

# MySQL 03

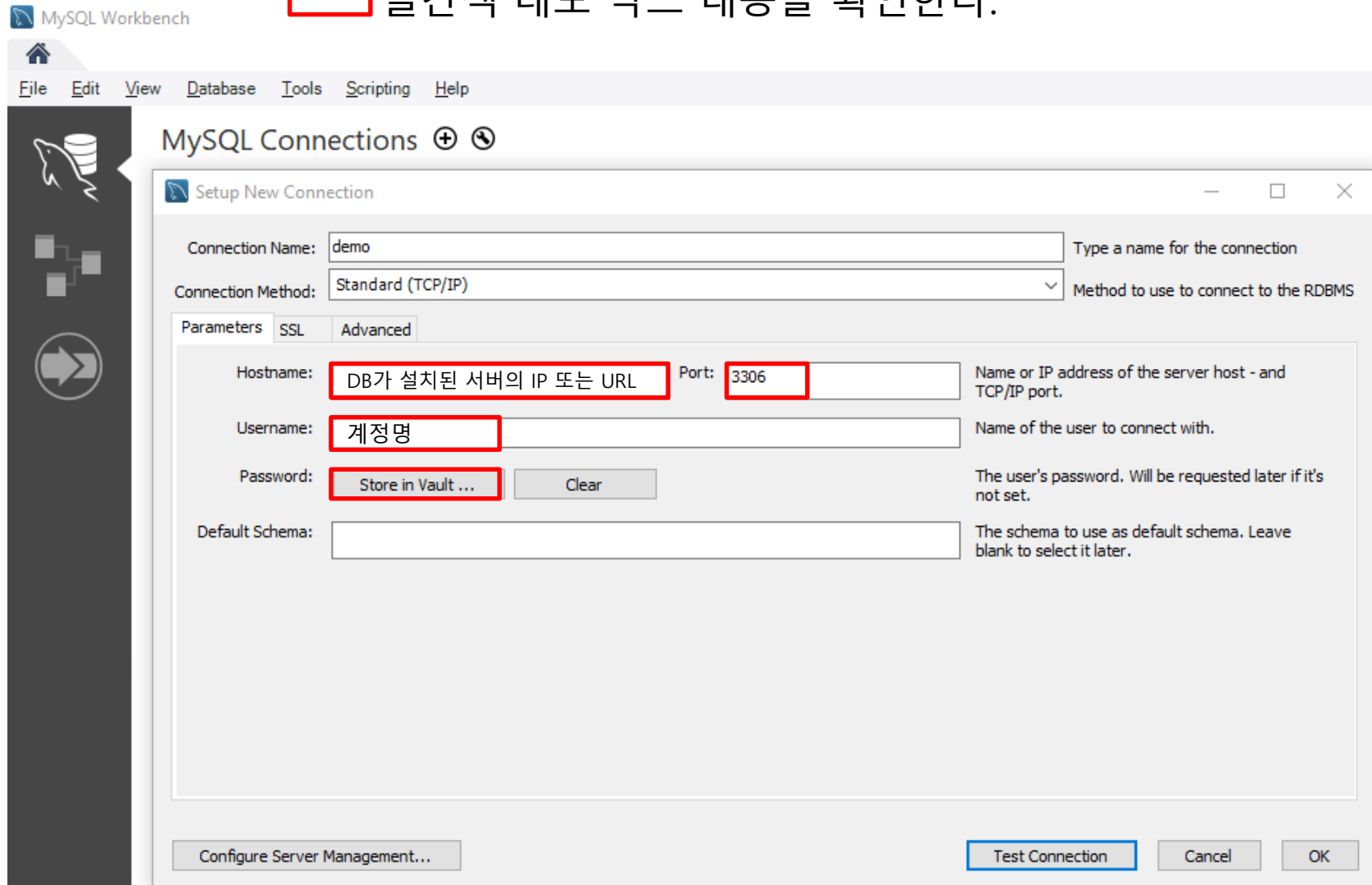
2023.06

# Workbench 사용전 준비사항

- Workbench는 Client용 Database 도구이다.
- Windows용 Workbench를 다운로드 하고 설치한다.
- SSH(22)가 아닌 MySQL(3306)으로 접속할 수 있도록 방화벽에서 접속을 허용해야 한다.  
firewall-cmd --permanent --add-service=mysql  
firewall-cmd --reload
- MariaDB(또는 MySQL)에서, 외부에서 접속 가능한 계정을 만들고 권한을 부여해야 한다.  
create user ~~~;  
grant ~~~;

# 접속 준비하기

  빨간색 네모 박스 내용을 확인한다.



The image shows the MySQL Workbench interface with the 'Setup New Connection' dialog box open. The dialog has tabs for 'Parameters', 'SSL', and 'Advanced'. The 'Parameters' tab is selected. The 'Connection Name' is 'demo'. The 'Connection Method' is 'Standard (TCP/IP)'. The 'Hostname' is 'DB가 설치된 서버의 IP 또는 URL' and the 'Port' is '3306'. The 'Username' is '계정명'. The 'Password' field has a 'Store in Vault ...' button. The 'Default Schema' is empty. At the bottom, there are buttons for 'Configure Server Management...', 'Test Connection', 'Cancel', and 'OK'.

MySQL Workbench

File Edit View Database Tools Scripting Help

MySQL Connections + -

Setup New Connection

Connection Name: demo Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: DB가 설치된 서버의 IP 또는 URL Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: 계정명 Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

Configure Server Management... Test Connection Cancel OK

# Workbench 사용 방법

The screenshot shows the MySQL Workbench interface with several annotations. A red arrow points from the 'Execute' button (a lightning bolt icon) in the SQL editor toolbar to a text box. Another red bracket points from the 'Result Grid' button in the toolbar to a text box. A third red bracket points from the 'Output' tab in the bottom panel to a text box. The SQL editor contains a script with various database commands. The Result Grid shows the results of the last query. The Output panel shows a log of database actions.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 2\*

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

sys

```
1 -- Create a database
2 DROP DATABASE IF EXISTS quickstartdb;
3 CREATE DATABASE quickstartdb;
4 USE quickstartdb;
5
6 -- Create a table and insert rows
7 DROP TABLE IF EXISTS inventory;
8 CREATE TABLE inventory (id SERIAL PRIMARY KEY, name VARCHAR(50), quantity INTEGER);
9 INSERT INTO inventory (name, quantity) VALUES ('banana', 150);
10 INSERT INTO inventory (name, quantity) VALUES ('orange', 154);
11 INSERT INTO inventory (name, quantity) VALUES ('apple', 100);
12
13 -- Read
14 SELECT * FROM inventory;
15
16 -- Update
17 UPDATE inventory SET quantity = 200 WHERE id = 1;
18 SELECT * FROM inventory;
19
20 -- Delete
21 DELETE FROM inventory WHERE id = 2;
22 SELECT * FROM inventory;
```

Result Grid

	id	name	quantity
▶	1	banana	200
	3	apple	100
*	NULL	NULL	NULL

inventory 1 inventory 2 inventory 3 x Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
21	17:11:46	INSERT INTO inventory (name, quantity) VALUES ('apple', 100)	1 row(s) affected	
22	17:11:47	SELECT * FROM inventory LIMIT 0, 1000	3 row(s) returned	
23	17:11:47	UPDATE inventory SET quantity = 200 WHERE id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	
24	17:11:47	SELECT * FROM inventory LIMIT 0, 1000	3 row(s) returned	
25	17:11:48	DELETE FROM inventory WHERE id = 2	1 row(s) affected	
26	17:11:48	SELECT * FROM inventory LIMIT 0, 1000	2 row(s) returned	0.078 sec / 0.000 sec

실행할 명령어줄에 커서를 놓고,  
커서모양이 있는 번개를 누르면 실행됩니다.

명령어 줄  
명령어를 여기에 입력  
(한꺼번에 미리 넣어놔도 됨)

결과가 표시되는 부분

작업에 대한 로그가 기록됨  
성공 or 실패에 대한 기록

# 데이터베이스와 테이블 생성하기

---

```
create database Kset default character set utf8 collate utf8_general_ci;
```

```
use Kset;
```

```
create table member (  
    custno int(6),  
    custname varchar(20),  
    phone varchar(13),  
    address varchar(60),  
    joindate date,  
    grade char(1),  
    city char(2),  
    primary key(custno)  
);
```

# 테이블 생성하기

---

```
create sequence member_seq  
start with 10001  
increment by 1  
minvalue 10001  
cache 10;
```

```
create table money (  
  custno int(6),  
  salenol int(8),  
  pcost int(8),  
  amount int(4),  
  price int(8),  
  pcode varchar(4),  
  sdate date,  
  primary key(custno, salenol)  
);
```

# 데이터 입력 (회원 가입)

```
insert into member values(10001, '김행복', '010-1111-2222', '서울 동대문구 휘경1동', '20151202', 'A', '01');  
insert into member values(10002, '이축복', '010-1111-3333', '서울 동대문구 휘경2동', '20151206', 'B', '01');  
insert into member values(10003, '장믿음', '010-1111-4444', '울릉군 울릉읍 독도1리', '20151001', 'B', '30');  
insert into member values(10004, '최사랑', '010-1111-5555', '울릉군 울릉읍 독도2리', '20151113', 'A', '30');  
insert into member values(10005, '진평화', '010-1111-6666', '제주도 제주시 외나무골', '20151225', 'B', '60');  
insert into member values(10006, '차공단', '010-1111-7777', '제주도 제주시 감나무골', '20151211', 'C', '60');  
insert into member values(10007, '내이름', '010-1234-5678', '서울 중구 필동로 1길', '20210302', 'A', '01');
```

'내이름'에 여러분들의 이름을 추가합니다.  
(개인정보를 너무 자세히 넣을 필요는 없음)

위와 같은 방법으로 데이터를 몇 줄 추가로 더 입력해봅니다.

이름 바꾸기

```
update member set custname= ' 홍길동' where custno=10007;
```

# 데이터 입력 (주문 입력)

```
insert into money values(10001, 20210001, 500, 5, 2500, 'A001', '20190101');
insert into money values(10001, 20210002, 1000, 4, 4000, 'A002', '20200501');
insert into money values(10001, 20210003, 500, 3, 1500, 'A008', '20190607');
insert into money values(10002, 20210004, 2000, 1, 2000, 'A004', '20200102');
insert into money values(10002, 20210005, 500, 1, 500, 'A001', '20200103');
insert into money values(10003, 20210006, 1500, 2, 3000, 'A003', '20210103');
insert into money values(10004, 20210007, 500, 2, 1000, 'A001', '20210104');
insert into money values(10004, 20210008, 300, 1, 300, 'A005', '20210104');
insert into money values(10004, 20210009, 600, 1, 600, 'A006', '20210104');
insert into money values(10004, 20210010, 3000, 1, 3000, 'A007', '20210106');
insert into money values(10006, 20210010, 500, 3, 1500, 'A001', '20210106');
insert into money values(10007, 20210011, 3000, 1, 3000, 'A007', '20210211');
```

위와 같은 방법으로 데이터를 추가로 더 입력해봅니다.



# select문을 사용해서 데이터 추출하기

---

```
select * from money;
```

```
select sum(price) from money;
```

```
select sum(price) from money group by custno;
```

```
select custno,sum(price) from money group by custno;
```

```
select custno,sum(price) from money group by custno order by sum(price) desc;
```

```
select * from member where grade='a';
```

이 외에도 다양한 방법으로 데이터를 추출해보자

# 컬럼 추가하고 데이터 입력하기

\* money 테이블에 mention이라는 컬럼을 추가합니다. 자료형은 text로 합니다.  
alter table money add mention text;

\* 2개 이상 구매한 경우, '좋아요'라고 mention에 추가합니다.  
update money set mention = '좋아요' where amount >= 2;

\* 1개만 구매한 경우, '그럭저럭 쓸만해요.'라고 mention컬럼에 추가합니다.  
update money set mention = '그럭저럭 쓸만해요' where amount = 1;

\* 테이블에 내용이 잘 들어갔는지 확인해봅니다.  
select \* from money;

```
MariaDB [Kset]> select * from money;
```

custno	salenol	pcost	amount	price	pcode	sdate	mention
10001	20160001	500	5	2500	A001	2016-01-01	좋아요
10001	20160002	1000	4	4000	A002	2016-01-01	좋아요
10001	20160003	500	3	1500	A008	2016-01-01	좋아요
10002	20160004	2000	1	2000	A004	2016-01-02	그럭저럭 쓸만해요
10002	20160005	500	1	500	A001	2016-01-03	그럭저럭 쓸만해요
10003	20160006	1500	2	3000	A003	2016-01-03	좋아요
10004	20160007	500	2	1000	A001	2016-01-04	좋아요
10004	20160008	300	1	300	A005	2016-01-04	그럭저럭 쓸만해요
10004	20160009	600	1	600	A006	2016-01-04	그럭저럭 쓸만해요
10004	20160010	3000	1	3000	A007	2016-01-06	그럭저럭 쓸만해요

```
10 rows inset (0.000 sec)
```

# Join 사용 방법

---

```
select member.custname,money.mention from member join money;
```

(문법 정리)  
테이블명.컬럼명

from 뒤에는 테이블명

테이블 두개는 join으로 연결  
TableA join TableB

합쳐보면  
select TableA.ColumnA, TableB.ColumnB from TableA join TableB;

TableA와 TableB의 공통 컬럼이 있으면 on 뒤에 조건으로 붙여줌

# Join (두개의 테이블에서 데이터 골라보기)

\* member테이블에서 이름과 전화번호를 추출하고, money테이블에서 mention을 추출한 다음에 양쪽 테이블의 custno가 일치하도록 매칭시켜서 하나의 테이블로 만들어봅니다.

```
select member.custname,member.phone,money.mention from member join  
money on member.custno=money.custno;
```

```
MariaDB [Kset]> select member.custname,member.phone,money.mention from member join money on member.custno=money.custno;
```

custname	phone	mention
김행복	010-1111-2222	좋아요
김행복	010-1111-2222	좋아요
김행복	010-1111-2222	좋아요
이축복	010-1111-3333	그럭저럭 쓸만해요
이축복	010-1111-3333	그럭저럭 쓸만해요
장민음	010-1111-4444	좋아요
최사랑	010-1111-5555	좋아요
최사랑	010-1111-5555	그럭저럭 쓸만해요
최사랑	010-1111-5555	그럭저럭 쓸만해요
최사랑	010-1111-5555	그럭저럭 쓸만해요

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

※ insert문을 얼마나 추가했는지에 따라 결과는 달라질 수 있음

# distinct : 중복 제거

\* 출력된 내용에 중복 데이터가 있으므로 중복을 제거

```
select distinct member.custname,member.phone,money.mention from  
member join money on member.custno=money.custno;
```

```
MariaDB [Kset]> select distinct member.custname,member.phone,money.mention from member join money  
on member.custno=money.custno;
```

custname	phone	mention
김행복	010-1111-2222	좋아요
이축복	010-1111-3333	그럭저럭 쓸만해요
장민음	010-1111-4444	좋아요
최사랑	010-1111-5555	좋아요
최사랑	010-1111-5555	그럭저럭 쓸만해요

```
5 rows in set (0.012 sec)
```

※ insert문을 얼마나 추가했는지에 따라 결과는 달라질 수 있음

# 상품 코드를 추가하여 만족도 조사

\* 상품에 따라 만족 정도가 달랐던 걸까?

```
select member.custname,member.phone,money.pcode,money.mention  
from member join money on member.custno=money.custno;
```

```
MariaDB [Kset]> select member.custname,member.phone,money.pcode,money.mention from member join money on member.custno=money.custno;
```

custname	phone	pcode	mention
김행복	010-1111-2222	A001	좋아요
김행복	010-1111-2222	A002	좋아요
김행복	010-1111-2222	A008	좋아요
이축복	010-1111-3333	A004	그럭저럭 쓸만해요
이축복	010-1111-3333	A001	그럭저럭 쓸만해요
장민음	010-1111-4444	A003	좋아요
최사랑	010-1111-5555	A001	좋아요
최사랑	010-1111-5555	A005	그럭저럭 쓸만해요
최사랑	010-1111-5555	A006	그럭저럭 쓸만해요
최사랑	010-1111-5555	A007	그럭저럭 쓸만해요

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

※ insert문을 얼마나 추가했는지에 따라 결과는 달라질 수 있음

# QnA

본 강의 자료는 K-DT의 강의를 위해 제작되었습니다.  
저자의 서면 허락없이 배포를 금합니다. 저자 : 소영재 jdrsecure@gmail.com  
Copyright© 2023 소영재. All rights reserved