

ReferenceBook

My SQL



Author Name
KIMJUNGHO



MySQL이란?

MySQL은 전세계적으로 가장 널리 사용되고 있는 오픈 소스 데이터베이스이다.

- 매우 빠르고, 유현하며, 사용하기 쉬운 특징이 있다.
- 다중사용자, 다중 쓰레드를 지원하고, 응용프로그램 인터페이스(API)를 제공한다.
- 유닉스나 리눅스, Windows 운영체제 등에서 사용할 수 있다.

사용 이유



- 1 데이터 저장 및, 입출력 관리가 효율적이다.
- 2 인게임 내부 데이터를 유연하게 수정이 가능하다.
- 3 데이터 스토리지 분리로 운영 안정성 확보가 크다.





데이터 베이스(Database)

database는 서로 관련된 table들을 모아놓은 폴더

- 특징 !

실시간 접근과 삽입, 삭제 갱신을 통해서 현재의 데이터를 동적으로 유지할 수 있는 계속적인 변화를 제공

- database 생성 순서

```
> mysql -uroot -p
```

: 루트 권한으로 로그인을 하고, 데이터베이스를 확인합니다.



```
mysql> show DATABASES;
```

: sql이라는 언어를 사용하고 있고, 명령의 끝을 ;로 구분하기 때문에, 반드시 ;를 입력해주어야합니다.



〈 B 〉

테이블(table) 만들기

CREATE TABLE

```
CREATE TABLE myMember ( (0)
myMemberID int(10) unsigned auto_increment comment "회원의 고유번호", (1)

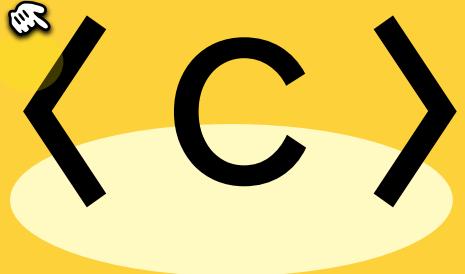
youEmail varchar(40) NOT NULL comment "회원 아이디", (2)
youName varchar(12) NOT NULL comment "회원 이름",
youPass varchar(20) NOT NULL comment "회원 비밀번호",
youBirth varchar(11) NOT NULL comment "회원 생일", (3)
regTime int(11) NOT NULL comment "회원 가입 시간",
PRIMARY KEY (myMemberID)) (4)
charset=utf8 comment ="회원 정보 테이블"; (5)
```

```
show tables;
+-----+
| Tables_in_test02 |
+-----+
| mymember         |
+-----+
1 row in set (0.00 sec)
```

SHOW TABLES

- (0) : 테이블을 생성하는 명령어입니다.
- (1) : myMemberID 라는 컬럼을 추가하는 데, INT 타입으로 지정합니다.
- (2) : 가변길이로 40글자까지 허용합니다.(20글자가 넘어가면 40글자에서 자릅니다.)
- (3) : NOT NULL은 **값을 비워둘 수 없음**을 의미합니다.
- (4) : PRIMARY KEY를 id 컬럼으로 지정하겠다는 의미
- (5) : utf-8은 모든 언어를 표기할 수 있는 세계표준 코드





테이블(table) 수정하기

● 테이블 보기 (DESC 테이블 이름)

```
DESC myMember;
```

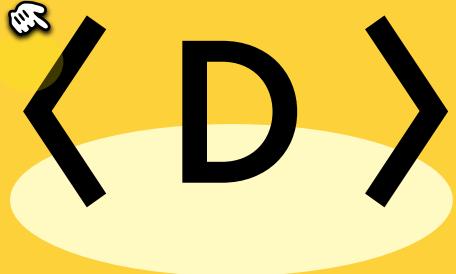
: 앞서 설정했던 테이블 칼럼들과 옵션을 확인할 수 있습니다.

```
mysql> desc myMember;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra       |
+-----+-----+-----+-----+-----+
| myMemberID | int(10) unsigned | NO | PRI | NULL    | auto_increment |
| youEmail   | varchar(40)    | NO |     | NULL    |             |
| youName    | varchar(12)    | NO |     | NULL    |             |
| youPass    | varchar(20)    | NO |     | NULL    |             |
| youBirth   | varchar(11)    | NO |     | NULL    |             |
| regTime    | int(11)       | NO |     | NULL    |             |
+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

● 테이블 삭제 (DROP TABLE 테이블 이름)

```
mysql> drop table myMember;
Query OK, 0 rows affected (0.01 sec)
```





테이블(table) 필드 추가 / 수정 / 삭제

필드 추가하기

ALTER TABLE 테이블명 ADD 추가할 필드명 AFTER 필드 위치

```
mysql> ALTER TABLE myMember2 ADD youAge int(3) not null comment  
"회원의 나이" AFTER youBirth; Query OK, 0 rows affected (0.05 sec)
```

필드 수정하기

ALTER TABLE 테이블명 MODIFY 수정할 필드명

```
mysql> ALTER TABLE myMember2 MODIFY youGender enum('m','w');  
Query OK, 0 rows affected (0.04 sec)
```

필드 삭제하기

ALTER TABLE 테이블명 DROP 삭제할 필드명

```
mysql> ALTER TABLE myMember2 DROP youAge;  
Query OK, 0 rows affected (0.04 sec)
```





테이블 데이터(table data)

데이터 입력하기

INSERT INTO 테이블 이름(필드명) VALUES(데이터);

```
INSERT INTO mymember(youEmail, youName, youPass, youBirth, regTime)
VALUES(□@□.□, □, □, □-□-□, now());
```

데이터 불러오기

SELECT * FROM 테이블명 WHERE 조건

```
MariaDB [test01] > SELECT * FROM mymember;
```

myMemberID	youEmail	youName	youPass	youBirth	regTime
1	dkdlwo@naver.com	jungho	1234	1994-06-28	2147483647





테이블 데이터(table data) : 불러오기(1)

데이터 불러오기

SELECT * FROM 테이블명 WHERE 조건

● 전체 데이터 불러오기

```
SELECT * FROM mymember;
```

```
MariaDB [test01] > SELECT * FROM mymember;
```

myMemberID	youEmail	youName	youPass	youBirth	regTime
1	dkdlwo@naver.com	jungho	1234	1994-06-28	2147483647
2	wow2002r@naver.com	Ripia	1234	1994-11-04	333
3	guswjd8694@naver	dallae	1234	2016-05-25	333

```
3 rows in set (0.000 sec)
```





테이블 데이터(table data) : 불러오기(2)

데이터 불러오기

SELECT * FROM 테이블명 WHERE 조건

ex) myMemberID가 6번인 경우

```
SELECT * FROM mymember WHERE myMemberID = □;
```

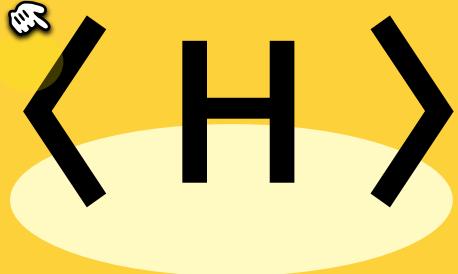
```
SELECT * FROM mymember WHERE myMemberID = 6;
```

```
MariaDB [test01] > SELECT * FROM mymember WHERE myMemberID = 6;
```

```
+-----+-----+-----+-----+-----+
| myMemberID | youEmail      | youName    | youPass    | youBirth   | regTime   |
+-----+-----+-----+-----+-----+
| 6       | seonyoung@gmail.com | SEONYOUNG | 1234      | 1997-04-03 | 111      |
+-----+-----+-----+-----+-----+
```

```
1 row in set (0.006 sec)
```





테이블 데이터(table data) : 불러오기(3)

데이터 불러오기

SELECT * FROM 테이블명 WHERE 조건

ex) email 중에 naver 텍스트를 포함하는 경우

```
SELECT * FROM mymember WHERE myMemberID = □;
```

```
SELECT * FROM mymember WHERE youEmail LIKE '%naver%';
```

```
MariaDB [test01] > SELECT * FROM mymember WHERE youEmail LIKE '%naver%';
```

myMemberID	youEmail	youName	youPass	youBirth	regTime
1	dkdlwo@naver.com	jungho	1234	1994-06-28	2147483647
2	wow2002r@naver.com	Ripia	1234	1994-11-04	333
3	guswjd8694@naver	dallae	1234	2016-05-25	333
7	akside111@naver	akside	1234	2000-01-01	123

```
4 rows in set (0.001 sec)
```





테이블 데이터(table data) : 불러오기(4)

데이터 불러오기

SELECT * FROM 테이블명 WHERE 조건

ex) email 중에 akside 텍스트로 시작하는 경우

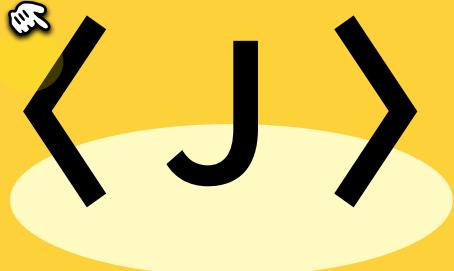
```
SELECT * FROM mymember WHERE youEmail LIKE '%akside%';
```

```
SELECT * FROM mymember WHERE youEmail LIKE '%akside%';
MariaDB [test01] > SELECT * FROM mymember WHERE youEmail LIKE '%akside%';
```

```
+-----+-----+-----+-----+
| myMemberID | youEmail      | youName | youPass | youBirth | regTime |
+-----+-----+-----+-----+
|      7     | akside111@naver | akside  | 1234   | 2000-01-01 | 123    |
+-----+-----+-----+-----+
```

```
1 row in set (0.000 sec)
```





테이블 데이터(table data) : 불러오기(5)

데이터 불러오기

SELECT * FROM 테이블명 WHERE 조건

ex) email 중에 com 텍스트로 끝나는 경우

```
SELECT * FROM mymember WHERE youEmail LIKE '%com';
```

```
SELECT * FROM mymember WHERE youEmail LIKE '%com';
```

```
MariaDB [test01] > SELECT * FROM mymember WHERE youEmail LIKE '%com';
```

myMemberID	youEmail	youName	youPass	youBirth	regTime
1	dkdlwo@naver.com	jungho	1234	1994-06-28	2147483647
2	wow2002r@naver.com	Ripia	1234	1994-11-04	333
4	younyc0413@gmail.com	youn	0413	1992-04-13	333
6	seonyoung@gmail.com	SEONYOUNG	1234	1997-04-03	111





테이블 데이터(table data) : 불러오기(6)

데이터 불러오기

SELECT * FROM 테이블명 WHERE 조건

ex) email 중에 자기 이름이 텍스트로 끝나는 경우

```
SELECT * FROM mymember WHERE youName LIKE '%□□□□□%';
```

```
SELECT * FROM mymember WHERE youName LIKE '%jungho%';
```

```
MariaDB [test01] > SELECT * FROM mymember WHERE youName LIKE '%jungho%';
```

```
+-----+-----+-----+-----+-----+
| myMemberID | youEmail      | youName | youPass | youBirth | regTime   |
+-----+-----+-----+-----+-----+
|      1 | dkdlwo@naver.com | jungho | 1234    | 1994-06-28 | 2147483647 |
+-----+-----+-----+-----+-----+
```

```
1 row in set (0.000 sec)
```





테이블 데이터(table data) : 불러오기(7)

데이터 불러오기

SELECT * FROM 테이블명 WHERE 조건

ex) 멤버 아이디 조건 검색 (멤버 ID가 2 ~ 5인 경우를 검색)

```
SELECT * FROM mymember WHERE myMemberID >= □ AND myMemberID <=□;
```

```
MariaDB [test01] > SELECT * FROM mymember WHERE myMemberID >= 2  
AND myMemberID <= 5;
```

myMemberID	youEmail	youName	youPass	youBirth	regTime
2	wow2002r@naver.com	Ripia	1234	1994-11-04	333
3	guswd8694@naver	dallae	1234	2016-05-25	333
4	younyc0413@gmail.com	youn	0413	1992-04-13	333
5	billiejeong08@gmail	billie	1234	1995-08-12	333

4 rows in set (0.005 sec)





테이블 데이터(table data) : 불러오기(8)

데이터 불러오기

SELECT * FROM 테이블명 WHERE 조건

ex) 멤버 이름 중에 첫 글자가 '김'씨 이거나 '황'씨 인 경우를 검색

```
UPDATE mymember SET youName = "황상연" WHERE myMemberID = 1;
```

```
MariaDB [test01] > SELECT * FROM mymember WHERE youName LIKE '김%' OR  
youName LIKE '황%';
```

myMemberID	youEmail	youName	youPass	youBirth	regTime
1	dkdlwo@naver.com	황상연	1234	1994-06-28	2147483647
3	guswd8694@naver	김우석	1234	2016-05-25	333
4	younyc0413@gmail.com	왕우석	0413	1992-04-13	333
6	seonyoung@gmail.com	황우석	1234	1997-04-03	111

4 rows in set (0.000 sec)





테이블 데이터(table data) : 수정하기

데이터 수정하기

UPDATE 테이블명(필드명) SET 필드명 조건

멤버 ID 값 변경하기

```
UPDATE mymember SET youName = "□□□" WHERE myMemberID = □;
```

ex) 회원 아이디 1번 이름 변경

```
UPDATE mymember SET youName = "김우석" WHERE myMemberID = 1;
```

```
1 | guswjd8694@naver | 김우석 | 1234 | 2016-05-25 | 333 |
```

모든 회원 비밀번호 값 변경하기

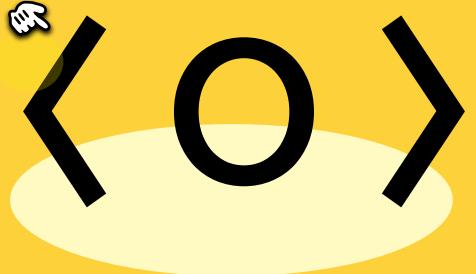
```
UPDATE mymember SET youPass = "□□□";
```

:결과
youPass
1234
1234
1234
1234

ex) 모든 회원 비밀번호를 '1234'로 변경

```
UPDATE mymember SET youPass = "1234";
```





테이블 데이터(table data) : 삭제하기

데이터 수정하기

UPDATE FROM 테이블명 조건

- 멤버 ID 레코드 삭제하기

```
DELETE FROM mymember WHERE myMemberID = □;
```

ex) 회원 아이디 1번 이름 변경

```
DELETE FROM mymember WHERE myMemberID = 3;
```

- 멤버 ID 레코드 복수 삭제하기

```
DELETE FROM mymember WHERE myMemberID IN(□,□,□);
```

```
DELETE FROM mymember WHERE myMemberID IN(4,5,6);
```





테이블(table Join) 조인

데이터 JOIN문 사용하기

SELECT 필드명 FROM 테이블명 앤리어스 JOIN 연결할 테이블명 앤리어스 ON(조건문)

테이블 생성하기

```
INSERT INTO myMember(youID, youName, youPass, youEmail, regTime)
VALUES('webstoryboy', '황상연', '1234', 'webs@naver.com', '1122');
INSERT INTO myMember(youID, youName, youPass, youEmail, regTime)
VALUES('webs3', '황가연', '1234', 'jayeon@naver.com', '1122');
INSERT INTO myMember(youID, youName, youPass, youEmail, regTime)
VALUES('bboyda', '김가연', '12df34', 'gayun@naver.com', '1122');
INSERT INTO myMember(youID, youName, youPass, youEmail, regTime)
VALUES('linkMe', '홍주연', '9999', 'juyeun@naver.com', '1122');
INSERT INTO myMember(youID, youName, youPass, youEmail, regTime)
VALUES('Juuuu', '박주연', '12df34', 'parkjuyun@naver.com', '1122');
```

```
INSERT INTO myReview(myMemberID, youCont, regTime) VALUES(3, '이 가방 너무 좋네요!! 당장 사야줘', '1122');
INSERT INTO myReview(myMemberID, youCont, regTime) VALUES(3, '이거 사지 마세요~~ 사기예요~~', '1122');
INSERT INTO myReview(myMemberID, youCont, regTime) VALUES(1, '여기 제품 좋습니다. 여기 알바생이에요~', '1122');
INSERT INTO myReview(myMemberID, youCont, regTime) VALUES(2, '이거 엄청 좋네요~~ 굳 추천!!', '1122');
```

테이블 생성 결과

```
mysql> SELECT * FROM myMember;
+-----+-----+-----+-----+
| myMemberID | youID   | youName | youPass | youEmail    | regTime |
+-----+-----+-----+-----+
| 1 | webstoryboy | 황상연 | 1234   | webs@naver.com | 1122 |
| 2 | webs3       | 황가연 | 1234   | jayeon@naver.com | 1122 |
| 3 | bboyda     | 김가연 | 12df34 | gayun@naver.com | 1122 |
| 4 | linkMe     | 홍주연 | 9999   | juyeun@naver.com | 1122 |
| 5 | Juuuu      | 박주연 | 12df34 | parkjuyun@naver.com | 1122 |
+-----+-----+-----+-----+
```

```
mysql> SELECT * FROM myreview;
+-----+-----+-----+
| myReviewID | myMemberID | youCont          | regTime |
+-----+-----+-----+
| 1 | 3 | 이 가방 너무 좋네요!! 당장 사야줘 | 1122 |
| 2 | 3 | 이거 사지 마세요~~ 사기예요~~ | 1122 |
| 3 | 1 | 여기 제품 좋습니다. 여기 알바생이에요~ | 1122 |
| 4 | 2 | 이거 엄청 좋네요~~ 굳 추천!! | 1122 |
+-----+-----+-----+
```





테이블(table Join) 조인

두개의 테이블 JOIN 하기

```
mysql> SELECT m.youName, r.youCont, r.regTime FROM mymember m JOIN myreview r ON (m.myMemberID = r.myMemberID);
```

youName	youCont	regTime
김가연	이 가방 너무 좋네요!! 당장 사야줘	1122
김가연	이거 사지 마세요~~ 사기예요~~	1122
황상연	여기 제품 좋습니다. 여기 알바생이영요~	1122
황가연	이거 엄청 좋네요~~ 굳 추천!!	1122

myMemberID 와 myReviewID를 서로 결합해서 사용하였습니다.

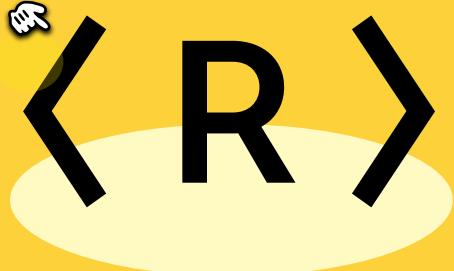
필드명을 변경하여 표시하는 방법

```
mysql> SELECT m.youName, r.youCont, r.regTime AS reviewTime FROM mymember m JOIN myreview r ON (m.myMemberID = r.myMemberID);
```

youName	youCont	reviewTime
김가연	이 가방 너무 좋네요!! 당장 사야줘	1122
김가연	이거 사지 마세요~~ 사기예요~~	1122
황상연	여기 제품 좋습니다. 여기 알바생이영요~	1122
황가연	이거 엄청 좋네요~~ 굳 추천!!	1122

필드명이 변경 되었습니다.





테이블(table Join) 조인

- 필드 내용이 없는 경우 표시하는 방법

```
mysql> SELECT m.youName, r.youCont, r.regTime FROM mymember m  
LEFT JOIN myreview r ON (m.myMemberID = r.myMemberID);  
+-----+-----+-----+  
| youName | youCont | regTime |  
+-----+-----+-----+  
| 김가연 | 이 가방 너무 좋네요!! 당장 사야줘 | 1122 |  
| 김가연 | 이거 사지 마세요~~ 사기예요~~ | 1122 |  
| 황상연 | 여기 제품 좋습니다. 여기 알바생이에요~ | 1122 |  
| 황가연 | 이거 엄청 좋네요~~ 굳 추천!! | 1122 |  
| 홍주연 | NULL | NULL |  
| 박주연 | NULL | NULL |  
+-----+-----+-----+  
6 rows in set (0.00 sec)
```

