

Database and Applications

Project 1 - Web Tutorial

Reference

<http://devlecture.tistory.com/>

Kwangwoon Univ.
Dept. of Computer Engineering
Ki-Hoon Lee



Contents

1. Node.js 설치 및 예제 실행
2. Express 설치 및 예제 실행
3. Ejs
4. 회원가입 폼 만들기 및 nodemon 사용
5. Database(MySQL) 연동
6. 게시판 구축



1. node.js 설치 및 예제 실행

- nodejs.org 접속 후 node.js 설치



Node.js®는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.

다운로드 - Windows (x64)

LTS 버전 설치

12.16.1 LTS

안정적, 신뢰도 높음

13.11.0 현재 버전

최신 기능

[다른 운영 체제](#) | [변경사항](#) | [API 문서](#)

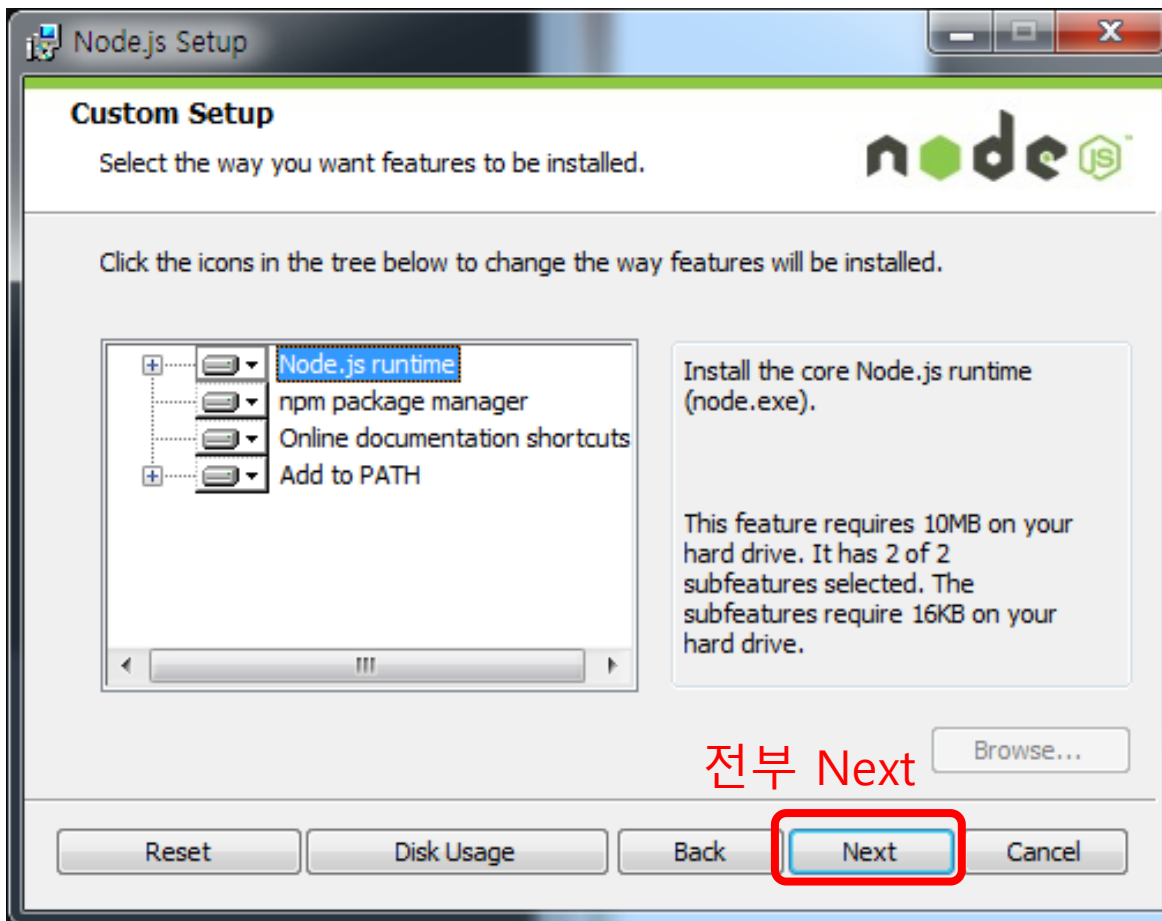
[다른 운영 체제](#) | [변경사항](#) | [API 문서](#)

LTS 일정은 [여기서](#) 확인하세요.



1. node.js 설치 및 예제 실행

■ 설치파일 실행



1. node.js 설치 및 예제 실행

- helloNode 폴더 생성 후 폴더 안에 hello.js 파일 생성



내 PC > 로컬 디스크 (C:) > helloNode			
이름	수정된 날짜	유형	크기
hello.js	2020-03-06 오후...	JS 파일	1KB

- 예제소스 작성

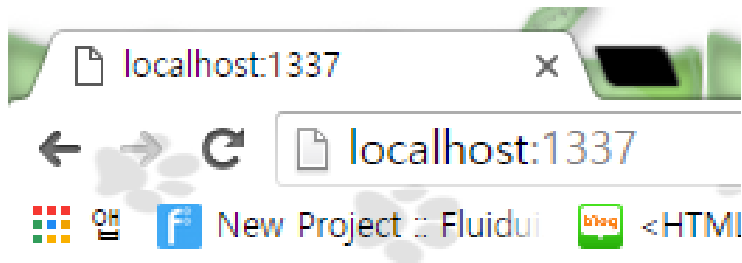
```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```



1. node.js 설치 및 예제 실행

- cmd > node.js 파일이 있는 폴더로 이동(cd 명령어 이용)
=>node hello.js 입력

```
C:\ node hello.js  
  
D:\Users\User\Documents\helloNode>node hello.js  
Server running at http://127.0.0.1:1337/
```



Hello World

1. node.js 설치 및 예제 실행

- Ctrl + C를 통해 종료

```
C:\WINDOWS\system32

D:\Users\User\Documents\helloNode>node hello.js
Server running at http://127.0.0.1:1337/
^C
D:\Users\User\Documents\helloNode>
```



2. Express 설치 및 예제 실행

- 이전 예제와 같이 새로운 폴더 생성 후 명령프롬프트 실행
- >npm install -g express-generator 입력

```
C:\Windows\system32\cmd.exe

C:\myNode>npm install -g express-generator
C:\Users\<redacted>\AppData\Roaming\npm\express -> C:\Users\<redacted>\AppData\Roaming\npm\express-cli.js
+ express-generator@4.16.1
updated 1 package in 0.507s

New minor version of npm available! 6.13.4 -> 6.14.2
Changelog: https://github.com/npm/cli/releases/tag/v6.14.2
Run npm install -g npm to update!

C:\myNode>
```




2. Express 설치 및 예제 실행

- > express example1 입력

```
D:\Users\User\Documents\myNode>express example1
```

```
create : example1
create : example1/package.json
create : example1/app.js
create : example1/public
create : example1/public/stylesheets
create : example1/public/stylesheets/style.css
create : example1/public/images
create : example1/public/javascripts
create : example1/routes
create : example1/routes/index.js
create : example1/routes/users.js
create : example1/views
create : example1/views/index.jade
create : example1/views/layout.jade
create : example1/views/error.jade
create : example1/bin
create : example1/bin/www
```

```
install dependencies:
```

```
> cd example1 && npm install
```

```
run the app:
```

```
> SET DEBUG=example1:* & npm start
```



2. Express 설치 및 예제 실행

- >cd example1 # example1로 이동
- >npm install # 모듈 설치

```
C:\WINDOWS\system32\cmd
D:\Users\User\Documents\myNode>cd example1

D:\Users\User\Documents\myNode\example1>npm install
debug@2.2.0 node_modules\debug
├── ms@0.7.1

cookie-parser@1.3.5 node_modules\cookie-parser
├── cookie@0.1.3
├── cookie-signature@1.0.6

serve-favicon@2.3.0 node_modules\serve-favicon
├── etag@1.7.0
├── fresh@0.3.0
├── parseurl@1.3.1
├── ms@0.7.1

morgan@1.6.1 node_modules\morgan
├── on-headers@1.0.1
├── basic-auth@1.0.3
├── depd@1.0.1
├── on-finished@2.3.0 (ee-first@1.1.1)

express@4.13.4 node_modules\express
├── utils-merge@1.0.0
```



2. Express 설치 및 예제 실행

- >npm start 혹은 >node .\bin\www 실행



npm

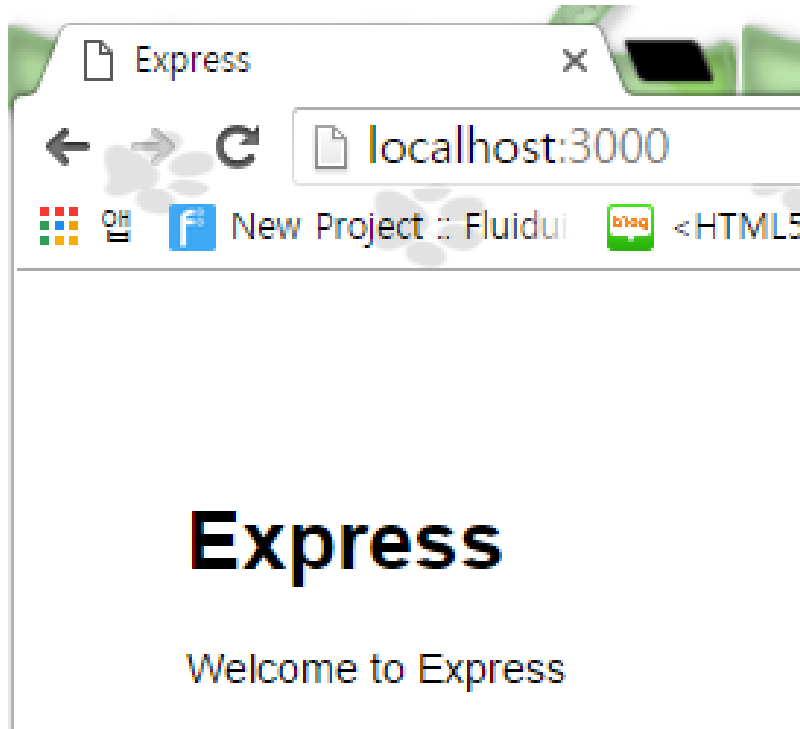
```
D:\Users\User\Documents\myNode\example1>npm start
```

```
> example1@0.0.0 start D:\Users\User\Documents\myNode\example1
```

```
> node ./bin/www
```

2. Express 설치 및 예제 실행

- 웹 브라우저를 통해 확인





2. Express 설치 및 예제 실행

- app.js에서 url을 매핑하는 부분 확인

```
8 var routes = require('./routes/index');
9 var users = require('./routes/users');
10
25 app.use('/', routes);
26 app.use('/users', users);
--
```



3. ejs

- Embedded JavaScript(ejs) : html에 자바스크립트가 내장된 파일

ejs

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>document.write()</title>
  </head>
  <body>
    <h2>document.write()의 예제</h2>
    <% for (var i=0; i<5; i++) { %>
      예제 실행 <%= i+1 %>번째.<br>
    <% } %>
  </body>
</html>
```

html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>document.write()</title>
  </head>
  <body>
    <h2>document.write()의 예제</h2>
    <script language="javascript">
      for (var i = 0; i < 5; i++) {
        document.write("예제 실행" + (i+1) + "번째<br>");
      }
    </script>
  </body>
</html>
```



4. 회원가입 폼 만들기 및 nodemon 사용

- 아이디, 비밀번호, 이름, 이메일, 전화번호, 주소, 직업, 성별, 생일을 입력받음
 - 모든 값에 공백 검사
 - 가입 버튼을 클릭 시 입력한 내용을 json 객체 형태로 표시
- 순서
 - 1) express 프로젝트 생성(ejs 템플릿엔진으로 생성)
 - 2) routes 폴더에 로직 담당하는 js 파일 생성
 - 3) app.js에 소스를 적어 2에서 생성한 js파일 연결
 - 4) views 폴더에 ejs 파일 생성
 - 5) 2에서 만든 js파일에 소스를 적어 4번에서 만든 ejs 파일 연결



4. 회원가입 폼 만들기 및 nodemon 사용

- > express --ejs joinForm # 프로젝트 생성

```
C:\Windows\system32\cmd.exe

C:\myNode>cd joinForm

C:\myNode\joinForm>npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
added 53 packages from 38 contributors and audited 141 packages in 1.162s
found 0 vulnerabilities

C:\myNode\joinForm>
```




4. 회원가입 폼 만들기 및 nodemon 사용

- >cd joinForm # joinForm 폴더로 이동
- >npm install # 모듈 설치

C:\Windows\system32\cmd.exe

```
C:\myNode>cd joinForm
```

```
C:\myNode\joinForm>npm install
```

```
npm notice created a lockfile as package-lock.json. You should commit this file.  
added 53 packages from 38 contributors and audited 141 packages in 1.162s  
found 0 vulnerabilities
```

```
C:\myNode\joinForm>
```



4. 회원가입 폼 만들기 및 nodemon 사용

- routes 폴더에 joinForm.js 파일 생성
- 같은 폴더에 생성된 users.js에 res.send 부분만 변경

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET users listing. */
5 router.get('/', function(req, res, next) {
6   res.send('joinForm.js check');
7 });
8
9 module.exports = router;
```



4. 회원가입 폼 만들기 및 nodemon 사용

- app.js에 소스를 추가하여 joinForm.js 파일 연결

```
8  var routes = require('./routes/index');
9  var users = require('./routes/users');
10 var join = require('./routes/joinForm');

26 app.use('/', routes);
27 app.use('/users', users);
28 app.use('/join', join);
```



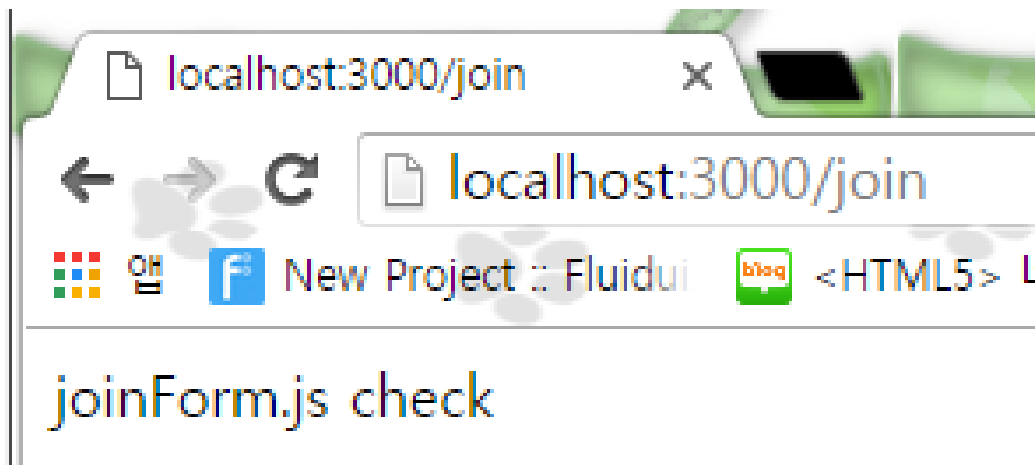
4. 회원가입 폼 만들기 및 nodemon 사용

- 서버를 시작하여 웹 브라우저를 통해 동작 확인

npm

```
D:\Users\User\Documents\myNode\joinForm>npm start
```

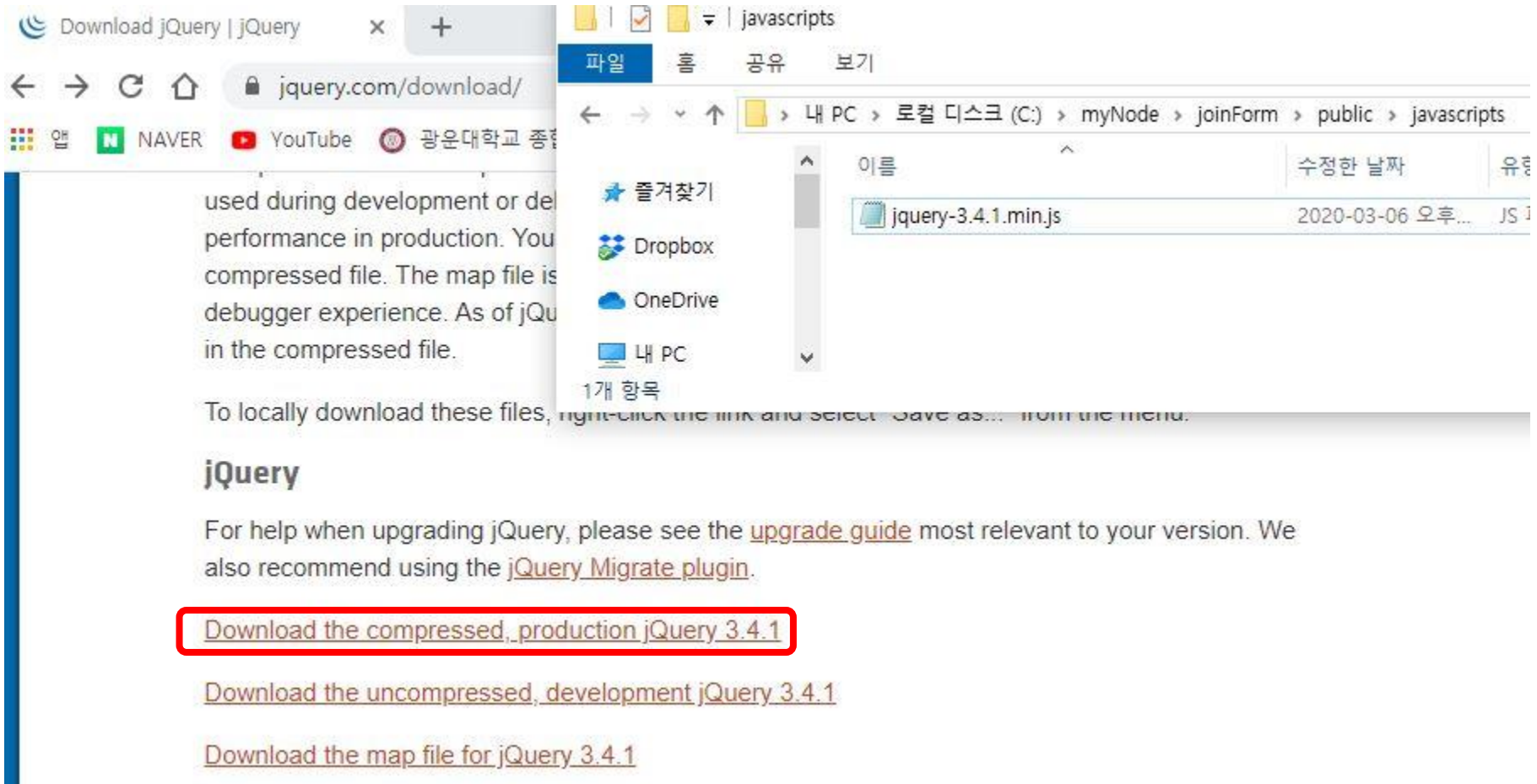
```
> joinForm@0.0.0 start D:\Users\User\Documents\myNode\joinForm
> node ./bin/www
```





4. 회원가입 폼 만들기 및 nodemon 사용

- jQuery 다운로드 후 public\javascripts 폴더에 저장



The screenshot shows a web browser window displaying the jQuery download page (jquery.com/download/). The page content includes instructions for downloading jQuery and a list of download links. The link "Download the compressed, production jQuery 3.4.1" is highlighted with a red rectangle. To the right, a Windows File Explorer window is open, showing the path "내 PC > 로컬 디스크 (C:) > myNode > joinForm > public > javascripts". The file "jquery-3.4.1.min.js" is visible in the file list, with a modification date of "2020-03-06 오후..." and a file type of "JS".

used during development or deployment. It is not optimized for performance in production. You can use the compressed file. The map file is not included in the compressed file. As of jQuery 3.4.1, the map file is not included in the compressed file.

To locally download these files, right-click the link and select "Save as..." from the menu.

jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relevant to your version. We also recommend using the [jQuery Migrate plugin](#).

[Download the compressed, production jQuery 3.4.1](#)

[Download the uncompressed, development jQuery 3.4.1](#)

[Download the map file for jQuery 3.4.1](#)



4. 회원가입 폼 만들기 및 nodemon 사용

- 다음 소스코드를 따라서 작성하여 views 폴더에 joinForm.ejs 템플릿 생성

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title><%= title %></title>
5     <meta charset='utf-8'>
6     <link rel='stylesheet' href='/stylesheets/style.css' />
7     <script src="/javascripts/jquery-3.4.1.min.js"></script>
8     <script>
9       $(function(){
10
11         $("#btnJoin").click(
12           function(){
13
14             if($("#id").val() == "")
15             {
16               alert("아이디를 꼭 입력 하세요");
17               $("#id").focus();
18               return;
19             }
20             if($("#passwd").val() == "")
21             {
22               alert("비밀번호를 꼭 입력 하세요");
23               $("#passwd").focus();
24               return;
25             }
26             if($("#name").val() == "")
27             {
28               alert("이름을 꼭 입력 하세요");
29               $("#name").focus();
30               return;
31             }
32             if($("#email").val() == "")
33             {
34               alert("이메일을 꼭 입력 하세요");
35               $("#email").focus();
36               return;
37             }
38           }
39         )
40       }
41     </script>
42   </head>
43   <body>
44     <div>
45       <div>
46         <div>
47           <div>
48             <div>
49               <div>
50                 <div>
51                   <div>
52                     <div>
53                       <div>
54                         <div>
55                           <div>
56                             <div>
57                               <div>
58                                 <div>
59                                   <div>
60                                     <div>
61                                       <div>
62                                         <div>
63                                           <div>
64                                             <div>
65                                             </div>
66                                           </div>
67                                         </div>
68                                       </div>
69                                     </div>
70                                   </div>
71                                 </div>
72                               </div>
73                             </div>
74                           </div>
75                         </div>
76                       </div>
77                     </div>
78                   </div>
79                 </div>
80               </div>
81             </div>
82           </div>
83         </div>
84       </div>
85     </div>
86   </body>
87 </html>
```



4. 회원가입 폼 만들기 및 nodemon 사용

```
38 if($("#tel2").val() == "")
39 {
40     alert("전화번호 두번째 자리를 확인해 주세요");
41     $("#tel2").focus();
42     return;
43 }
44 if($("#tel3").val() == "")
45 {
46     alert("전화번호 세번째 자리를 확인해 주세요");
47     $("#tel3").focus();
48     return;
49 }
50
51 var tel= $("#tel1").val() + '-' + $("#tel2").val() + '-' + $("#tel3").val();
52 $("#tel").val(tel);
53
54 if($("#address").val() == "")
55 {
56     alert("주소를 꼭 입력 하세요");
57     $("#address").focus();
58     return;
59 }
60 if($("#birth").val() == "")
61 {
62     alert("생일을 꼭 입력 하세요");
63     $("#birth").focus();
64     return;
65 }
66 if($("#birth").val().length != 8)
67 {
68     alert("생일은 8글자로 입력해 주세요");
69     $("#birth").focus();
70     return;
71 }
72 if(isNaN($("#birth").val()))
73 {
74     alert("생일은 숫자만 입력해 주세요");
75     $("#birth").focus();
76     return;
77 }
```

< 소스코드 2/4 >



4. 회원가입 폼 만들기 및 nodemon 사용

```
78         $("#joinForm").submit();
79     });
80 });
81 </script>
82 </head>
83 <body>
84     <h1><%= title %></h1>
85     <form id="joinForm" action="/join" method="post">
86         <table width="400" border="1">
87             <tr>
88                 <th>아이디</th>
89                 <td><input type="text" name="id" id="id" size="12" maxLength="12"></td>
90             </tr>
91             <tr>
92                 <th>비밀번호</th>
93                 <td><input type="password" name="passwd" id="passwd" size="12" maxLength="12"></td>
94             </tr>
95             <tr>
96                 <th>이름</th>
97                 <td><input type="text" name="name" id="name" size="10" maxLength="10"></td>
98             </tr>
99             <tr>
100                 <th>이메일</th>
101                 <td><input type="email" name="email" id="email" size="30" maxLength="50"></td>
102             </tr>
103             <tr>
104                 <th>전화번호</th>
105                 <td>
106                     <input type="hidden" name="tel" id="tel">
107                     <select name="tel1" id="tel1">
108                         <option value="010">010</option>
109                         <option value="011">011</option>
110                         <option value="016">016</option>
111                         <option value="017">017</option>
112                         <option value="018">018</option>
113                         <option value="019">019</option>
114                     </select>
115                     -
116                     <input type="text" name="tel2" id="tel2" size="4" maxLength="4">
117                     -
118                     <input type="text" name="tel3" id="tel3" size="4" maxLength="4">
119                 </td>
120             </tr>
```




4. 회원가입 폼 만들기 및 nodemon 사용

```
121     <tr>
122         <th>주소</th>
123         <td><input type="text" id="address" size="30" maxLength="50"></td>
124     </tr>
125     <tr>
126         <th>직업</th>
127         <td><select name="job" id="job">
128             <option value="학생">학생</option>
129             <option value="직장인">직장인</option>
130             <option value="주부">주부</option>
131         </select></td>
132     </tr>
133     <tr>
134         <th>성별</th>
135         <td>
136             <input type="radio" name="gender" value="남" checked="checked">남
137             <input type="radio" name="gender" value="여">여
138         </td>
139     </tr>
140     <tr>
141         <th>생일</th>
142         <td><input type="text" name="birth" id="birth" size="8" maxLength="*YYYYMMDD"></td>
143     </tr>
144     <tr>
145         <td colspan="2">
146             <input type="button" id="btnJoin" name="join" value="가입">
147             <input type="reset" name="cancel" value="취소">
148         </td>
149     </tr>
150 </table>
151 </form>
152 </body>
153 </html>
```

< 소스코드 4/4 >



4. 회원가입 폼 만들기 및 nodemon 사용

- Routes폴더의 joinForm.js의 res.send 부분을 다음과 같이 수정

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET users listing. */
5 router.get('/', function(req, res, next) {
6   res.render('joinForm', { title: 'Join Form!' });
7 });
8
9 module.exports = router;
10
```



4. 회원가입 폼 만들기 및 nodemon 사용

- 서버 재시작 후 웹 브라우저를 통해 결과 확인

Join Form!

localhost:3000/join

아이디	<input type="text"/>
비밀번호	<input type="password"/>
이름	<input type="text"/>
이메일	<input type="text"/>
전화번호	010 ▾ - <input type="text"/> - <input type="text"/>
주소	<input type="text"/>
직업	학생 ▾
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여
생일	<input type="text"/> *YYYYMMDD
<input type="button" value="가입"/> <input type="button" value="취소"/>	



4. 회원가입 폼 만들기 및 nodemon 사용

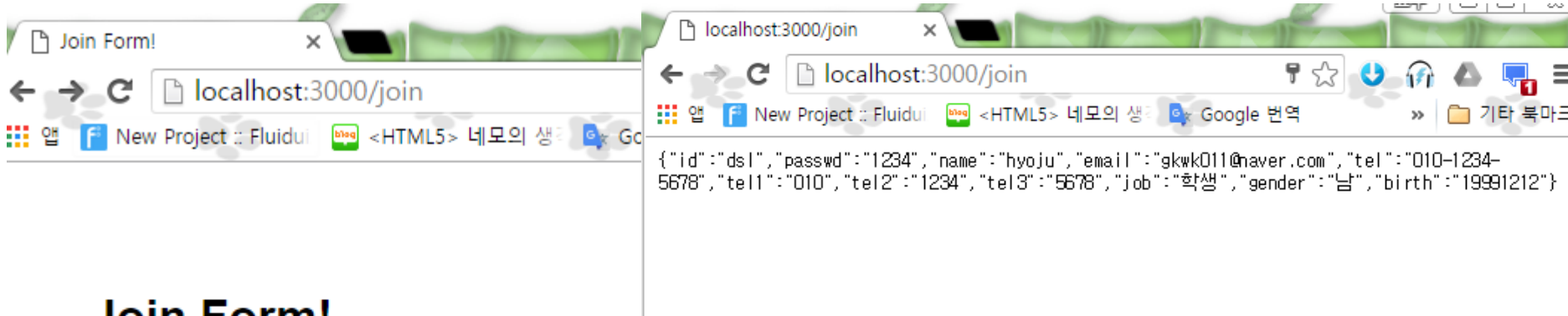
- 사용자 입력값을 JSON으로 출력하기 위해 joinForm.js에 다음 소스코드 추가

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET users listing. */
5 router.get('/', function(req, res, next) {
6   res.render('joinForm', { title: 'Join Form!' });
7 });
8
9 router.post('/', function(req, res, next){
10   console.log('req.body : '+req.body);
11   res.json(req.body);
12 });
13
14 module.exports = router;
```



4. 회원가입 폼 만들기 및 nodemon 사용

- 서버 재시작 후 웹 브라우저를 통해 결과 확인



Join Form!

아이디	<input type="text" value="dsl"/>		
비밀번호	<input type="password" value="...."/>		
이름	<input type="text" value="hyoju"/>		
이메일	<input type="text" value="gkwk011@naver.com"/>		
전화번호	010 ▼	- 1234	- 5678
주소	<input type="text" value="광운대학교"/>		
직업	<input type="text" value="학생"/>		
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여		
생일	<input type="text" value="19991212"/>	*YYYYMMDD	
<input type="button" value="가입"/> <input type="button" value="취소"/>			



4. 회원가입 폼 만들기 및 nodemon 사용

- nodemon
 - 소스코드를 수정할 때 마다 서버를 재시작하지 않아도 됨
 - 수정된 소스가 콘솔창에 나타남
- > npm install -g nodemon

```
C:\Windows\system32\cmd.exe
C:\myNode\joinForm>npm install -g nodemon
C:\Users\micky\AppData\Roaming\npm\nodemon -> C:\Users\micky\AppData\Roaming\npm\node_modules\nodemon\bin\nodemon.js

> nodemon@2.0.2 postinstall C:\Users\micky\AppData\Roaming\npm\node_modules\nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.1.2 (node_modules\nodemon\node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current:
{"os":"win32","arch":"x64"})

+ nodemon@2.0.2
added 105 packages from 49 contributors in 3.854s

C:\myNode\joinForm>
```



4. 회원가입 폼 만들기 및 nodemon 사용

- package.json 수정

```
1 {  
2   "name": "joinform",  
3   "version": "0.0.0",  
4   "private": true,  
5   "scripts": {  
6     "start": "nodemon ./bin/www"  
7   },  
8   "dependencies": {  
9     "cookie-parser": "~1.4.4",  
10    "debug": "~2.6.9",  
11    "ejs": "~2.6.1",  
12    "express": "~4.16.1",  
13    "http-errors": "~1.6.3",  
14    "morgan": "~1.9.1"  
15  }  
16 }  
17
```



5. Database(MySQL)

- MySQL 설치
 - 다음 URL을 웹 브라우저에 입력함
 - <https://dev.mysql.com/downloads/installer>

General Availability (GA) Releases Archives ⓘ

MySQL Installer 8.0.19

Select Operating System:

Microsoft Windows ▼

Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.19.0.msi)	8.0.19	18.6M	Download
MD5: 32043776cb2239db45fddaa86dc0ad61 Signature			
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.19.0.msi)	8.0.19	398.9M	Download
MD5: 1a882015da7fb93f20c4717e63b6817c Signature			

! We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.



5. Database(MySQL)

Begin Your Download

mysql-installer-web-community-8.0.11.0.msi

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system
- Comment in the MySQL Documentation

Login »

using my Oracle Web account

Sign Up »

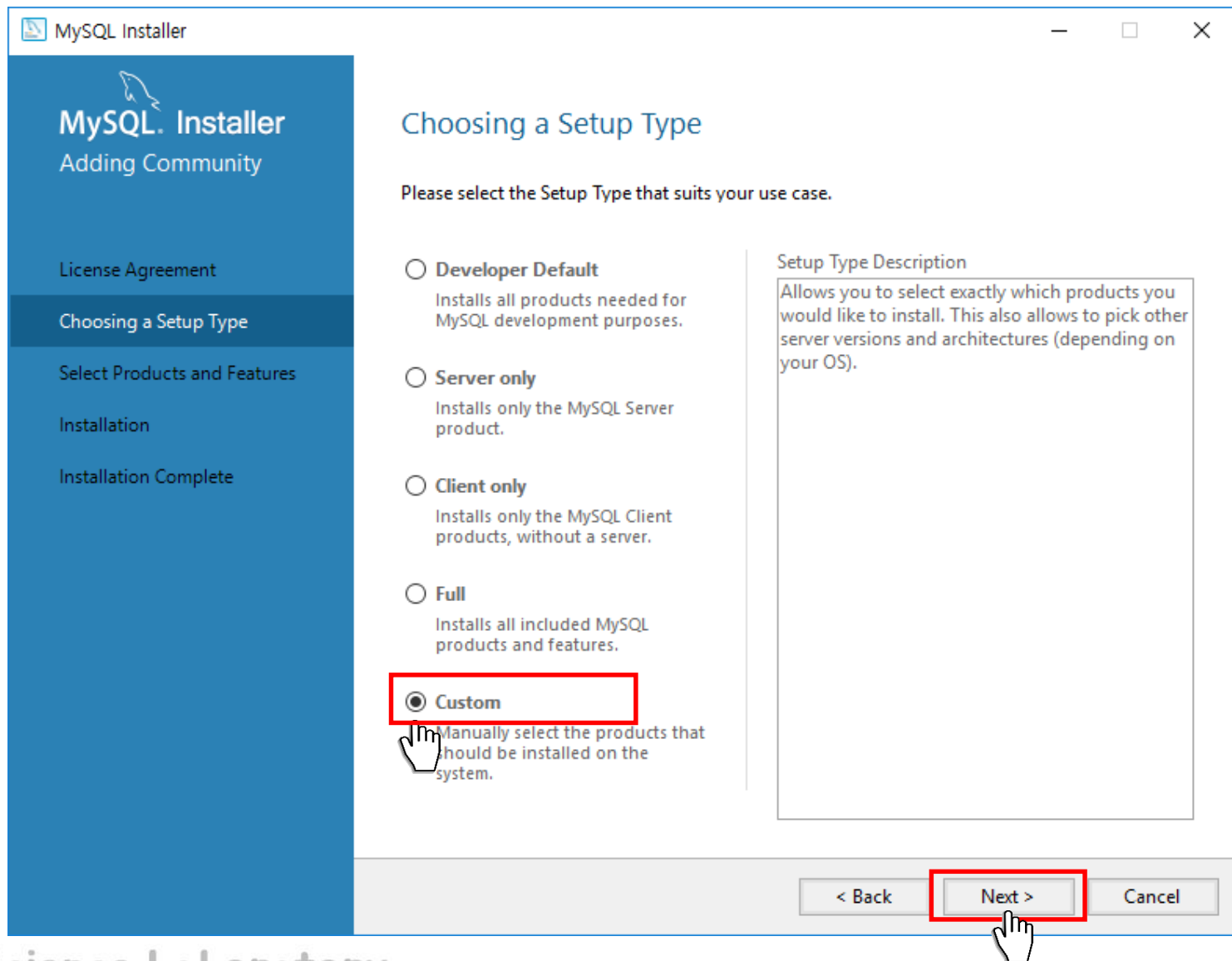
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

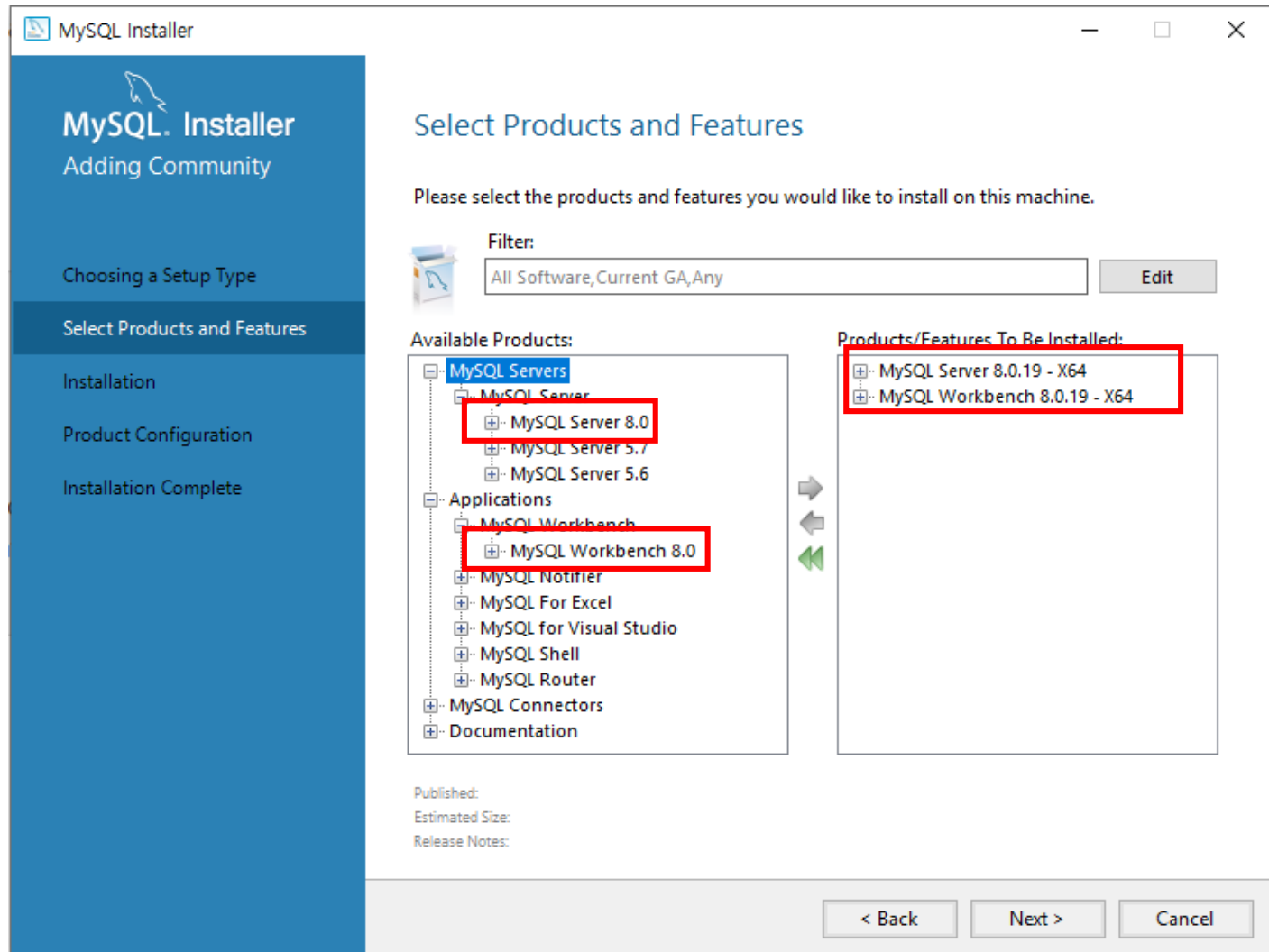


5. Database(MySQL)



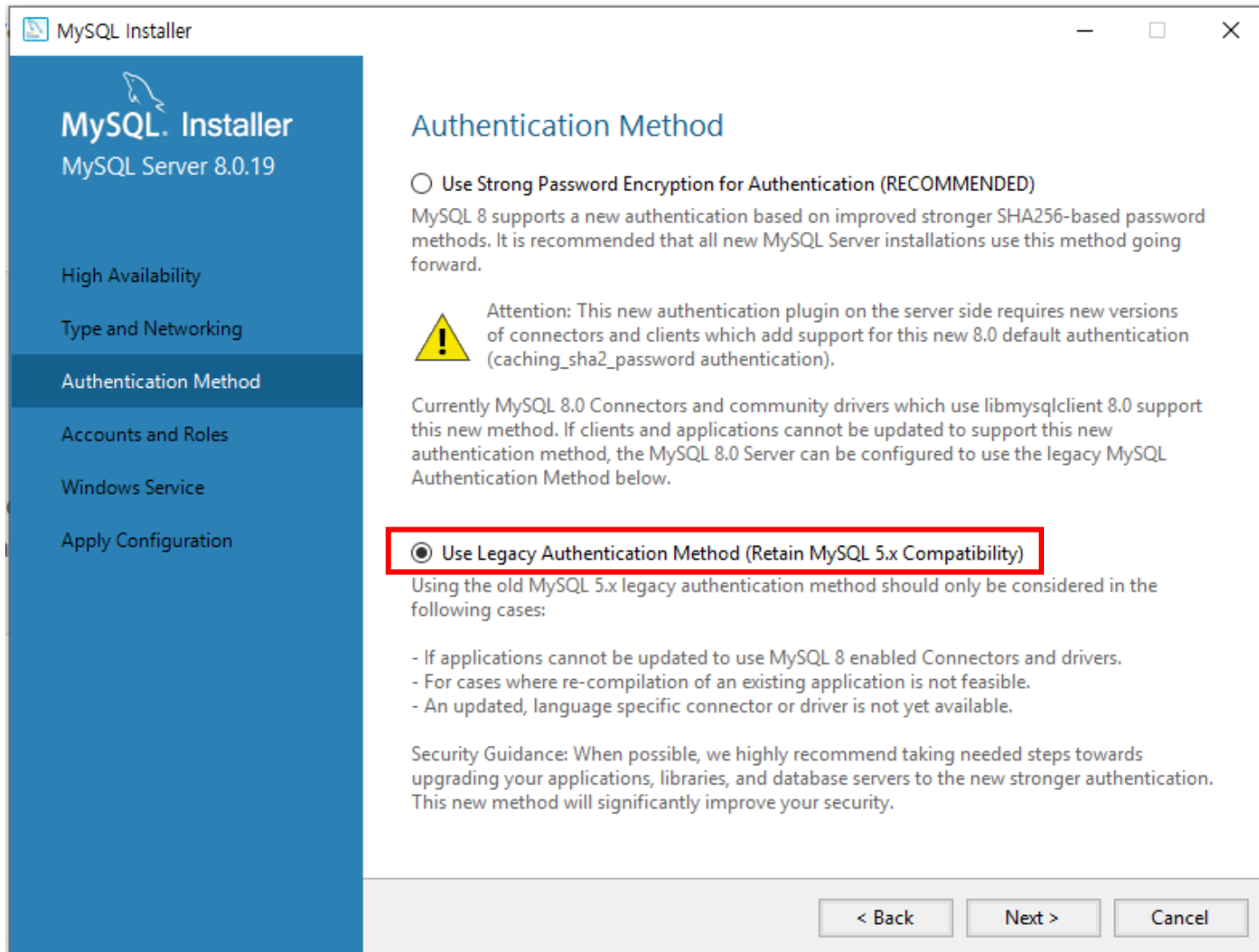


5. Database(MySQL)





5. Database(MySQL)





5. Database(MySQL)

MySQL Installer

MySQL Server 8.0.11

- Group Replication
- Type and Networking
- Authentication Method
- Accounts and Roles**
- Windows Service
- Plugins and Extensions
- Apply Configuration

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

Repeat Password:

패스워드 설정

MySQL User Accounts

Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

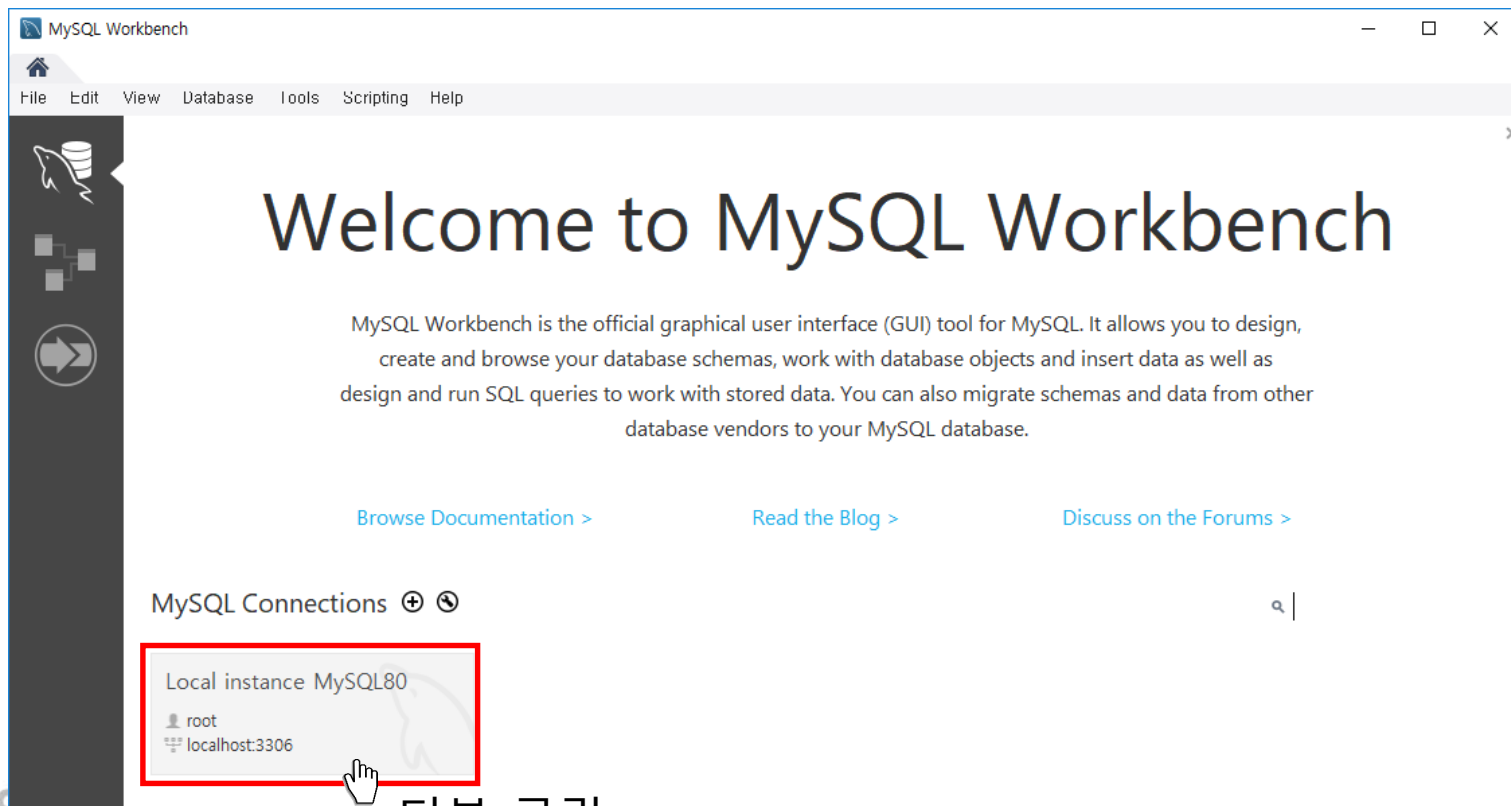
MySQL Username	Host	User Role
----------------	------	-----------



5. Database(MySQL)

■ MySQL Workbench

- 데이터베이스 설계 및 구현을 위한 GUI 프로그램
- SQL 편집, 데이터베이스 모델링, 데이터베이스 관리, 데이터베이스 마이그레이션 등의 기능을 제공함



더블 클릭



5. Database(MySQL)

- MySQL Workbench 실행

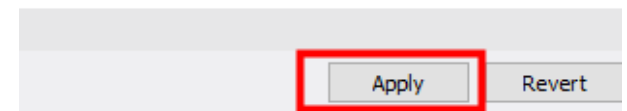
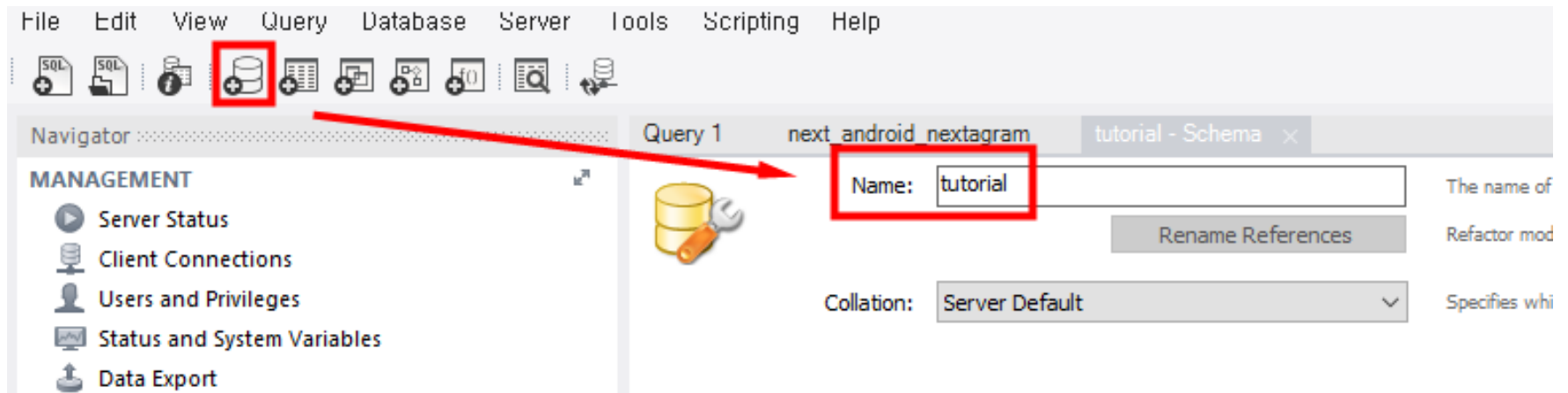


앞에서 설정한 암호입력



5. Database(MySQL)

■ Tutorial 데이터베이스 생성





5. Database(MySQL)

Apply SQL Script to Database

Review SQL Script
Apply SQL Script

Review the SQL Script to be Applied on the Database

Online DDL

Algorithm: Lock Type:

```
1 CREATE SCHEMA `tutorial` ;
2
```

Back **Apply** Cancel



5. Database(MySQL)

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

SQL script was successfully applied to the database.

Show Logs

Back

Finish

Cancel



5. Database(MySQL)

■ SQL 편집기 사용법

- 단일 명령문 실행
 - 실행하고자 하는 명령문으로 커서를 옮긴 다음 “Ctrl키 + Enter키” 혹은 번개+커서 아이콘 클릭
- 다중 명령문 실행
 - 실행하고자 하는 명령문들을 선택한 다음 “Ctrl키 + Shift키 + Enter키” 혹은 번개 아이콘 클릭
 - 명령문들을 선택하지 않으면 현재 편집창의 모든 명령문을 실행함
- 모든 명령문은 세미콜론(;)으로 끝나야 함





5. Database(MySQL)

- SELECT 문은 SQL 질의(또는 쿼리)라고도 부름
- 기본적인 SELECT 문은 다음과 같은 구조를 가짐

```
SELECT [DISTINCT] { * | column [alias], ...}  
FROM table  
[ WHERE condition ]  
[ GROUP BY group_by_expression ]  
[ HAVING group_condition ]  
[ ORDER BY column ];
```

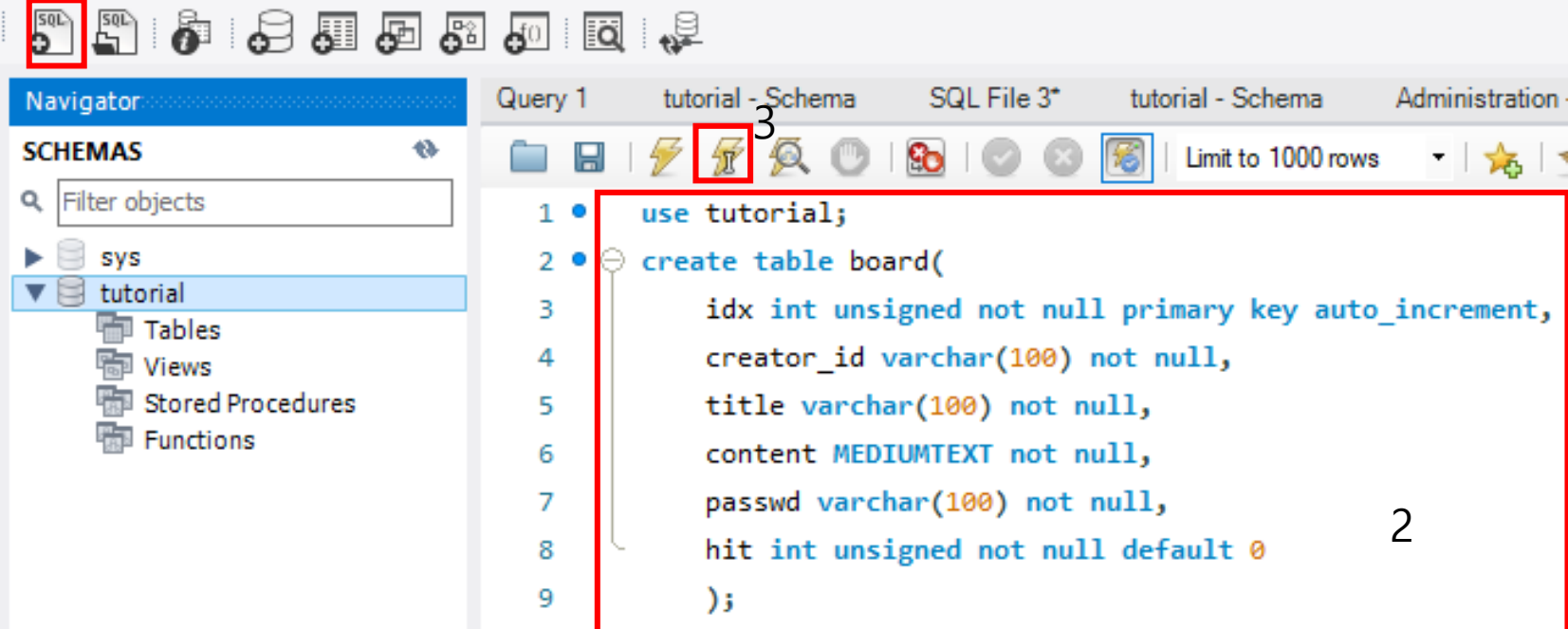
- SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY 각각을 절(clause)이라고 부르며 순서대로 나와야 함
- '[']' 는 생략 가능하다는 것을 의미하고, '|' 는 OR를 의미함
- condition은 TRUE 또는 FALSE 값을 가지는 조건식을 의미함
- 컬럼명 앞에 테이블명을 명시할 수 있음
 - 예: EMP 테이블의 NAME 컬럼을 EMP.NAME으로 표시



5. Database(MySQL) 연동

- board 테이블 생성

1



3

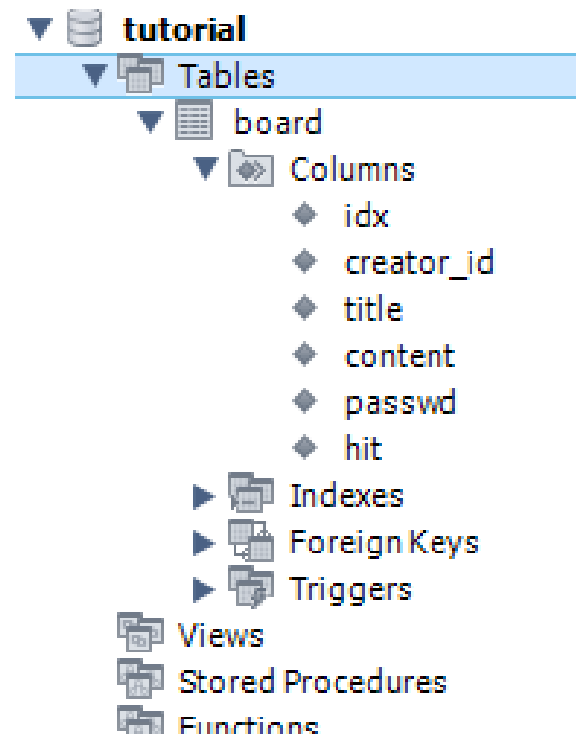
```
1 • use tutorial;
2 • create table board(
3     idx int unsigned not null primary key auto_increment,
4     creator_id varchar(100) not null,
5     title varchar(100) not null,
6     content MEDIUMTEXT not null,
7     passwd varchar(100) not null,
8     hit int unsigned not null default 0
9 );
```

2



5. Database(MySQL) 연동

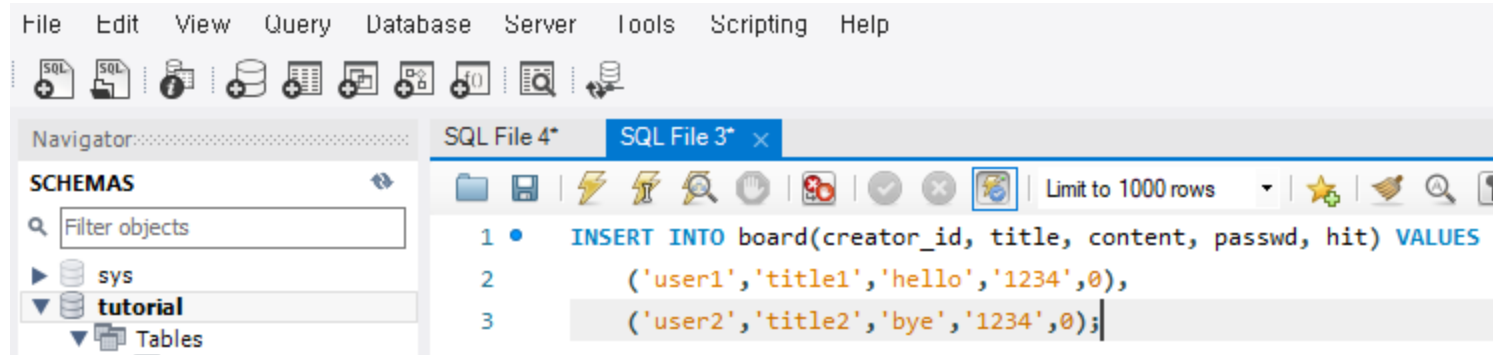
- Tables 우클릭 -> Refresh All 하여, 생성한 테이블 확인





5. Database(MySQL) 연동

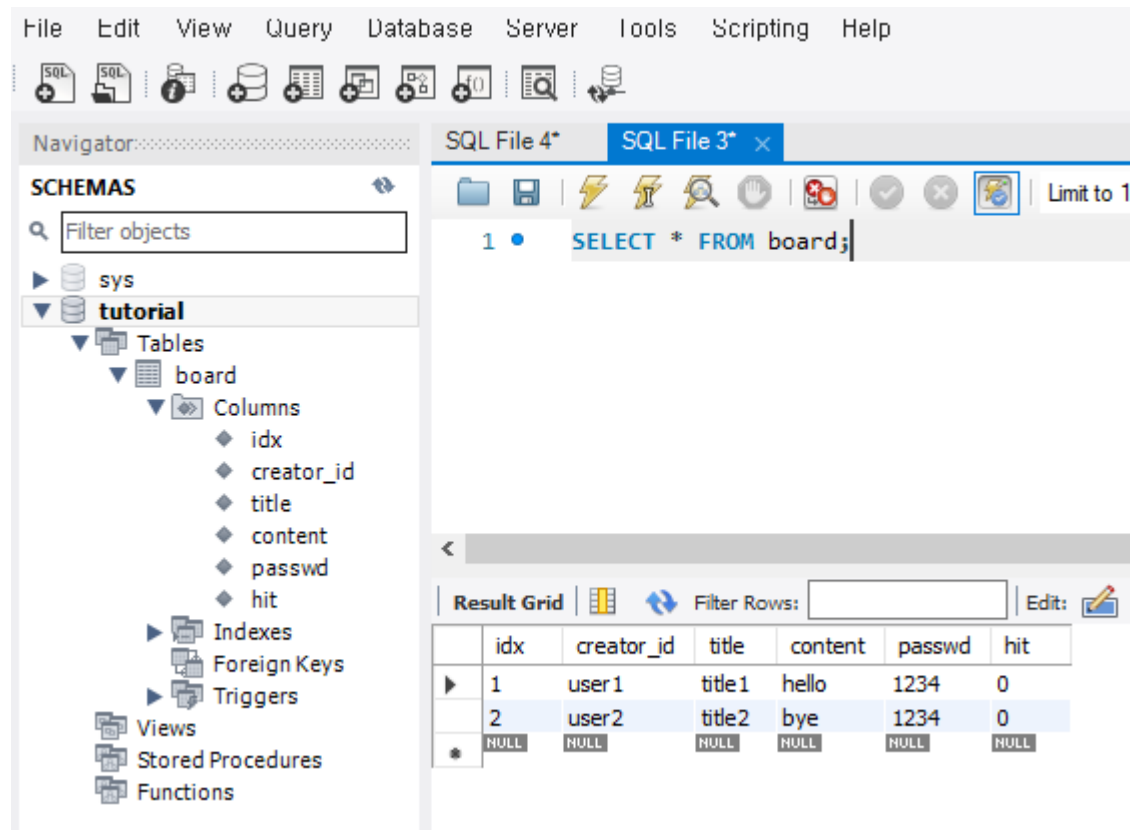
- Test 데이터 입력





5. Database(MySQL) 연동

- Test 데이터 확인



The screenshot shows a MySQL IDE interface. On the left, the 'Navigator' pane displays a tree structure of databases and tables. The 'tutorial' database is expanded, showing a 'board' table with columns: idx, creator_id, title, content, passwd, and hit. The main editor area shows a SQL query: `SELECT * FROM board;`. Below the query, the 'Result Grid' displays the data returned by the query. The grid has 7 columns: idx, creator_id, title, content, passwd, and hit. The first two rows of data are visible, showing entries for 'user1' and 'user2'.

	idx	creator_id	title	content	passwd	hit
▶	1	user1	title1	hello	1234	0
	2	user2	title2	bye	1234	0
*	NULL	NULL	NULL	NULL	NULL	NULL



5. Database(MySQL) 연동

- Node.js 에 mysql 설치 > npm install mysql

C:\Windows\system32\cmd.exe

```
C:\myNode\joinForm>npm install mysql
+ mysql@2.18.1
added 9 packages from 14 contributors and audited 154 packages in 0.869s
found 0 vulnerabilities

C:\myNode\joinForm>
```



6. 게시판 구축

- URL 설계(GET Method)
 - 게시판 글 목록 표시 화면
 - <http://localhost/board/list/?page=1>
 - 게시판 글 조회 화면
 - <http://localhost/board/read/?idx=1>
 - 게시판 글 쓰기 화면
 - <http://localhost/board/write>
 - 게시판 글 수정 화면
 - <http://localhost/board/modify>
- URL 설계(POST Method)
 - 게시판 글 쓰기 액션
 - <http://localhost/board/write>
 - 게시판 글 수정 액션
 - <http://localhost/board/modify>



6. 게시판 구축

- app.js 소스 추가

```
8 var routes = require('./routes/index');
9 var users = require('./routes/users');
10 var join = require('./routes/joinForm');
11 var board = require('./routes/board');

27 app.use('/', routes);
28 app.use('/users', users);
29 app.use('/join', join);
30 app.use('/board', board);
```



6. 게시판 구축

- routes#board.js 생성
- users.js, index.js를
이용해 작성

```
1 var express = require('express');
2 var router = express.Router();
3
4 // MySQL 로드
5 var mysql = require('mysql');
6 var pool = mysql.createPool({
7   connectionLimit: 5,
8   host: 'localhost',
9   user: 'root',
10  database: ' ',
11  password: ' '
12 });
13
14 /* GET users listing. */
15 router.get('/', function(req, res, next) {
16   // 그냥 board/ 로 접속할 경우 전체 목록 표시로 리다이렉팅
17   res.redirect('/board/list/1');
18 });
19
20 router.get('/list/:page', function(req,res,next){
21
22   pool.getConnection(function (err, connection) {
23     // Use the connection
24     var sqlForSelectList = "SELECT idx, creator_id, title, hit FROM board";
25     connection.query(sqlForSelectList, function (err, rows) {
26       if (err) console.error("err : " + err);
27       console.log("rows : " + JSON.stringify(rows));
28
29       res.render('list', {title: ' 게시판 전체 글 조회', rows: rows});
30       connection.release();
31
32       // Don't use the connection here, it has been returned to the pool.
33     });
34   });
35 });
36
37 module.exports = router;
```



6. 게시판 구축

■ views\list.ejs 생성

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title><%= title %></title>
5   <link rel='stylesheet' href='/stylesheets/style.css' />
6 </head>
7 <body>
8 <h1><%= title %></h1>
9
10 <a href="/board/write">글쓰기로 이동</a>
11 <br>
12 <br>
13
14 <table border="1">
15   <tr>
16     <td>번호</td>
17     <td>작성자</td>
18     <td>제목</td>
19     <td>조회수</td>
20   </tr>
21
```

< 소스코드 1/2 >



6. 게시판 구축

■ views\list.ejs 생성

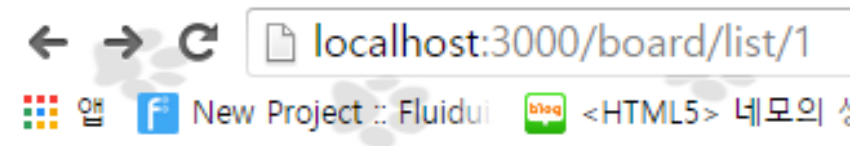
```
22 <%  
23     for(var i=0; i<rows.length; i++)  
24     {  
25         var oneItem = rows[i];  
26     %>  
27     <tr>  
28         <td><%=oneItem.idx%></td>  
29         <td><%=oneItem.creator_id%></td>  
30         <td><%=oneItem.title%></td>  
31         <td><%=oneItem.hit%></td>  
32     </tr>  
33  
34 <%  
35     }  
36 %>  
37 </table>  
38  
39 </body>  
40 </html>
```

< 소스코드 2/2 >



6. 게시판 구축

■ 결과 확인



게시판 전체 글 조회

[글쓰기로 이동](#)

번호	작성자	제목	조회수
1	user1	title1	0
2	user2	title2	0



6. 게시판 구축

- 글쓰기
- views/write.ejs

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/javascripts/jquery-1.12.2.min.js"></script>
5   <title><%= title %></title>
6   <link rel='stylesheet' href='/stylesheets/style.css' />
7 </head>
8 <body>
9 <h1><%= title %></h1>
10
11 <form action="/board/write" method="post" onsubmit="return onWriteSubmit()">
12   <table border="1">
13     <tr>
14       <td>작성자</td>
15       <td><input type="text" name="creator_id" id="creator_id" required/></td>
16     </tr>
17     <tr>
18       <td>제목</td>
19       <td><input type="text" name="title" id="title" required/></td>
20     </tr>
21     <tr>
22       <td>내용</td>
23       <td><textarea name="content" id="content" cols="30" rows="10" required></textarea></td>
24     </tr>
25     <tr>
26       <td>패스워드</td>
27       <td><input type="password" name="passwd" id="passwd" required/></td>
28     </tr>
29     <tr>
30       <td colspan="2">
31         <button type="submit">글쓰기</button>
32       </td>
33     </tr>
34   </table>
35 </form>
36
```

< 소스코드 1/3 >



6. 게시판 구축

■ 글쓰기

```
37 <script>
38     function onWriteSubmit()
39     {
40         if ( $("#creator_id").val().trim() == "" )
41         {
42             var message = "아이디를 입력해 주세요";
43             $("#creator_id").val("");
44             $("#creator_id").focus();
45             alert(message);
46             return false;
47         }
48
49         if ( $("#title").val().trim() == "" )
50         {
51             var message = "제목을 입력해 주세요";
52             $("#title").val("");
53             $("#title").focus();
54             alert(message);
55             return false;
56         }
57
58         if ( $("#content").val().trim() == "" )
59         {
60             var message = "본문 내용을 입력해 주세요";
61             $("#content").val("");
62             $("#content").focus();
63             alert(message);
64             return false;
65         }
66     }
```

< 소스코드 2/3 >



6. 게시판 구축

■ 글쓰기

```
66
67     if ( $("#passwd").val().trim() == "" )
68     {
69         var message = "패스워드를 입력해 주세요";
70         $("#passwd").val("");
71         $("#passwd").focus();
72         alert(message);
73         return false;
74     }
75 }
76 </script>
77
78 </body>
79 </html>
```

< 소스코드 3/3 >

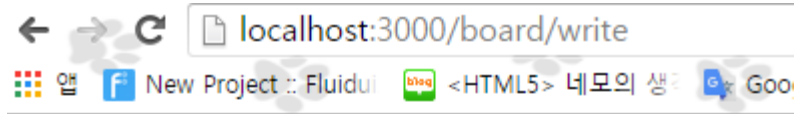
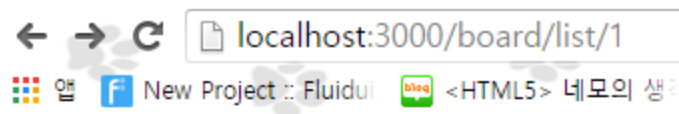
■ board.js에 소스 추가

```
37 // 글쓰기 화면 표시 GET
38 router.get('/write', function(req,res,next){
39     res.render('write',{title : "게시판 글 쓰기"});
40 });
```



6. 게시판 구축

글쓰기 결과 확인



게시판 전체 글 조회

[글쓰기로 이동](#)

번호	작성자	제목	조회수
1	user1	title1	0
2	user2	title2	0

게시판 글 쓰기

작성자	<input type="text"/>
제목	<input type="text"/>
내용	<div><div></div></div>
패스워드	<input type="password"/>
<input type="button" value="글쓰기"/>	



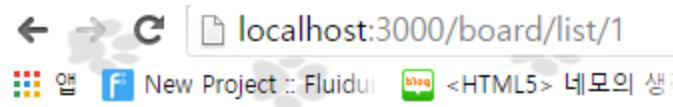
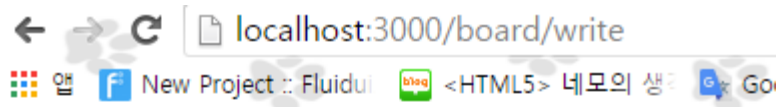
6. 게시판 구축

■ board.js에 POST 추가

```
42 // 글쓰기 로직 처리 POST
43 router.post('/write', function(req,res,next){
44
45     var creator_id = req.body.creator_id;
46     var title = req.body.title;
47     var content = req.body.content;
48     var passwd = req.body.passwd;
49     var datas = [creator_id,title,content,passwd];
50
51     pool.getConnection(function (err, connection)
52     {
53         // Use the connection
54         var sqlForInsertBoard = "insert into board(creator_id, title, content, passwd) values(?,?,?,?)";
55         connection.query(sqlForInsertBoard,datas, function (err, rows) {
56             if (err) console.error("err : " + err);
57             console.log("rows : " + JSON.stringify(rows));
58
59             res.redirect('/board');
60             connection.release();
61
62             // Don't use the connection here, it has been returned to the pool.
63         });
64     });
65
66 });
67
68 module.exports = router;
```

6. 게시판 구축

■ 결과 확인



게시판 글 쓰기

작성자	user3
제목	글쓰기테스트
내용	테스트 테스트
패스워드
글쓰기	

게시판 전체 글 조회

[글쓰기로 이동](#)

번호	작성자	제목	조회수
1	user1	title1	0
2	user2	title2	0
3	user3	글쓰기테스트	0

6. 게시판 구축

- 조회
- viewsWread.ejs 생성

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/javascripts/jquery-3.4.1.min.js"></script>
5   <title><%= title %></title>
6   <link rel='stylesheet' href='/stylesheets/style.css' />
7 </head>
8 <body>
9   <h1><%= title %></h1>
10
11   <form action="/board/update" method="get">
12     <table border="1">
13       <input type="hidden" name="idx" value="<%=row.idx%>" />
14       <tr>
15         <td>작성자</td>
16         <td><%=row.creator_id%></td>
17       </tr>
18       <tr>
19         <td>제목</td>
20         <td><%=row.title%></td>
21       </tr>
```

< 소스코드 1/2 >



6. 게시판 구축

■ 조회

```
22         <tr>
23             <td>내용</td>
24             <td><%=row.content%></td>|
25         </tr>
26         <tr>
27             <td>조회수</td>
28             <td><%=row.hit%></td>
29         </tr>
30         <tr>
31             <td colspan="2">
32                 <button type="submit">글 수정</button>
33                 <a href="/board">리스트로 돌아가기</a>
34             </td>
35         </tr>
36     </table>
37 </form>
38 </body>
39 </html>
```

< 소스코드 2/2 >



6. 게시판 구축

■ 조회

- board.js 파일에 아래 소스 추가

```
68 // 글조회 로직 처리 GET
69 router.get('/read/:idx',function(req,res,next)
70 {
71     var idx = req.params.idx;
72
73     pool.getConnection(function(err,connection)
74     {
75         var sql = "select idx, creator_id, title, content, hit from board where idx=?";
76         connection.query(sql,[idx], function(err,row)
77         {
78             if(err) console.error(err);
79             console.log("17# 글 조회 결과 확인 : ",row);
80             res.render('read', {title:"글 조회", row:row[0]});
81             connection.release();
82         });
83     });
84 });
85 });
```




6. 게시판 구축

■ 조회

- list.ejs 파일에 글 조회 링크 추가

```
22 <%
23     for(var i=0; i<rows.length; i++)
24     {
25         var oneItem = rows[i];
26     %>
27     <tr>
28         <td><%=oneItem.idx%></td>
29         <td><%=oneItem.creator id%></td>
30         <td><a href="/board/read/<%=oneItem.idx%>"><%=oneItem.title%></a></td>
31         <td><%=oneItem.hit%></td>
32     </tr>
33
34 <%
35     }
36 %>
37 </table>
38
39 </body>
40 </html>
```



6. 게시판 구축

- 업데이트 조회
 - viewsWupdate.ejs 생성

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/javascripts/jquery-3.4.1.min.js"></script>
5   <title><%= title %></title>
6   <link rel='stylesheet' href='/stylesheets/style.css' />
7 </head>
8 <body>
9   <h1><%= title %></h1>
10
11   <form action="/board/update" method="post" onsubmit="return onWriteSubmit()">
12     <table border="1">
13       <input type="hidden" name="idx" value="<%=row.idx%>" />
14       <tr>
15         <td>작성자</td>
16         <td><input type="text" name="creator_id" id="creator_id" value="<%=row.creator_id%>" required/></td>
17       </tr>
18       <tr>
19         <td>제목</td>
20         <td><input type="text" name="title" id="title" value="<%=row.title%>" required/></td>
21       </tr>
22       <tr>
23         <td>내용</td>
24         <td><textarea name="content" id="content" cols="30" rows="10" required><%=row.content%></textarea></td>
25       </tr>
26       <tr>
27         <td>패스워드</td>
28         <td><input type="password" name="passwd" id="passwd" required/></td>
29       </tr>
30       <tr>
31         <td colspan="2">
32           <button type="submit">글쓰기</button>
33         </td>
34       </tr>
35     </table>
36   </form>
```



6. 게시판 구축

■ 업데이트 조회

```
37
38 ▼ <script>
39     function onWriteSubmit()
40     {
41         if ( $("#creator_id").val().trim()=="")
42         {
43             var message = "아이디를 입력해 주세요";
44             $("#creator_id").val("");
45             $("#creator_id").focus();
46             alert(message);
47             return false;
48         }
49         if ( $("#title").val().trim()=="")
50         {
51             var message = "제목을 입력해 주세요";
52             $("#title").val("");
53             $("#title").focus();
54             alert(message);
55             return false;
56         }
57         if ( $("#content").val().trim()=="")
58         {
59             var message = "본문 내용을 입력해 주세요";
60             $("#content").val("");
61             $("#content").focus();
62             alert(message);
63             return false;
64         }
65         if ( $("#passwd").val().trim()=="")
66         {
67             var message = "패스워드를 입력해 주세요";
68             $("#passwd").val("");
69             $("#passwd").focus();
70             alert(message);
71             return false;
72         }
73     }
74 </script>
75 </body>
76 </html>
```



6. 게시판 구축

■ 업데이트 조회

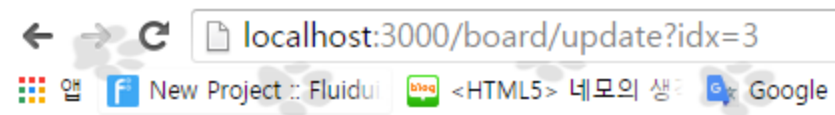
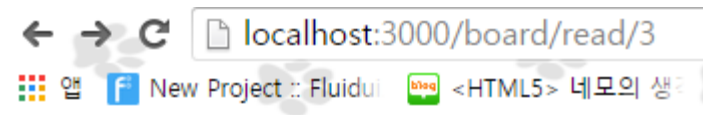
- board.js 파일에 아래 소스 추가

```
87 // 글수정 화면 표시 GET
88 router.get('/update',function(req,res,next)
89 {
90     var idx = req.query.idx;
91
92     pool.getConnection(function(err,connection)
93     {
94         if(err) console.error("커넥션 객체 얻어오기 에러 : ",err);
95
96         var sql = "select idx, creator_id, title, content, hit from board where idx=?";
97         connection.query(sql, [idx], function(err,rows)
98         {
99             if(err) console.error(err);
100             console.log("update에서 1개 글 조회 결과 확인 : ",rows);
101             res.render('update', {title:"글 수정", row:rows[0]});
102             connection.release();
103         });
104     });
105
106 });
```



6. 게시판 구축

- 업데이트 조회
 - 수정 화면 확인



글 조회

작성자	user3
제목	글쓰기테스트
내용	테스트 테스트
조회수	0
글 수정	리스트로 돌아가기

글 수정

작성자	<input type="text" value="user3"/>
제목	<input type="text" value="글쓰기테스트"/>
내용	<div>테스트 테스트</div>
패스워드	<input type="password"/>
글쓰기	<input type="button" value="글쓰기"/>



6. 게시판 구축

■ 업데이트 쓰기

- board.js 파일에 아래 소스 추가

```
108 // 글수정 로직 처리 POST
109 router.post('/update',function(req,res,next)
110 {
111     var idx = req.body.idx;
112     var creator_id = req.body.creator_id;
113     var title = req.body.title;
114     var content = req.body.content;
115     var passwd = req.body.passwd;
116     var datas = [creator_id,title,content,passwd];
117
118     pool.getConnection(function(err,connection)
119     {
120         var sql = "update board set creator_id=?, title=?, content=? where idx=? and passwd=?";
121         connection.query(sql,[creator_id,title,content,idx,passwd],function(err,result)
122         {
123             console.log(result);
124             if(err) console.error("글 수정 중 에러 발생 err : ",err);
125
126             if(result.affectedRows == 0)
127             {
128                 res.send("<script>alert('패스워드가 일치하지 않거나, 잘못된 요청으로 인해 값이 변경되지 않았습니다.');
```



6. 게시판 구축

- 업데이트 쓰기
 - 업데이트 결과 확인

← → ↺ localhost:3000/board/read/3

앱 New Project :: FluidUI <HTML5> 네모의 생

글 조회

작성자	user3
제목	글쓰기테스트
내용	테스트 테스트
조회수	0
글 수정	리스트로 돌아가기

← → ↺ localhost:3000/board/update?idx=3

앱 New Project :: FluidUI <HTML5> 네모의 생 Goo

글 수정

작성자	user3
제목	글쓰기테스트
내용	수정 수정됨
패스워드
글 쓰기	

← → ↺ localhost:3000/board/read/3

앱 New Project :: FluidUI <HTML5> 네모의 생

글 조회

작성자	user3
제목	글쓰기테스트
내용	수정수정됨
조회수	0
글 수정	리스트로 돌아가기