

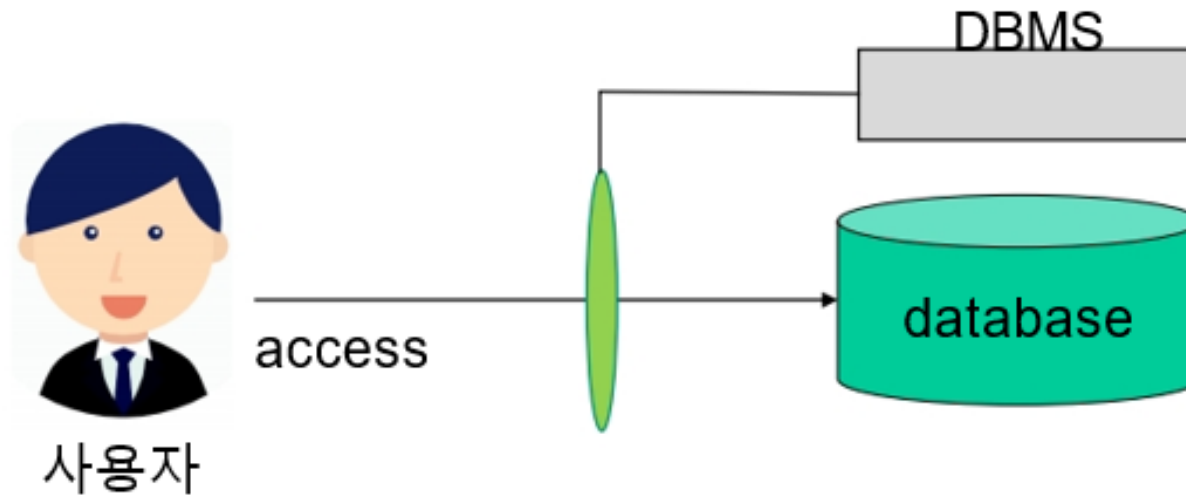
## Chapter 11

# 데이터베이스 보안

- DBMS 의 목적중 하나는 데이터베이스에 저장된 정보를 안전하게 보호하는 것
- 이를 수행하기 위한 수단
  - 추가 로그인
  - 암호화
  - 사용자 관리
  - 권한 관리

# 1. 사용자 관리

- 사용자가 Database 를 이용하기 위해서는 계정(account) 이 필요
- 사용자가 DBMS에 로그인하면 DBMS 는 등록된 사용자인지를 검사한다 .
- 이를 위해 사전에 계정 생성이 필요



# 1. 사용자 관리

- root
  - 최고 권한을 가진 사용자 계정
  - 데이터베이스 및 DBMS 에 대한 모든 권한을 갖는다
  - 비밀번호가 노출되면 위험
  - 일반 사용자(개발자)들에게 root 권한을 주는 것은 위험하기 때문에 별도의 계정을 만들어 사용하도록 해야 한다

# 1. 사용자 관리

- 시스템에 등록된 사용자 알아보기

Navigation: 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

Administration Schemas

Information

No object selected

Query 1 Administration - Users and Priv...

Local instance MySQL80

## Users and Privileges

User Accounts

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost

Details for account root@localhost

Login Account Limits Administrative Roles Schema Privileges

Login Name: root

Authentication Type: caching\_sha2\_password

Limit to Hosts Matching: localhost

Password: \*\*\*\*\*

Confirm Password: \*\*\*\*\*

Expire Password

Authentication String: \$A\$005\$-v\*a\*t9PsV9KG7Wcw%ETf

See the plugin documentation for valid values and details.

Add Account Delete Refresh

# 1. 사용자 관리

- 사용자 생성

Query 1 Administration - Users and Privil... x

Local instance MySQL80  
Users and Privileges

User Accounts

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
newuser	%
root	localhost

Details for account newuser@%

Login Account Limits Administrative Roles Schema Privileges

Login Name: user\_1

Authentication Type: Standard

Limit to Hosts Matching: localhost

Password: \*\*\*\*

Confirm Password: \*\*\*\*

Expire Password

Add Account Delete Refresh

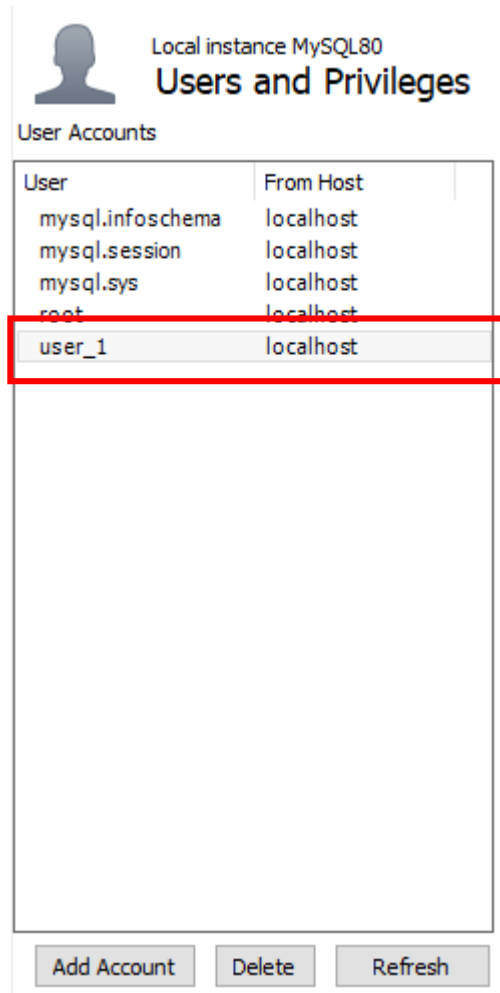
Revert Apply

① 컴퓨터 접속이 인터넷도 접속

② Weak password.

\* localhost ⇒ 원본은 X  
only 내 컴퓨터에서만 가능..!

# 1. 사용자 관리



The screenshot shows the 'Users and Privileges' window for a 'Local instance MySQL80'. It features a 'User Accounts' table with columns 'User' and 'From Host'. The table lists several users, with 'user\_1' highlighted by a red rectangle. Below the table are three buttons: 'Add Account', 'Delete', and 'Refresh'.

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
user_1	localhost

# 1. 사용자 관리

- 권한 부여

Local instance MySQL80  
**Users and Privileges**

User Accounts

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
user_1	localhost

Details for account user\_1@localhost

Login Account Limits Administrative Roles **Schema Privileges**

Schema	Privileges
--------	------------

Schema and Host fields may use % and \_ wildcards.  
The server will match specific entries before wildcarded ones.

Revoke All Privileges Delete Entry **Add Entry..**

모든권한 회수

권한 추가

현재 아무 권한이 없음



# 1. 사용자 관리

New Schema Privilege Definition

Select the Schema for which the user 'user\_1' will have the privileges you want to define.

Schema

☐ All Schema (%) *모든 DB에 대해서 권한 주겠다*

☐ Schemas matching pattern:  *Pattern에 맞는 DB만 줄*

☒ Selected schema:  *선택*

This rule will apply to any schema name.

This rule will apply to schemas that match the given name or pattern. You may use \_ and % as wildcards in a pattern. Escape these characters with \ in case you want their literal value.

Select a specific schema name for the rule to apply to.

Cancel OK

# 1. 사용자 관리

**Details for account user\_1@localhost**

Login Account Limits Administrative Roles Schema Privileges

Schema	Privileges
my_db	SELECT, SHOW VIEW

< [Progress Bar]

Schema and Host fields may use % and \_ wildcards.  
The server will match specific entries before wildcarded ones.

Revoke All Privileges Delete Entry Add Entry..

The user 'user\_1'@'localhost' will have the following access rights to schemas matching 'my\_db':

**Object Rights**

- ☒ SELECT
- ☐ INSERT
- ☐ UPDATE
- ☐ DELETE
- ☐ EXECUTE
- ☒ SHOW VIEW

**DDL Rights**

- ☐ CREATE
- ☐ ALTER
- ☐ REFERENCES
- ☐ INDEX
- ☐ CREATE VIEW
- ☐ CREATE ROUTINE
- ☐ ALTER ROUTINE
- ☐ EVENT
- ☐ DROP
- ☐ TRIGGER

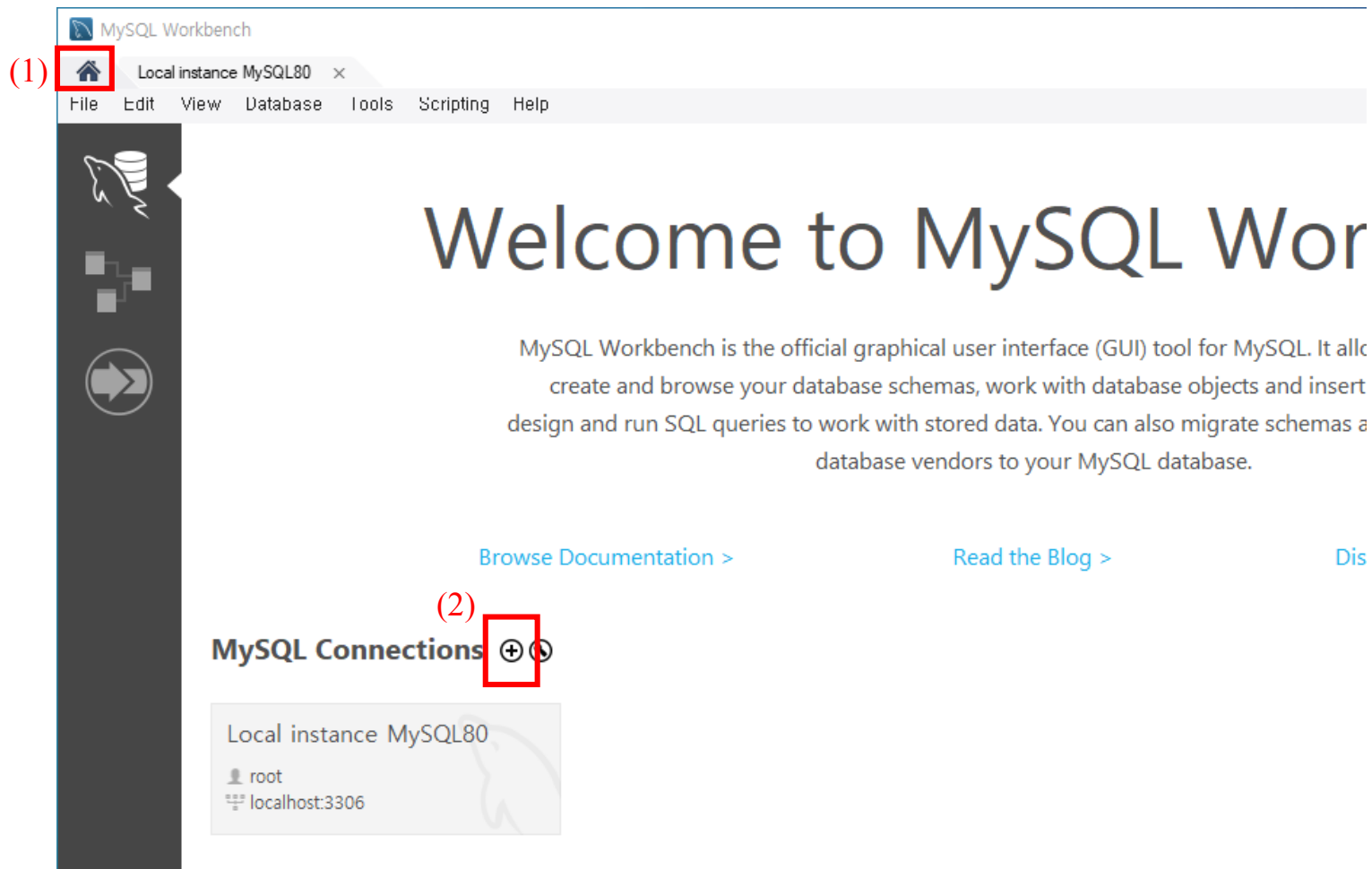
**Other Rights**

- ☐ GRANT OPTION
- ☐ CREATE TEMPORARY TABLES
- ☐ LOCK TABLES

Revert Apply

# 1. 사용자 관리

- Mysql workbench 커넥션 생성



# 1. 사용자 관리

- Mysql workbench 커넥션 생성

Setup New Connection

Connection Name:  Type a name for the connection

Connection Method:  Method to use to connect to the RDBMS

Parameters SSL Advanced

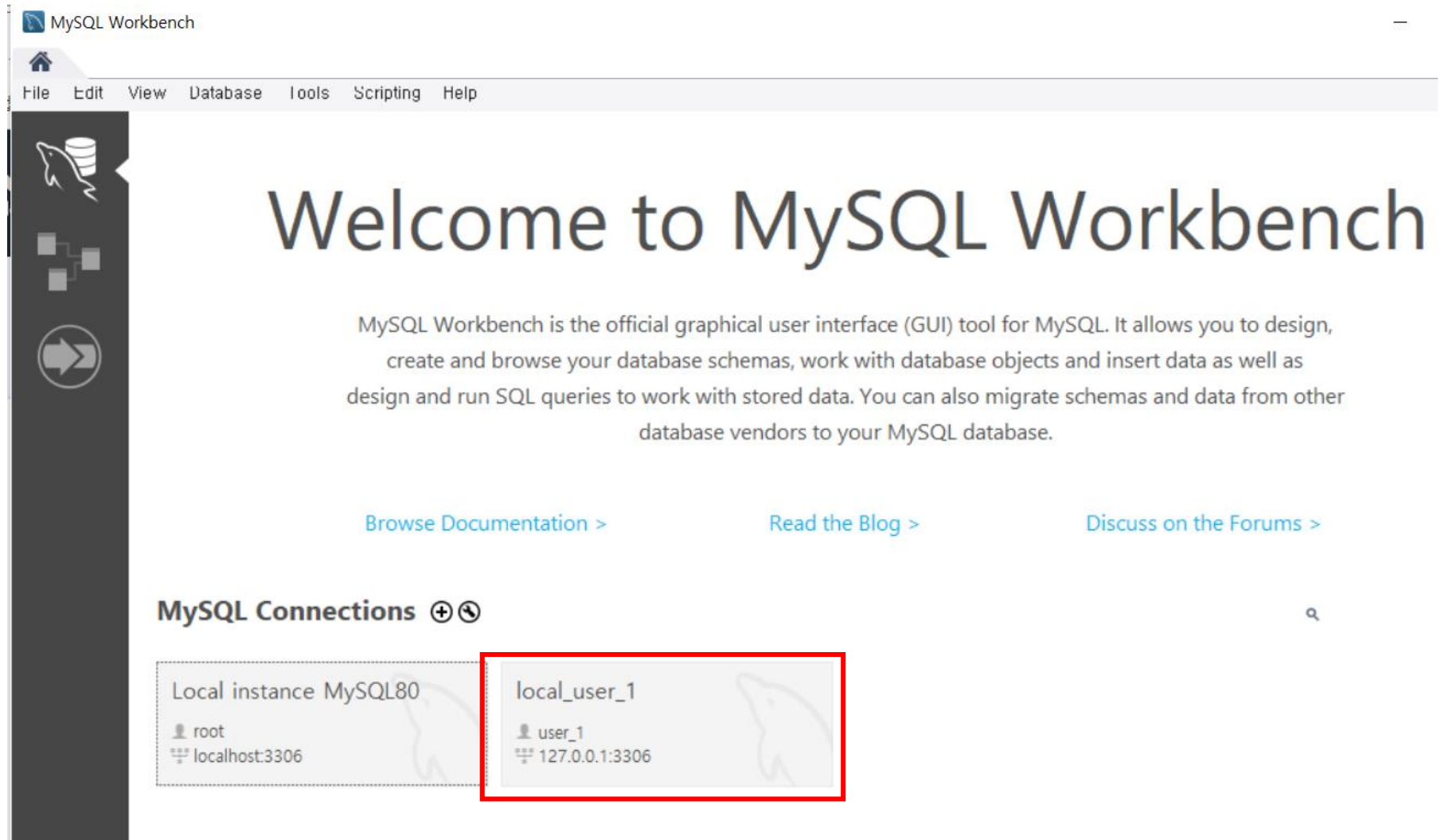
Hostname:  Port:  Name or IP address of the server host - and TCP/IP port.

Username:  Name of the user to connect with.

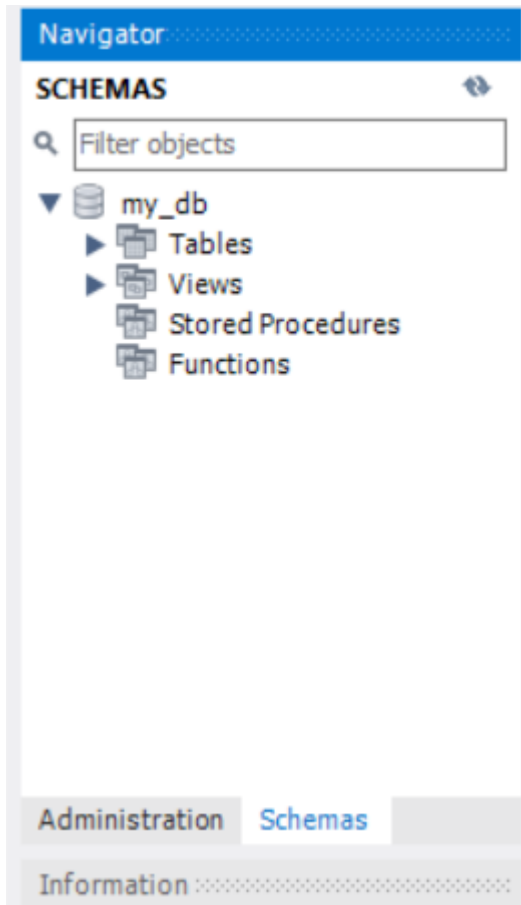
Password:   The user's password. Will be requested later if it's not set.

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

# 1. 사용자 관리



# 1. 사용자 관리



- 권한 부여를 my\_db 에 대해서만 했으므로  
X 다른 것은 보이지 않음
- my\_db 의 테이블들에 대해서는 조회 권한  
만 있으므로 (insert, update, delete) 는 안된다.

X


## Apply SQL Script to Database

Review SQL Script

Apply SQL Script

### Applying SQL script to the database

The following tasks will now be executed. Please monitor the execution.  
Press Show Logs to see the execution logs.

 Execute SQL Statements

Error: There was an error while applying the SQL script to the database.

#### Message Log

```
Operation failed: There was an error while applying the SQL script to the database.  
Executing:  
INSERT INTO `my_db`.`dept` (`DEPTNO`, `DNAME`, `LOC`) VALUES ('50', 'test', 'Seoul');  
  
ERROR 1142: 1142: INSERT command denied to user 'user_1'@'localhost' for table 'dept'  
SQL Statement:  
INSERT INTO `my_db`.`dept` (`DEPTNO`, `DNAME`, `LOC`) VALUES ('50', 'test', 'Seoul')
```

Hide Logs

Back

Finish

Cancel

# 1. 사용자 관리

- Note

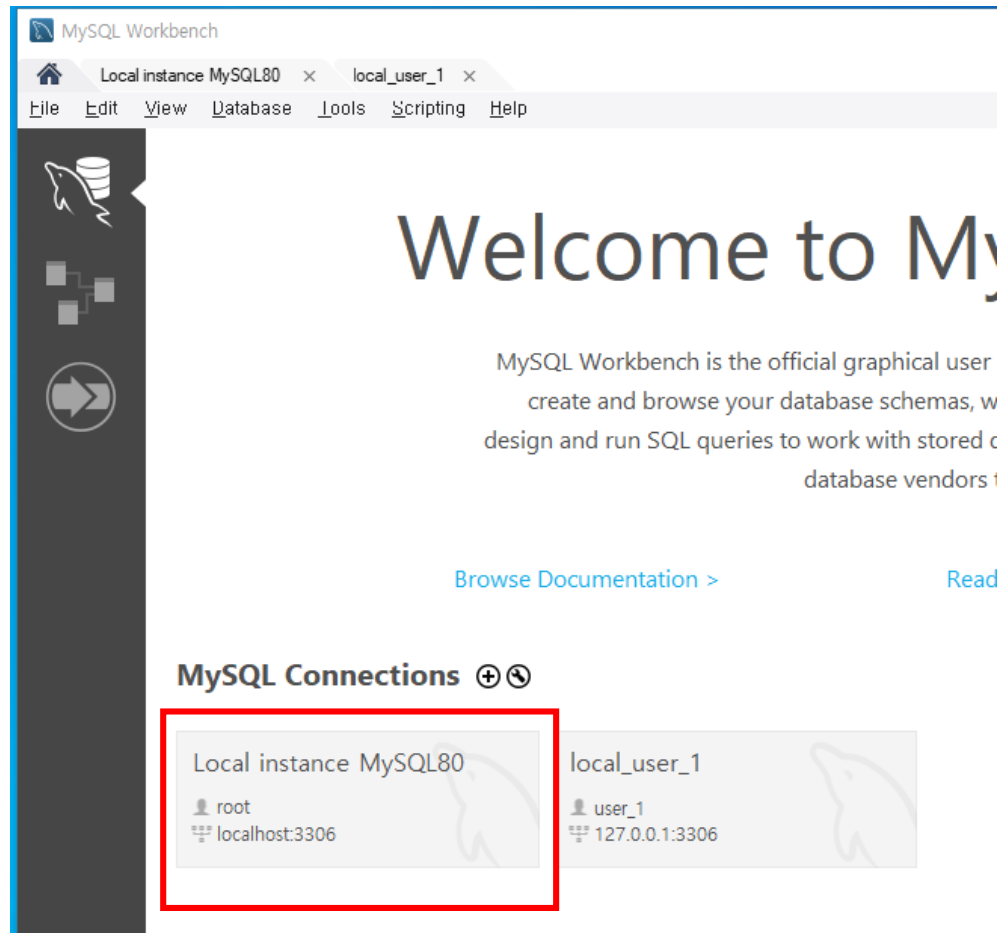
- MySQL workbench는 DB 단위로 권한 부여/회수
- SQL 명령문 (grant, revoke) 을 이용하면 테이블 단위로도 권한을 부여/회수 할 수 있다.

*select 권한*  
**GRANT** select **ON** my\_db.emp **TO** user\_1@localhost;  
*여기*  
**GRANT** (select, insert, update) **ON** my\_db.dept **TO** user\_1@localhost;



## 2. 사용자 관리 SQL

- SQL 문을 이용해서 사용자 생성, 권한 부여 및 회수가 가능함
- user\_1 은 조회권한 밖에 없으므로 root 로 접속하여 작업한다.



## 2. 사용자 관리 SQL

- 사용자의 생성

Localhost 사용자

```
create user user_2@localhost identified by  
'4321';
```

원격접속 사용자

이 는 꼭 있어야 함

비번은 암호화 되어 저장됨

```
create user 'user_2'@'%' identified by '4321';
```

원격

생성된 사용자 확인

sys b13

```
Select * from mysql.user ;
```

	Host	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv
▶	%	user_2	N	N	N	N	N	N
	localhost	mysql.infoschema	Y	N	N	N	N	N
	localhost	mysql.session	N	N	N	N	N	N
	localhost	mysql.sys	N	N	N	N	N	N
	localhost	root	Y	Y	Y	Y	Y	Y
	localhost	user_1	N	N	N	N	N	N
	localhost	user_2	N	N	N	N	N	N
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## 2. 사용자 관리 SQL

- 권한의 부여

데이터베이스에 대한 모든 권한 부여

```
grant all privileges on my_db.* to user_2@localhost;
```

데이터베이스에 대한 일부 권한 부여

```
grant select, insert on my_db.* to user_1@localhost;
```

테이블에 대한 모든 권한 부여

```
grant all privileges on my_db.emp to user_1@localhost;
```

테이블에 대한 일부 권한 부여

```
grant select, insert on my_db.emp to user_1@localhost;
```

## 2. 사용자 관리 SQL

- 권한의 부여

데이터베이스에 대한 모든 권한 부여 with grant option

```
grant all privileges on my_db.* to user_2@localhost  
with grant option;
```

↳ 다른 사용자에게도

자신의 권한 부여 가능



DB \* 1이 2에게도 보인이 받은 권한은 계속 있음!!

부여된 권한 확인

```
flush privileges; // 변경된 내용을 메모리에 반영 (권한 적용)
```

```
SHOW GRANTS FOR user_1@localhost;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Grants for user_1@localhost			
▶ GRANT USAGE ON *.* TO 'user_1'@'localhost'			
GRANT SELECT, SHOW VIEW ON 'my_db'.* TO 'user_1'@'localhost'			
GRANT ALL PRIVILEGES ON 'my_db'.'emp' TO 'user_1'@'localhost'			

## 2. 사용자 관리 SQL

- 권한의 회수

```
revoke delete on my_db.emp from user_2@localhost;
```

delete 권한 회수

- 사용자 삭제

```
drop user user_2@localhost;
```

## 2. 사용자 관리 SQL

- 6 그룹에 권한 줌, 2 그룹에 사용자 등록
- 역할(role)  $\Rightarrow$  user group
    - 영업 업무를 하는 세명의 사용자(s\_user1, s\_user2, s\_user3)이 있다
    - 필요한 데이터 베이스 권한은 emp 에 대한 select, update 이다.
    - 세명에게 각각 권한을 부여하지 않고 쉽게 할 수 있는 방법은?  $\Rightarrow$  role  
 $\Rightarrow$  group에 권한 줌!! (\* 좋은 아이디어~(축하...))

```
create role sales_role; 이름  
grant select, update on my_db.emp to sales_role;  
grant sales_role to s_user1@localhost;  
grant sales_role to s_user2@localhost;  
grant sales_role to s_user3@localhost;
```

그냥 테이블 권한 No??  
왜냐하면?...

## 2. 사용자 관리 SQL

- Mysql privileges

Privilege	Privilege	Privilege
<u>ALL [PRIVILEGES]</u>	<u>GRANT OPTION</u>	
<u>ALTER</u>		<u>TRIGGER</u>
<u>ALTER ROUTINE</u>	<u>INDEX</u>	<u>UPDATE</u>
<u>CREATE</u>	<u>INSERT</u>	<u>USAGE</u>
<u>CREATE ROLE</u>	<u>LOCK TABLES</u>	
<u>CREATE ROUTINE</u>		
<u>CREATE TABLESPACE</u>	<u>PROCESS</u>	
<u>CREATE TEMPORARY TABLES</u>	<u>PROXY</u>	
<u>CREATE USER</u>	<u>REFERENCES</u>	
	<u>RELOAD</u>	
<u>CREATE VIEW</u>	<u>REPLICATION CLIENT</u>	
<u>DELETE</u>	<u>REPLICATION SLAVE</u>	
<u>DROP</u>	<u>SELECT</u>	
<u>DROP ROLE</u>	<u>SHOW DATABASES</u>	
<u>EVENT</u>	<u>SHOW VIEW</u>	
<u>EXECUTE</u>	<u>SHUTDOWN</u>	
<u>FILE</u>	<u>SUPER</u>	

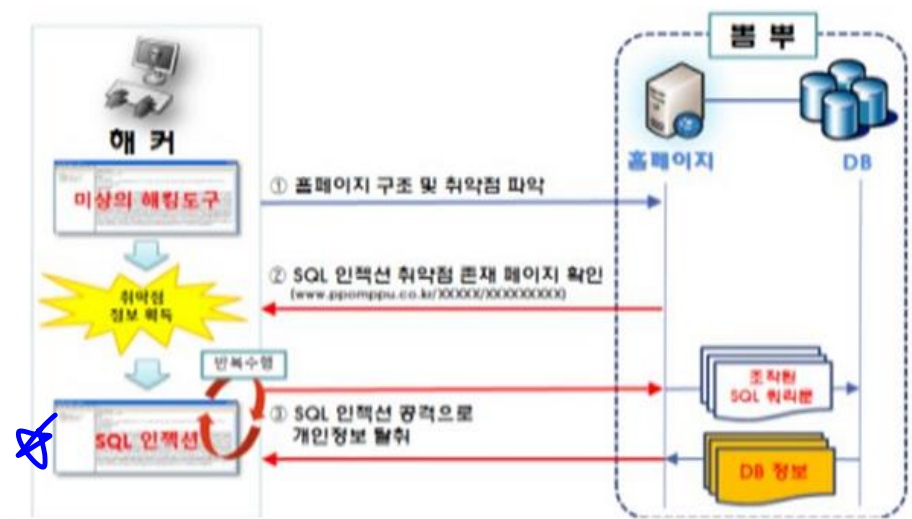
# 3. SQL 삽입 공격

## 뽐뿌 해킹사건, 웹 취약점을 악용한 데이터베이스 공격으로 밝혀져

원태영 기자(won@sisabiz.com)

승인 2015.10.20 12:51 댓글 0

공유 인쇄 이메일 카카



자료-미래부 제공

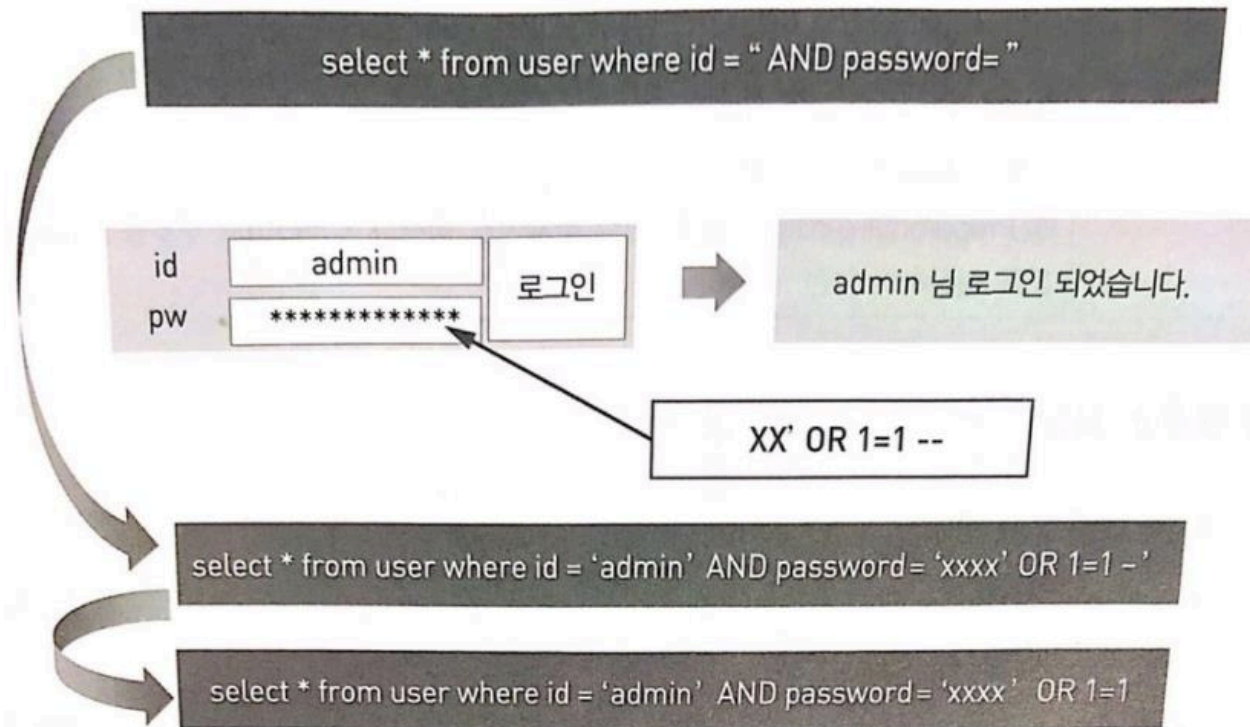
휴대폰 온라인 커뮤니티 뽐뿌의 해킹이 웹 취약점을 악용한 데이터베이스 공격인 것으로 밝혀졌다.

20일 미래창조과학부는 지난 9월 발생한 뽐뿌 홈페이지 해킹사고와 관련해 해킹방법, 사고원인 등에 대한 민관 합동조사단의 조사결과를 발표했다. 뽐뿌는 지난 해킹사건으로 약 196만명의 회원정보가 유출됐다.



### 3. SQL 삽입 공격

- 웹 애플리케이션에서 입력데이터에 대한 유효성 검증을 하지 않을 경우, 공격자가 입력 창 및 URL에 SQL 문을 삽입하여 DB로부터 정보를 열람, 조작할 수 있는 취약점 공격기법



<https://yoonfit.tistory.com/59>

## [연습 1]

1. 사용자 m\_user1, muser2 를 workbench 의 기능을 이용하여 생성하시오
2. 사용자 m\_user3, muser4 를 SQL문을 이용하여 생성하시오
3. m\_user1 에게 sakila 데이터베이스에 대한 모든 권한을 부여 하시오 ( grant option 포함)
4. m\_user2 에게 sakila 데이터베이스의 모든 테이블에 대한 select 권한을 부여하시오
5. m\_user3 에게 sakila 데이터베이스의 actor 테이블에 대한 select, insert 권한을 부여하시오
6. m\_user4 에게 sakila 데이터베이스의 city 테이블에 대한 select, update, delete 권한을 부여하시오
7. m\_user4 로부터 sakila 데이터베이스의 city 테이블에 대한 delete 권한을 회수하시오
8. m\_user3 로부터 sakila 데이터베이스의 actor 테이블에 대한 insert 권한을 회수하시오
9. 역할 manager 를 생성하고 world 데이터베이스에 대한 select, insert, update 권한을 부여하시오.
10. 역할 manager 를 m\_user1, m\_user2, m\_user3 에게 부여하시오
11. 사용자 m\_user4 를 삭제하시오.