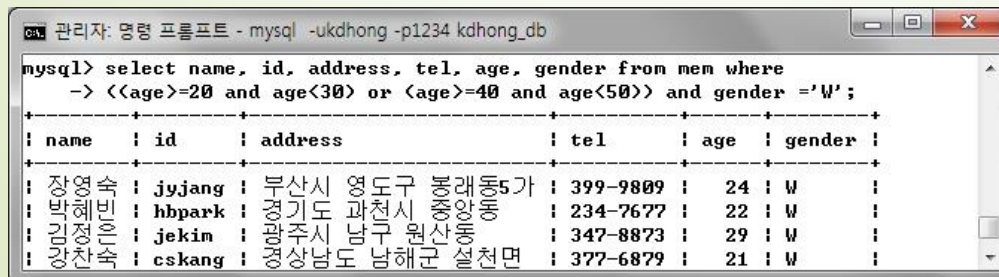
name	address	age	gender
차범길	대전시 서구 둔산1동	49	M
심영찬	부산시 영도구 청학동	41	M
김진배	경기도 과천시 과천동	47	M
정영훈	광주시 서구 치평동	47	M

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 sex='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 중앙동	234-7677	22	W
김정은	jekim	광주시 남구 원산동	347-8873	29	W
강찬숙	cskang	경상남도 남해군 설천면	377-6879	21	W

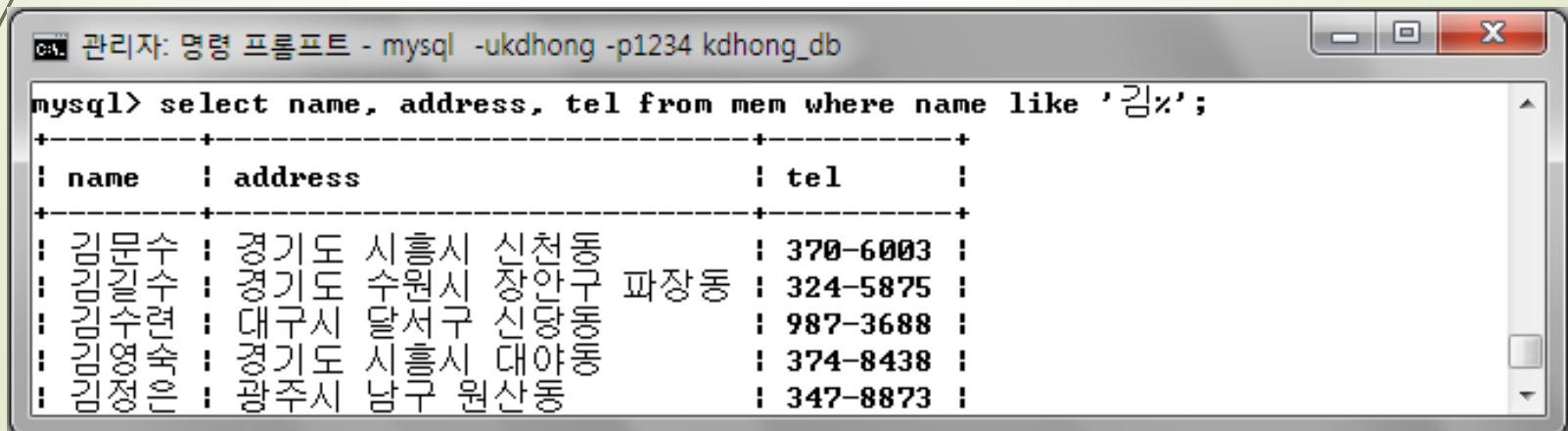
4.3 레코드 검색 명령

- 특정 문자열이 포함된 레코드 검색 명령

특정 문자열이 포함된 레코드 검색 명령

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

- 성 이 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

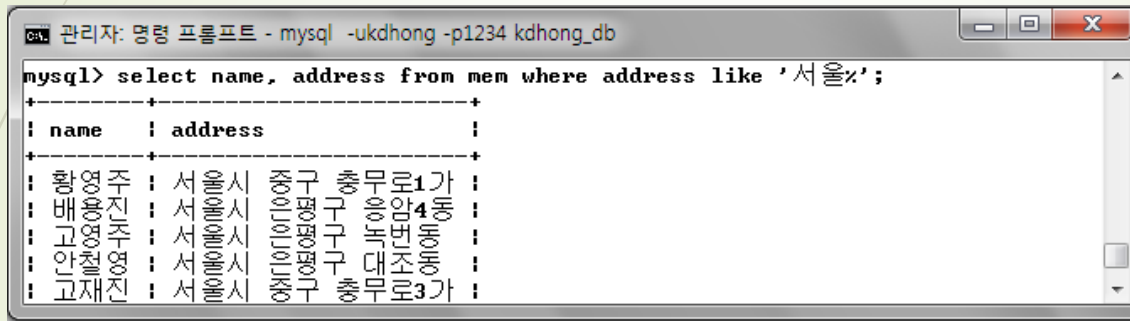
```
mysql> select name, address, tel from mem where name like '김%';
```

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%';



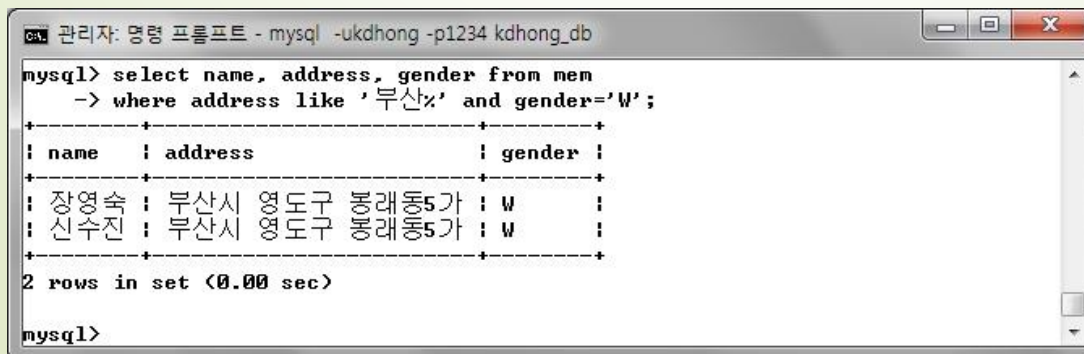
```
관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

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 여성의 이름, 주소, 성별 보기

▶ mysql> select name, address, sex from mem
-> where address like '부산%' and sex='W' ;



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

mysql> select name, address, gender from mem
-> where address like '부산%' and gender='W';

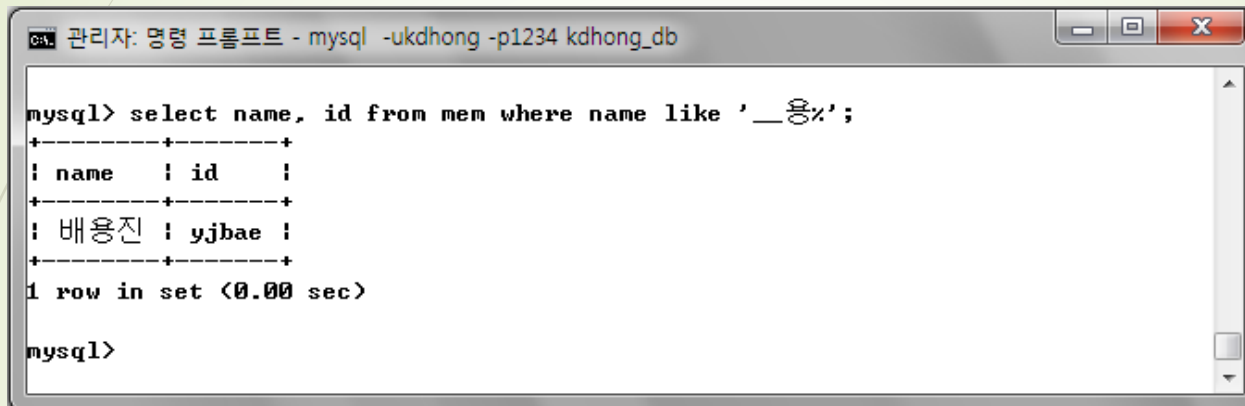
+-----+-----+-----+
| name | address | gender |
+-----+-----+-----+
| 장영숙 | 부산시 영도구 봉래동5가 | W |
| 신수진 | 부산시 영도구 봉래동5가 | W |
+-----+-----+-----+

2 rows in set (0.00 sec)

mysql>
```

4.3 레코드 검색 명령

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

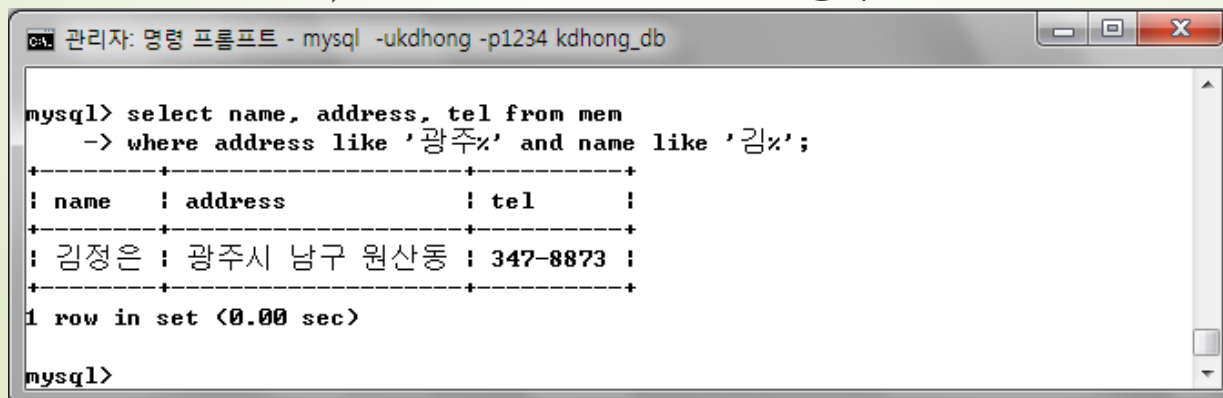


```
C:\ 관리자: 명령 프롬프트 - mysql -ukdhong -p1234 kdhong_db

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

mysql>
```

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



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

mysql> select name, address, tel from mem
-> where address like '광주%' and name like '김%';
+-----+-----+-----+
| 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, sex, 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

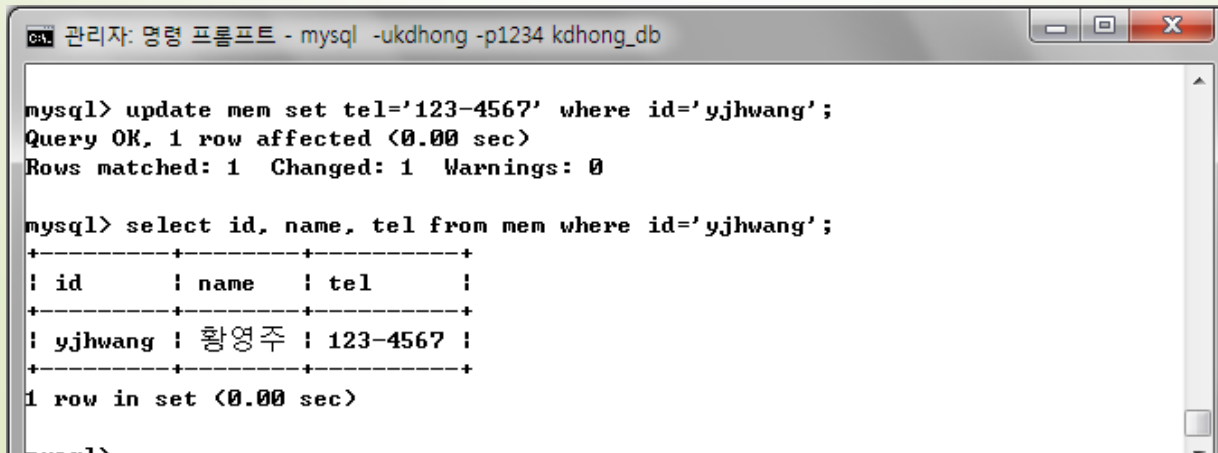
레코드 수정 명령

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';



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

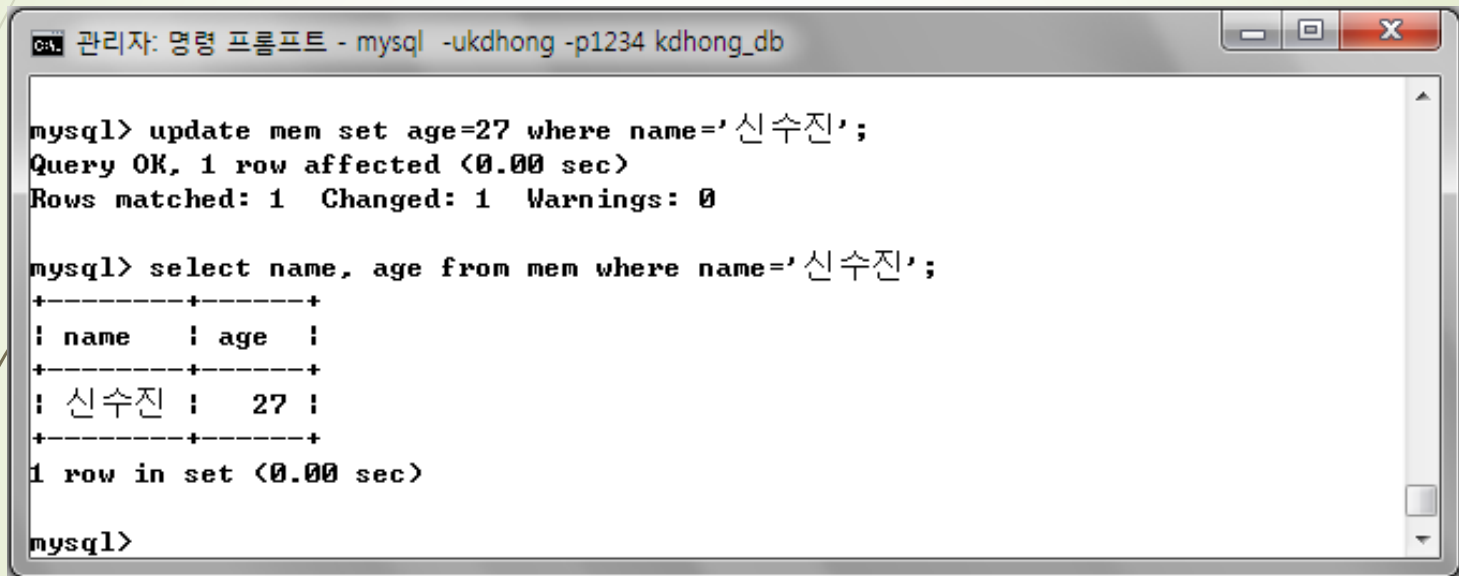
mysql> update mem set tel='123-4567' where id='yjhwang';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

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

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';



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

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>
```

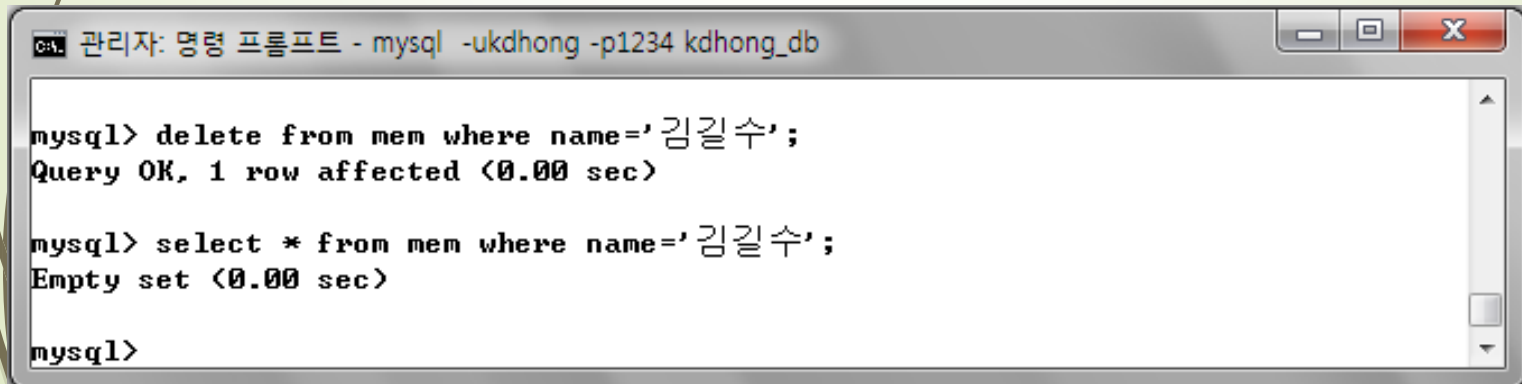
4.6 레코드 삭제 명령

조건식을 만족하는 특정 레코드 삭제 명령

```
delete from 테이블명 where 조건식
```

ex) mem 테이블에서 이름이 김길수인 레코드를 삭제

1. mysql> delete from mem where name= '김길수';
2. mysql> select * from mem where name= '김길수';



```
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)

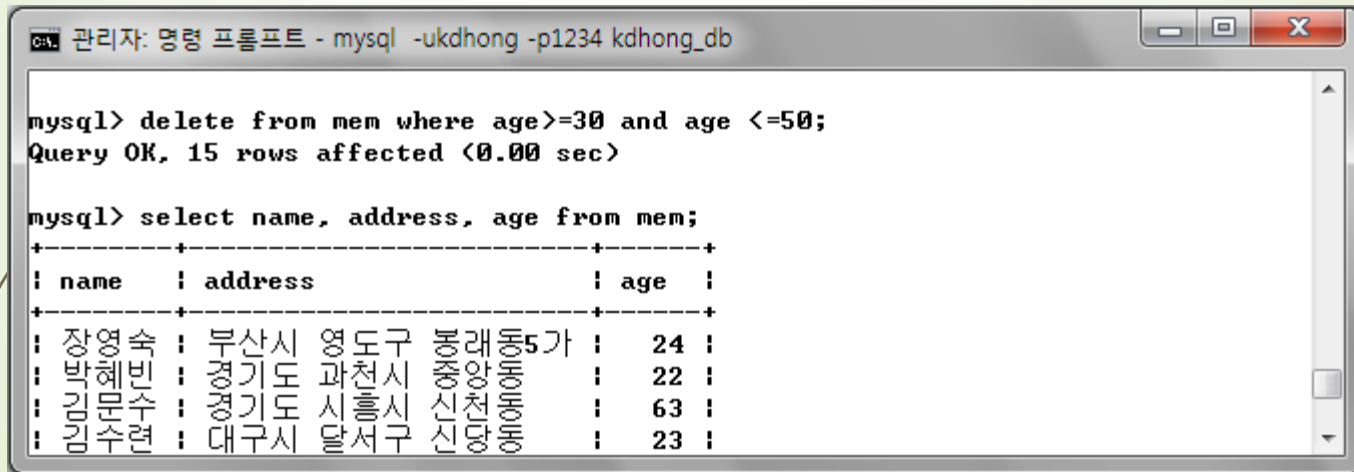
mysql>
```

4.6 레코드 삭제 명령

▶ 30~50세 레코드 삭제

▶ 나이가 30~50세인 레코드를 삭제

1. mysql> delete from mem where age>=30 and age<=50;
2. 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

5. 데이터베이스 백업 및 복원 명령

데이터베이스 백업

데이터베이스 백업 명령

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

- ▶ root 계정의 yangjung 를 yangjung.sql에 백업
 - ▶ C:\Wkdhong> mysqldump -uroot -p1234 yangjung >

```
C:\Users\kim>mysqldump -uroot -p1234 yangjung > yangjung.sql  
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

5. 데이터베이스 백업 및 복원 명령

```
mysql> create database yang;  
Query OK, 1 row affected (0.06 sec)  
  
mysql>
```

▶ 백업 파일 복원

데이터베이스 백업 파일 복원 명령

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

- ▶ kdhong_db.sql을 다른 사용자 계정에 복원
 - ▶ 계정 : root, 비밀번호 : 1234, 데이터베이스명 : yang
 - ▶ C:\₩kdhong>mysql -uroot -p1234 yang < yangjung.sql

```
C:\₩Users₩kim>mysql -uroot -p1234 yang < yangjung.sql  
mysql: [Warning] Using a password on the command line interface can be insecure.  
C:\₩Users₩kim>
```

5. 데이터베이스 백업 및 복원 명령

✓ 데이터베이스 확인

```
C:\Users\kim>mysql -uroot -p1234
mysql: [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 159
Server version: 8.0.13 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> show databases;
```

Database
information_schema
javadb
mysql
performance_schema
sakila
sampledb
sys
world
yang
yangjung
yangjung1
yangjung2

```
12 rows in set (0.01 sec)

mysql> _
```


5. 데이터베이스 백업 및 복원 명령

✓ 데이터베이스 확인

```
mysql> use yang;  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_yang |  
+-----+  
| department  
| employee  
| friend  
| gwamok  
| hakgwa  
| student  
| sungjuk  
+-----+  
7 rows in set (0.00 sec)  
  
mysql> █
```