

MySQL (DataBase)

MySQL -소개

MySQL은 무료로 사용 할 수 있어서 많은 기업에서 사용 중인 데이터베이스.
ORACLE에서 현재 소유.

일반 사용자들이 웹서비스를 이용하면서 많은 정보들이 데이터베이스에 입출력된다.

예를 들어
회원가입할 때의 개인정보, 게시글, 댓글, 상품 정보 등 많은 정보들이 데이터베이스에 저장.

개인정보, 게시글 정보, 댓글정보, 상품 정보등은 하나의 공간에 입력하지 않으며,

개인정보만의 공간, 게시글 정보의 공간, 댓글 정보의 공간 상품 정보의 공간등을 나누어
정보를 저장하며,
이러한 공간을 테이블(table)이라고 함.

다음은 테이블의 구조

번호	상품명	입고 물량	판매 수량	원산지
1	해피캣 티셔츠	200	150	미국
2	해피캣 연필	200	20	일본
3	해피캣 노트	200	30	독일

MySQL – 데이터베이스 만들기

MySQL에서 사용할 데이터베이스를 생성하는 방법

```
1 create database 데이터베이스 이름;
```

myFirstDB를 만든다고 하면 다음과 같다.

```
1 create database myFirstDB;
```

생성한 데이터베이스가 잘 생성되었는지 확인하려면 다음의 명령어로 데이터베이스 목록을 볼 수 있다.

```
1 show databases;
```

여러 데이터베이스중 자신이 사용하려는 데이터베이스를 사용하려면 use명령어를 사용한다.

```
1 use myFirstDB;
```

MySQL – 테이블

테이블은 데이터베이스에서 정보를 담는 역할을 수행.

테이블은 크게 필드와 레코드로 구분되며 다음의 그림과 같은 구조를 가진다.

번호	이름	생일	성별	필드
1	김미우	20yy-mm-dd	여	
2	김유나	20yy-mm-dd	여	레코드
3	김민후	20yy-mm-dd	남	
4	김해윤	20yy-mm-dd	여	

번호, 이름, 생일, 성별은 필드라고 하며 그외 정보는 레코드라고 한다.

MySQL – 필드의 데이터 크기와 데이터형 지정하기

테이블의 필드에는 어떠한 데이터가 입력되는지에 대한 데이터형과 그 데이터의 크기가 어느정도 되는지를 입력해야 한다.

즉 어떤 필드는 숫자만 들어갈 수 있고, 어떤 필드는 텍스트가 들어갈 수 있다. 이러한 정보를 지정하여 테이블을 최적화하는 작업이 필요하다.

다음은 숫자형 데이터에 대한 소개

데이터형	byte	저장 가능 수	양의 정수 전환 시 저장 가능 수(unsigned)
tinyint	1byte	-128 ~ 127	0 ~ 255
smallint	2byte	-32768 ~ 32767	0 ~ 65535
mediumint	3byte	-8388608 ~ 8388607	0 ~ 16777215
int 또는 integer	4byte	-2147483648 ~ 2147483647	0 ~ 4294967295
bigint	8byte	-9223372036854775808 ~ 9223372036854775807	0 ~ 18446744073709551615

위의 소개에서 unsigned는 양의 정수만을 사용할 때의 입력가능한 수를 의미한다. 즉 음의 정수는 입력하지 않으므로 그만큼의 수를 양의 정수로 사용할 수 있다.

MySQL – 필드의 데이터 크기와 데이터형 지정하기

다음은 문자형 데이터에 대한 소개

데이터형	저장 가능 글자 수
char	255
varchar	255
tinytext	255
text	65535
mediumtext	16777215
longtext	4294967295
enum	지정된 값 중 한 가지 값만 사용 가능
set	지정된 값 중 여러 가지 값을 사용 가능

장문의 글이 입력되는 게시글에는 그에 맞게끔 longtext를 사용하며, 이름이나 별명, 집 주소와 같은 정보는 char를 사용

각 필드의 용도에 맞는 데이터형을 지정한다.

MySQL – 필드의 데이터 크기와 데이터형 지정하기 - char과 varchar의 차이

char데이터형을 이용하여 네 글자까지 저장 공간을 지정했을 때 세 글자를 입력하면 한 글자의 공간이 불필요하게 남게된다.

하지만 varchar을 사용하면 나머지 남는 한 공간을 없애버려 불필요한 공간이 사라져 효율적으로 사용할 수 있으나 이러한 장점으로 인해 처리속도면에서는 char보다 느리다.

• char(4)를 지정한 경우

김	태	영	쓰지 않는 불필요 공간 발생
---	---	---	-----------------

• varchar(4)를 지정한 경우

김	태	영	쓰지 않는 공간 삭제하여 유용
---	---	---	------------------

MySQL – 필드의 데이터 크기와 데이터형 지정하기 – enum과 set의 활용방법

enum –

문자열 데이터형에서 char, varchar, tinytext, mediumtext, longtext는 입력될 정보량에 따라서 맞는 데이터형을 입력하면 된다.

그와 달리 enum은 어떤 특정 필드에 들어가는 값이 한정되어있는 경우 유용.
예를 들어 성별은 남성과 여성 뿐이므로 성별 정보를 담을 때 enum을 사용하면 데이터베이스 처리 속도면에서 더욱 유용
또한 지정한 값 이외에 다른 값은 저장되지 않으므로 원천적으로 다른 정보가 입력되지 않게끔 할 수 있다.

set –

enum은 한가지의 값만 지정이 가능한 반면 set은 지정된 값중 다수의 값을 지정할 수 있다.

예를 들어 어떤 고객이 이메일, 광고문자, 푸시알림을 모두 받고 싶지 않는다고 할 때 해당 필드의 정보로써 이 한정된 정보중에 1개만 입력하거나 2개 또는 모두를 입력할 수 있다.

MySQL – 필드의 데이터 크기와 데이터형 지정하기 – enum과 set의 활용방법

다음은 날짜형 데이터에 대한 소개

데이터베이스에 날짜를 입력할 때 사용하는 데이터형

데이터형	저장 범위	표시 형식	크기
date	1000-01-01 ~ 9999-12-31	YYYY-MM-DD	3byte
datetime	1000-01-01 00:00:00 ~ 9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS	8byte
timestamp	1970-01-01 00:00:00 ~ 2038-01-19 03:14:07	YYYYMMDDHHMMSS	4byte
time	-838:59:59 ~ 838:59:59	HH:MM:SS	3byte
year	1901~2155	YYYY	1byte

date는 년도, 월, 일만 저장하는 데이터형.

datetime는 년도, 월, 일, 시, 분, 초까지 저장하는 데이터형.

timestamp는 리눅스에서 사용하는 시간으로, 리눅스에서는 1970년 1월 1일 0시 0분 0초부터 시간을 세기 시작.

-그러므로 저장 범위가 1970년 1월 1일 0시 0분 0초부터 시작.

time은 시, 분, 초만 저장하는 데이터형.

year는 연도만 저장하는 데이터형

MySQL – 테이블 생성하기

테이블을 생성하는 명령어는 `create table` 이며 필드의 이름 데이터형 등을 지정

```
1 create table 테이블명(  
2 필드명 데이터형 필드설명  
3 )인코딩 설정 테이블 설명
```

테이블 명이 `member`, 필드명이 `name` 데이터는 한글 4글자라고 한다면 다음과 같이 생성

```
1 create table member(  
2 name char(4) comment '회원 이름'  
3 )charst=utf8 comment '고객 이름 테이블';
```

`comment` 명령어는 해당 필드 또는 테이블에 대한 설명을 입력할 때 사용하는 명령어이며 옵션이므로 작성하지 않아도 테이블 생성에 영향은 끼치지 않음.

테이블에 있는 레코드는 해당 레코드만의 고유한 번호를 가져야한다.

해당 테이블의 고유번호를 입력하는 필드에는 auto_increment를 옵션으로 입력한다.
고유번호는 자동적으로 1씩 추가된값이 입력된다.

이 고유번호는 현실세계에서 다음과 같은 역할을 한다.
학교라면 학번, 군대에서는 군번, 회사에서는 사번, 식당에서는 테이블번호

즉 회원가입 테이블에서의 고유번호는 해당 서비스의 회원번호가 된다.

다음은 auto_increment를 활용한 예제

```
1 create table member(  
2  memberID int unsigned auto_increment comment '회원번호'  
3  name char(4) comment '회원이름'  
4  )charst=utf8 comment '고객 이름 테이블';
```

memberID필드에서 회원번호를 음의정수를 사용하지 않으므로 unsigned 옵션을 사용하여 양의 정수를 사용하게 지정.

MySQL – 필드의 추가, 수정 삭제

이미 생성한 테이블에 필요에 의해 새로운 필드를 추가하거나, 수정, 삭제 할때는 ALTER 명령어를 사용.

필드를 추가하는 방법

1 ALTER TABLE 테이블명 ADD 필드명 옵션 코멘트 위치

위치에는 어떤 필드명 다음에 위치할지를 지정하며 위치를 지정하지 않을시 가장 마지막에 위치

다음은 성별 필드를 추가하는 명령문 예제

```
1 ALTER TABLE member ADD gender enum('m','w') comment '회원성별' AFTER 'name'
```

위치를 지정시에 AFTER를 작성후 앞에 위치하게 될 필드명을 입력

MySQL – 필드의 추가, 수정 삭제

필드를 수정하는 방법

```
1 ALTER TABLE 테이블명 MODIFY 변경할 필드명 옵션 코멘트 위치
```

이미 존재하는 필드를 변경할 때는 MODIFY를 사용 후 변경할 필드명을 입력

다음은 앞에서 추가한 gender필드에 남성, 여성 값 외에 아무것도 선택하지 않았다는 값 'x'를 추가하는 예제

```
1 ALTER TABLE member MODIFY gender enum('m','w','x') comment '회원성별' AFTER 'name';
```

MySQL – 필드의 추가, 수정 삭제

필드를 삭제하는 방법

```
1 ALTER TABLE 테이블명 DROP 삭제할 필드명
```

다음은 gender필드를 삭제하는 예제

```
1 ALTER TABLE member DROP gender;
```

MySQL – 테이블 삭제하기

테이블을 삭제하는 명령어는 DROP TABLE

DROP TABLE 명령어 사용방법

```
1 DROP TABLE 테이블명;
```

다음은 member 테이블을 삭제하는 예제

```
1 DROP TABLE member;
```

MySQL – 테이블에 데이터 입력하기

테이블에 데이터를 입력하는 명령문은 INSERT INTO

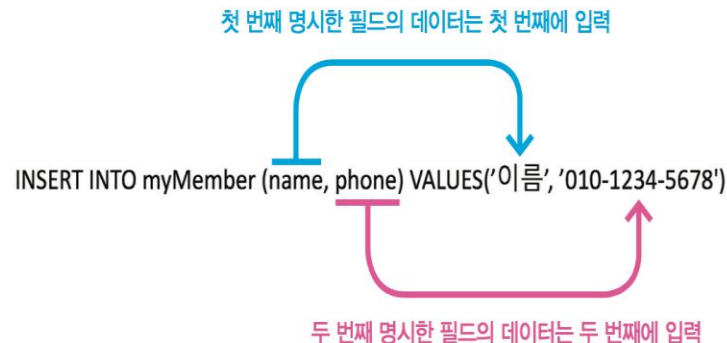
INSERT INTO 명령어 사용방법

1 INSERT INTO 테이블명(입력할 필드명) VALUES(입력할 데이터);

데이터를 입력할 필드가 name과 phone이고, 입력할 데이터가 '김태영', '010-1234-5678'일 때
데이터를 입력하는 예제는 다음과 같다.

1 INSERT INTO member(name, phone) VALUES('김태영', '010-1234-5678');

필드명의 순서에 맞게 데이터의 순서도 맞춰준다.



MySQL – 테이블에 데이터 입력하기

▶ 입력할 데이터 정보

아이디: everdevelHost

이름: 김태영

패스워드: mm281118

휴대전화 번호: 010-1234-5678

이메일: everdevel@everdevel.com

생일: 1986-04-04

성별: 남성

[코드 6-11] 레코드 입력

● 9-1_insert.sql

```
1: INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender,  
2: regTime)  
3: VALUES('everdevelHost', '김태영', 'mm281118', '010-1234-5678', 'everdevel@everdevel.com',  
4: '1986-04-04', 'm', now( ));
```

MymemberID 항목은 입력하지 않았다. 왜??

now() 함수는 현재 날짜..

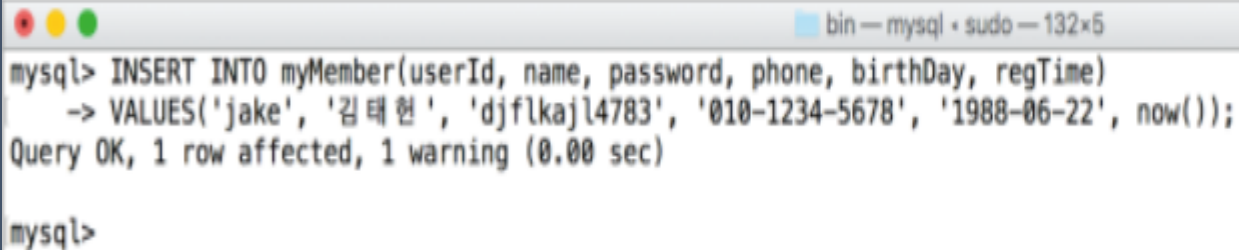
MySQL – 테이블에 데이터 입력하기

email과 gender 값 입력 않고 새로운 레코드 생성

[코드 6-12] 레코드 입력

● 9-2_insert.sql

```
1: INSERT INTO myMember(userId, name, password, phone, birthDay, regTime)
2: VALUES('jake', '김태현', 'djflkajl4783', '010-1234-5678', '1988-06-22', now( ));
```

A terminal window titled 'bin — mysql • sudo — 132x5' showing the execution of an SQL INSERT statement. The command is entered as 'mysql> INSERT INTO myMember(userId, name, password, phone, birthDay, regTime) -> VALUES('jake', '김태현', 'djflkajl4783', '010-1234-5678', '1988-06-22', now());'. The output shows 'Query OK, 1 row affected, 1 warning (0.00 sec)' followed by a new 'mysql>' prompt.

```
mysql> INSERT INTO myMember(userId, name, password, phone, birthDay, regTime)
-> VALUES('jake', '김태현', 'djflkajl4783', '010-1234-5678', '1988-06-22', now());
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql>
```

1 warning 이 나타난 이유??

몇 가지 레코드 더 입력

[코드 6-13] 레코드 입력

● 9-3_insert.sql

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES('haeyun', '이지형', 'dlwlgud85', '010-1234-5678', 'tigger@everdevel.com', '1985-11-18', 'm',
now( ));
```

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES('nelly', '전윤동', 'wjsdbsehd83', '010-1234-5678', 'nelly@everdevel.com', '1983-04-27', 'm',
now( ));
```

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES('cooper', '노진우', 'shwlsdn87', '010-1234-5678', 'cooper@everdevel.com', '1987-02-01', 'm',
now( ));
```

MySQL – 데이터 불러오기

테이블의 데이터를 불러오는 명령문은 SELECT

SELECT 명령어 사용방법

1 SELECT 불러올 필드명 FROM 테이블명

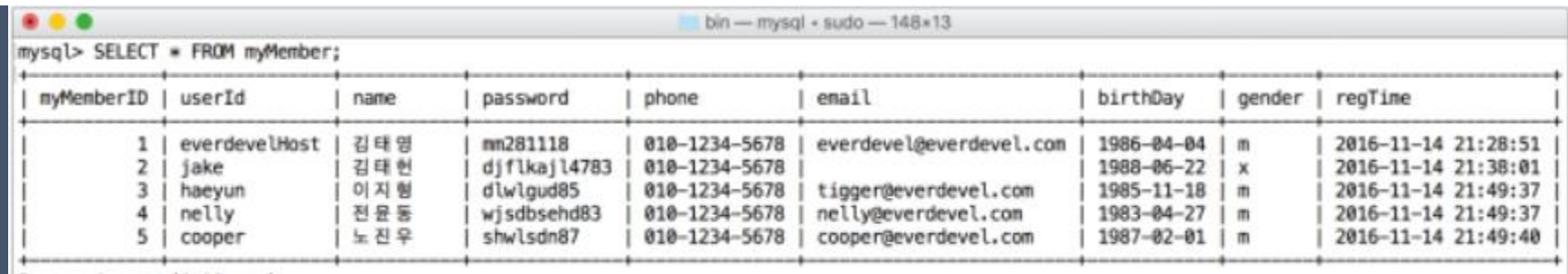
다음은 테이블명이 member이고 불러올 필드명이 name, gender 일 때의 예제

1 SELECT name, gender FROM member

테이블안의 모든 필드의 정보를 불러온다면 불러올 필드명을 적을 자리에 *를 입력.
*는 모든것을 의미.

다음은 테이블명이 member이고 모든 필드의 데이터를 불러올 때의 예제

1 SELECT * FROM member



```
mysql> SELECT * FROM myMember;
```

myMemberID	userId	name	password	phone	email	birthDay	gender	regTime
1	everdevelHost	김태영	mn281118	010-1234-5678	everdevel@everdevel.com	1986-04-04	m	2016-11-14 21:28:51
2	jake	김태현	djflkajl4783	010-1234-5678		1988-06-22	x	2016-11-14 21:38:01
3	haeyun	이지형	dlwlgud85	010-1234-5678	tigger@everdevel.com	1985-11-18	m	2016-11-14 21:49:37
4	nelly	전윤동	wjsdbsehd83	010-1234-5678	nelly@everdevel.com	1983-04-27	m	2016-11-14 21:49:37
5	cooper	노진우	shwlsdn87	010-1234-5678	cooper@everdevel.com	1987-02-01	m	2016-11-14 21:49:40

MySQL – 데이터 불러오기 – 조건에 따른 데이터 불러오기

테이블의 데이터를 불러올 때 특정 조건의 데이터를 불러오려면 조건문인 WHERE을 사용

SELECT 명령어에 WHERE문 사용방법

1 SELECT 불러올 필드명 FROM 테이블명 WHERE 조건

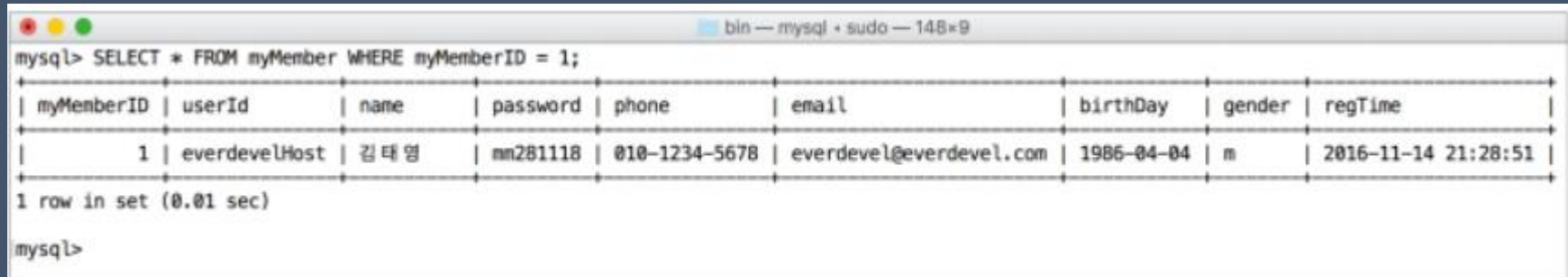
WHERE문에서 사용되는
조건식

기호	뜻
=	같다
!= 또는 <>	같지 않다
>=	크거나 같다
<=	작거나 같다
>	크다
<	작다

MySQL – 데이터 불러오기 – 조건에 따른 데이터 불러오기

테이블명이 myMember이고 아이디가 1인 사람의 모든 필드 검색

```
1 SELECT * FROM myMember WHERE myMemberID=1;
```



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

```
mysql> SELECT * FROM myMember WHERE myMemberID = 1;
```

myMemberID	userId	name	password	phone	email	birthDay	gender	regTime
1	everdevelHost	김태영	nn281118	010-1234-5678	everdevel@everdevel.com	1986-04-04	m	2016-11-14 21:28:51

```
1 row in set (0.01 sec)
```

```
mysql>
```

테이블명이 myMember이고 성별이 남성인 사람의 이름을 불러오는 예제

```
1 SELECT name FROM myMember WHERE gender = 'm'
```

테이블명이 myMember이고 회원번호가 100보다 큰 사람의 데이터를 불러오는 예제

```
1 SELECT * FROM myMember WHERE myMemberID > 100
```

테이블명이 myMember이고 이름이 김태영인 사람의 데이터를 불러오는 예제

```
1 SELECT * FROM myMember WHERE name = '김태영'
```

MySQL – 데이터 불러오기 – 조건에 따른 데이터 불러오기

Like - 부분 일치하는 값을 찾을 때는 LIKE를 사용

% - 모든 값을 나타냄

ex: 김% - 김으로 시작하는 모든 데이터, %김% - 김이 들어 있는 모든 것, %김 - 김으로 끝나는 모든 것

테이블명이 myMember이고 name필드의 값이 김으로 시작하는 사람의 데이터를 불러오는 예제

```
1 SELECT * FROM myMember WHERE name LIKE '김%'
```

이메일에 everdevel라는 텍스트가 있고 그 텍스트 앞에 어떠한 텍스트가 있고, 뒤에는 아무것도 없는 데이터를 찾는다면 앞에만 %를 붙여줍니다.

```
SELECT * FROM myMember WHERE email LIKE '%everdevel';
```

LIKE문을 실행한 후 그 결과를 살펴보겠습니다.

[코드 6-17] 데이터 조건에 따라 불러오기

● 10-4_select.sql

```
SELECT * FROM myMember WHERE email LIKE 'n%';
```

MySQL – 데이터 불러오기 – 조건에 따른 데이터 불러오기

2개 이상의 조건을 사용하고 두 조건이 모두 참일때의 조건에는 AND를 사용.
2개 이상의 조건을 사용하고 두 조건이중 하나라도 참일때의 조건에는 OR를 사용.

name 필드의 값이 김으로 시작하고 남자인 조건식을 충족하는 예제

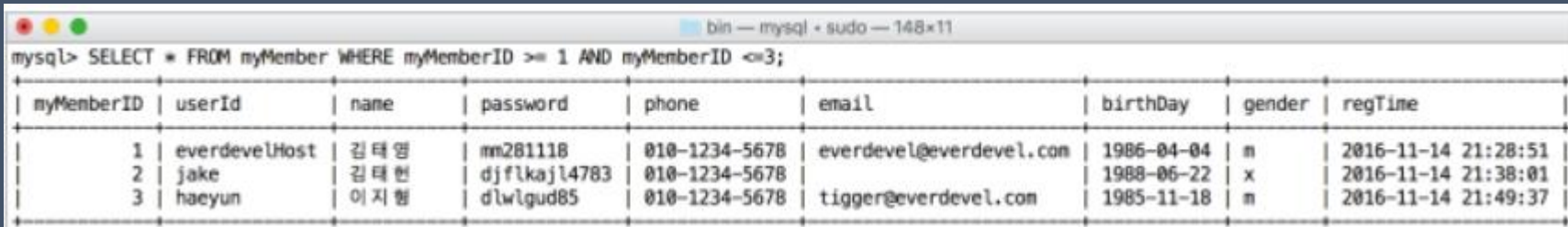
```
1 SELECT * FROM myMember WHERE name LIKE '김%' AND gender = 'm'
```

테이블명이 myMember이고 name필드의 값이 김으로 시작하거나 남자인 사람의 데이터를 불러오는 예제

```
1 SELECT * FROM myMember WHERE name LIKE '김%' OR gender = 'm'
```

테이블명이 myMember이고 ID가 1보다 크고 3보다 작은 식을 충족하는 예제

```
SELECT * FROM myMember WHERE myMemberID >= 1 AND myMemberID <=3;
```



The screenshot shows a terminal window with the title 'bin — mysql • sudo — 148x11'. The command entered is 'mysql> SELECT * FROM myMember WHERE myMemberID >= 1 AND myMemberID <=3;'. The output is a table with 9 columns: myMemberID, userId, name, password, phone, email, birthDay, gender, and regTime. There are 3 rows of data.

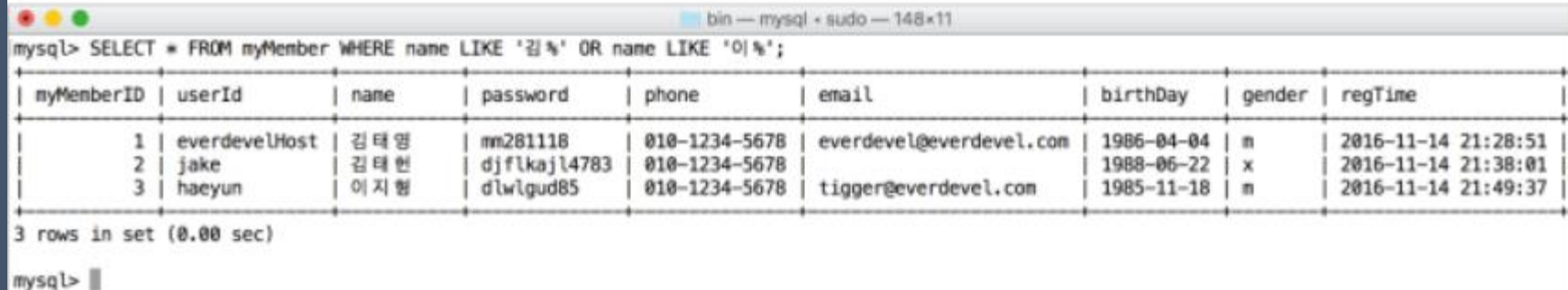
myMemberID	userId	name	password	phone	email	birthDay	gender	regTime
1	everdevelHost	김태영	mm281118	010-1234-5678	everdevel@everdevel.com	1986-04-04	m	2016-11-14 21:28:51
2	jake	김태현	djflkajl4783	010-1234-5678		1988-06-22	x	2016-11-14 21:38:01
3	haeyun	이지형	dwlqgud85	010-1234-5678	tigger@everdevel.com	1985-11-18	m	2016-11-14 21:49:37

MySQL – 데이터 불러오기 – 조건에 따른 데이터 불러오기

2개 이상의 조건을 사용하고 두 조건이 모두 참일때의 조건에는 AND를 사용.
2개 이상의 조건을 사용하고 두 조건이중 하나라도 참일때의 조건에는 OR를 사용.

name 필드의 값이 '김' 또는 '이'로 시작하는 조건식식을 충족하는 예제

```
SELECT * FROM myMember WHERE name LIKE '김%' OR name LIKE '이%';
```



The screenshot shows a terminal window titled 'bin -- mysql - sudo -- 148x11'. The prompt is 'mysql>'. The query entered is 'SELECT * FROM myMember WHERE name LIKE '김%' OR name LIKE '이%';'. The results are displayed in a table with 9 columns: myMemberID, userId, name, password, phone, email, birthDay, gender, and regTime. There are 3 rows of data. Below the table, it says '3 rows in set (0.00 sec)'. The prompt 'mysql>' is visible at the bottom.

myMemberID	userId	name	password	phone	email	birthDay	gender	regTime
1	everdevelHost	김태영	mm281118	010-1234-5678	everdevel@everdevel.com	1986-04-04	m	2016-11-14 21:28:51
2	jake	김태현	djflkajl4783	010-1234-5678		1988-06-22	x	2016-11-14 21:38:01
3	haeyun	이지형	dlwlgud85	010-1234-5678	tigger@everdevel.com	1985-11-18	m	2016-11-14 21:49:37

MySQL – 데이터의 값을 변경, 삭제하기

- 데이터 변경

데이터(레코드)의 값을 변경하기 위해 사용하는 명령문은 UPDATE

UPDATE 명령어 사용방법

1 UPDATE 테이블명 SET 필드명 값 조건

UPDATE문을 사용할 때 조건을 사용하지 않을 경우 모든 레코드의 값이 변경되므로 주의하여 사용

다음은 myMemberID가 5번인 고객의 휴대전화 번호를 0으로 변경하는 예제입니다.

[코드 6-21] 조건에 따른 데이터값 변경

● 11-1_update.sql

```
UPDATE myMember SET phone = 0 WHERE myMemberID = 5;
```

[코드 6-21]은 myMember 테이블의 phone 필드의 값을 0으로 변경하고, WHERE문을 사용하여 myMemberID가 5인 회원의 값을 변경하도록 하는 쿼리문입니다.

```
bin — mysql — sudo — 148x13
mysql> UPDATE myMember SET phone = '0', userId = 'cooper' WHERE myMemberID = 5;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM myMember WHERE myMemberID = 5;
```

myMemberID	userId	name	password	phone	email	birthDay	gender	regTime
5	cooper	노진우	shwlsdn87	0	cooper@everdevel.com	1987-02-01	m	2016-11-14 21:49:40

MySQL – 데이터의 값을 변경, 삭제하기

- 데이터 변경

1 UPDATE 테이블명 SET 필드명 값 조건

myMember 테이블에서 멤버 아이디가 5인 사람의 전화번호와 사용자 아이디를 아래와 같이 바꾼다.

```
UPDATE myMember SET phone = '010-1234-5678', userId = 'sheldon' WHERE myMemberID = 5;
```

필드에 값을 대입한 후 ',' 로 구분하여 다음 변경할 필드를 적고 값을 대입합니다.

bin — mysql • sudo — 148x13

```
mysql> UPDATE myMember SET phone = '010-1234-5678', userId = 'sheldon' WHERE myMemberID = 5;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> SELECT * FROM myMember WHERE myMemberID = 5;
```

myMemberID	userId	name	password	phone	email	birthDay	gender	regTime
5	sheldon	노진우	shwlsdn87	010-1234-5678	cooper@everdevel.com	1987-02-01	m	2016-11-14 21:49:40

myMember 테이블에서 모든 회원의 전화번호 변경

```
UPDATE myMember SET phone = '010-1234-5678';
```

MySQL – 데이터의 값을 변경, 삭제하기

- 데이터 삭제

데이터(레코드)의 값을 삭제하기 위해 사용하는 명령문은 DELETE

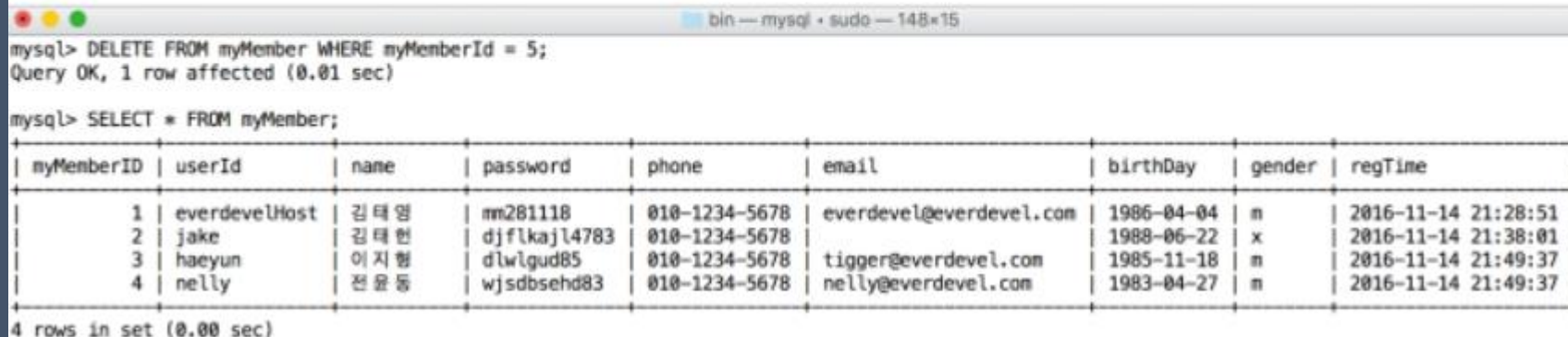
DELETE 명령어 사용방법

1 DELETE FROM 테이블명 조건

5번 회원의 레코드 삭제

```
DELETE FROM myMember WHERE myMemberId = 5;
```

[코드 6-26]에 WHERE문을 작성하지 않으면 테이블의 모든 레코드가 삭제되므로 조심해야 합니다.



```
bin — mysql • sudo — 148x15
mysql> DELETE FROM myMember WHERE myMemberId = 5;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM myMember;
```

myMemberID	userId	name	password	phone	email	birthDay	gender	regTime
1	everdevelHost	김태영	mm281118	010-1234-5678	everdevel@everdevel.com	1986-04-04	m	2016-11-14 21:28:51
2	jake	김태현	djflkajl4783	010-1234-5678		1988-06-22	x	2016-11-14 21:38:01
3	haeyun	이지형	dlwlgud85	010-1234-5678	tigger@everdevel.com	1985-11-18	m	2016-11-14 21:49:37
4	nelly	전윤동	wjsdbsehd83	010-1234-5678	nelly@everdevel.com	1983-04-27	m	2016-11-14 21:49:37

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

- 데이터 삭제

1 DELETE FROM 테이블명 조건

[코드 6-27] 데이터 삭제

● 11-7_deleteIN.sql

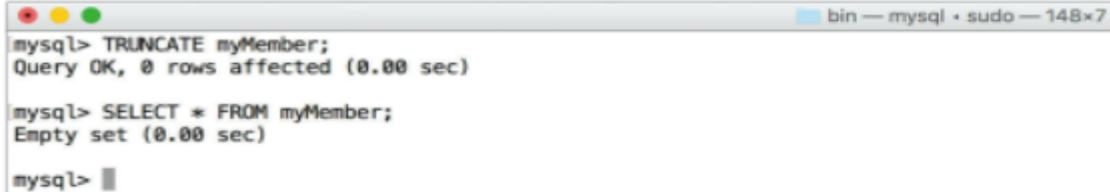
```
DELETE FROM myMember WHERE myMemberId IN(1,2,3);
```

[코드 6-27]에서는 IN절을 사용했습니다. myMemberId IN(1,2,3)은 myMemberID가 1, 2, 3인 레코드를 의미합니다. 그러므로 myMemberID가 1,2,3인 레코드를 삭제하는 쿼리문이 됩니다.

테이블을 처음만든 상태로 만드려면 TRUNCATE 명령문 사용

1 TRUNCATE 테이블명

```
TRUNCATE myMember;
```



The screenshot shows a terminal window titled "bin — mysql • sudo — 148x7". The terminal output is as follows:

```
mysql> TRUNCATE myMember;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM myMember;
Empty set (0.00 sec)

mysql> █
```

Join을 위해 데이터 입력

[코드 6-29] 데이터 입력

● 12-2_insert.sql

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES('everdevelHost', '김태영', 'mm281118', '010-1234-5678', 'everdevel@everdevel.com', '1928-11-18', 'm', now( ));
```

```
INSERT INTO myMember(userId, name, password, phone, birthDay, regTime)
VALUES('dreamLights', '김태현', 'djflkajl4783', '010-1234-5678', '1940-11-18', now( ));
```

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES('tigger', '이지형', 'dlwlgud85', '010-1234-5678', 'tigger@everdevel.com', '1985-11-18', 'm', now( ));
```

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES('nelly', '전윤동', 'wjsdbsehd83', '010-1234-5678', 'nelly@everdevel.com', '1983-04-27', 'm', now( ));
```

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES('cooper', '노진우', 'shwlsdn87', '010-1234-5678', 'cooper@everdevel.com', '1987-02-01', 'm', now( ));
```


Join을 위해 데이터 입력

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES(linu, '백창인', 'gorckddls81', '010-1234-5678', 'linu@everdevel.com', '1981-12-25', 'm',
now( ));
```

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES('hanje', '한정은', 'gkswjddms88', '010-1234-5678', 'hanje@everdevel.com', '1988-04-08', 'w',
now( ));
```

```
INSERT INTO myMember(userId, name, password, phone, email, birthDay, gender, regTime)
VALUES('stella', '이정미', 'dlwjdal84', '010-1234-5678', 'stella@everdevel.com', '1984-06-26', 'w',
now( ));
```

데이터 검색 결과

```
mysql> SELECT * FROM myMember;
```

myMemberID	userId	name	password	phone	email	birthDay	gender	regTime
1	everdevelHost	김태영	mn281118	010-1234-5678	everdevel@everdevel.com	1928-11-18	m	2016-11-15 11:28:02
2	dreamLights	김태현	djflkajl4783	010-1234-5678		1940-11-18	x	2016-11-15 11:28:02
3	tigger	이지형	dlwlgud85	010-1234-5678	tigger@everdevel.com	1985-11-18	m	2016-11-15 11:28:02
4	nelly	전윤동	wjsdbsehd83	010-1234-5678	nelly@everdevel.com	1983-04-27	m	2016-11-15 11:28:02
5	cooper	노진우	shwlsdn87	010-1234-5678	cooper@everdevel.com	1987-02-01	m	2016-11-15 11:28:02
6	hanje	한정은	gkswjddms88	010-1234-5678	hanje@everdevel.com	1988-04-08	w	2016-11-15 11:28:03
7	stella	이정미	dlwjdal84	010-1234-5678	stella@everdevel.com	1984-06-26	w	2016-11-15 11:28:05

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