네트워크서비스 프로토콜

9주차 2

- >> 소프트웨어학부
- >> 김형균 교수

수업에 들어가며



- 레포트 공지
- 오늘 학습할 내용
 - MySQL, 워크벤치 설치하기
 - 노드 & 데이터베이스 실습
 - 실습파일 다운로드 후 압축풀기
 - Opentutorials 데이터베이스 생성
 - 테이블 생성 및 데이터 저장
 - 생성 테이블 확인
 - Node.js 세팅하기
 - Node.js와 MySQL 연동
 - MySQL로 웹사이트 구현- 1단계 DB연결
 - MySQL로 웹사이트 구현- 2단계 DB추출

레포트 공지



- >> Node.js 와 mysql을 이용한 웹사이트 제작
- >> 주제 : 나만의 oo 다이어리
- >> 제출기한 : 11월 25일 자정까지
- >> 제출장소 : 가상캠퍼스
- >> 제출형태 : PPT파일 (파일명 : 학번성명)
- >> 필수내용
 - 사이트맵 구성도
 - DB구성도
 - 모듈 구성도(기능 설명 포함)
 - 모듈별 소스 코드
 - 시연
 - 화면단위 캡쳐
 - 시연내역 동영상 캡쳐

레포트 공지



>> 배점 20점

	과제내용(15)					발표자료평가(5)			
구분	기획	기획/설계	기능모듈화	개발		내용 - 41 -	자료		합계
	창의성 (1,2)	적정성 (1,2)	적정성 (1,2,3)	완성도 (5,6,8)	소계	충실도 (1, 2, 3)	완성도 (1, 2)	소계	
최고점	2	2	3	8	15	3	2	5	20
최저점	1	1	1	5	8	1	1	2	10



1. 윈도



>> 공식 사이트(https://dev.mysql.com/downloads/installer)

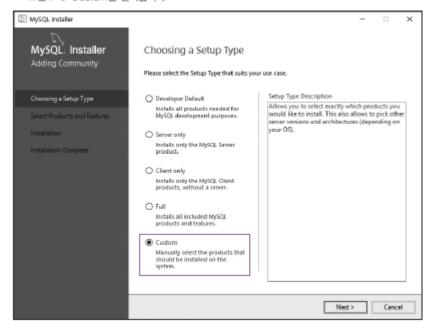




2. 윈도 MySQL Installer

Node. js

- >> 다운로드 받은 MySQL Installer 실행
 - 설치 진행 중 Choosing a Setup Type 부분에서 Custom 선택
 - ♥ 그림 7-5 Custom을 선택합니다

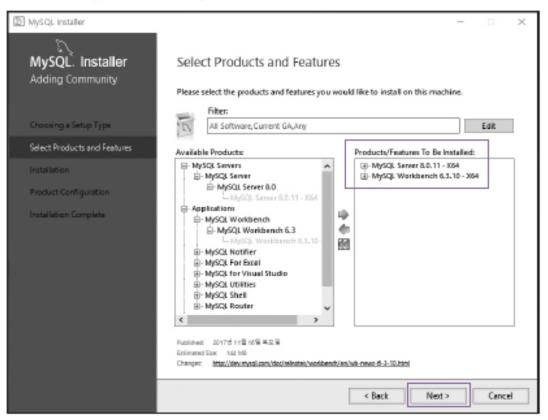


3. Server와 Workbench 설치



>> 자신의 운영체제(x86과 x64 중)에서 Server와 Workbench만 선택하여 설치

▼ 그림 7-6 MySQL Server와 MySQL Workbench 선택

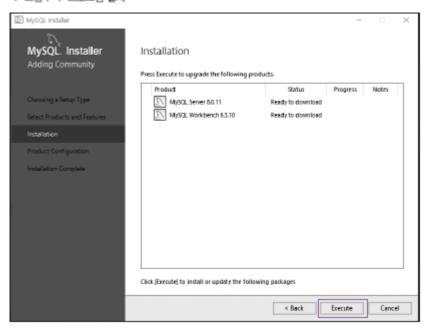


4. Execute

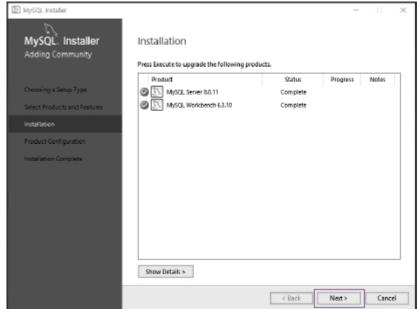


>> Execute로 설치한 후 Next로 마무리

❤ 그림 7-7 프로그램 설치



♥ 그림 7-8 설치 완료 화면

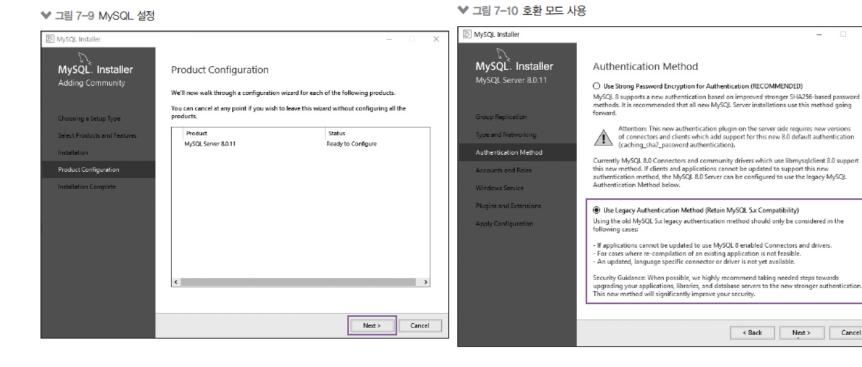


5. MySQL 기본 설정하기



Cancel

>> 계속 Next를 누르다가 Authentication Method 화면에서 Use Legacy Authentication Method 선택

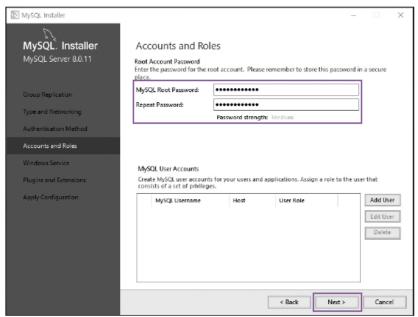


6. 비밀번호 설정하기



>>> Root 비밀번호와 비밀번호 확인 입력 후 설정 적용

♥ 그림 7-11 root 비밀번호 설정



♥ 그림 7-12 설정 반영 전 화면

MySQL Installer		_		×
MySQL. Installer MySQL Server 8.0.11	Apply Configuration Press [Execute] to apply the changes			
mysqcscret old m	Configuration Steps Log			
Group Replication	Writing configuration file Updating Windows Firewall rules			
Type and Networking	Adjusting Windows service			
Authentication Method	○ Initializing Database			
Accounts and Roles	O Starting Server			
Windows Service	Applying security settings Creating user accounts			
Plugins and Extensions	Updating Start Menu Link			
Apply Configuration				
	< Back	Execute	Cano	el

7. MySQL 접속해보기



- >> 콘솔(CMD)에서 MySQL이 설치된 경로로 이동
 - 기본 경로는 C:\Program Files\MySQL\MySQL Server 8.0\bin
 - -h는 호스트, -u는 사용자 -p는 비밀번호 의미

콘솔

```
$ mysql -h localhost -u root -p
Enter password: [비밀번호 입력]
mysql>
```

- 프롬프트가 mysql>로 바뀐다면 성공
- 프롬프트를 종료하려면 exit 입력

콘솔

mysql> exit Bye

8. macOS



>> 홈브루(homebrew)를 통해 설치하는 게 편함

콘솔

- \$ /usr/bin/ruby -e "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/
 master/install)"
- >> brew 설치 후 MySQL 설치

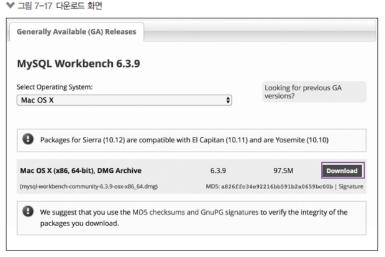
콘솔

- \$ brew install mysql
- \$ brew services start mysql
- \$ mysql_secure_installation
- >> root 비밀번호 설정 후, validate_password 플러그인을 설치하겠냐고 물으면 모두 n을 입력하고 엔터를 눌러 건너 뜀
 - 실서버 운영 시에는 설정해주는 것이 좋음
 - 윈도와 같은 방법으로 MySQL에 접속

9. macOS 워크벤치 설치



- >>> 공식 사이트(https://dev.mysql.com/downloads/workbench)
 - 파일 다운로드 후 응용 프로그램 폴더로 드래그



❤ 그림 7-19 DMG 파일 실행 회면



10. 리눅스(우분투)에 MySQL 설치



>> GUI를 사용하지 않으므로 콘솔에 다음 명령어를 순서대로 입력



>> root 비밀번호 설정



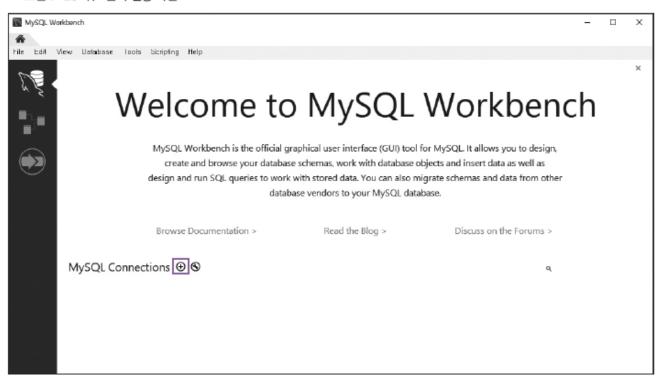
- >>> 윈도와 같은 방법으로 mysql에 접속
- >> 우분투의 경우는 워크벤치 대신 콘솔에서 작업

11. 윈도, 맥 워크벤치에서 커넥션 생성



- >> 워크벤치 프로그램 실행
- >> MySQL Connections 옆에 + 모양 클릭

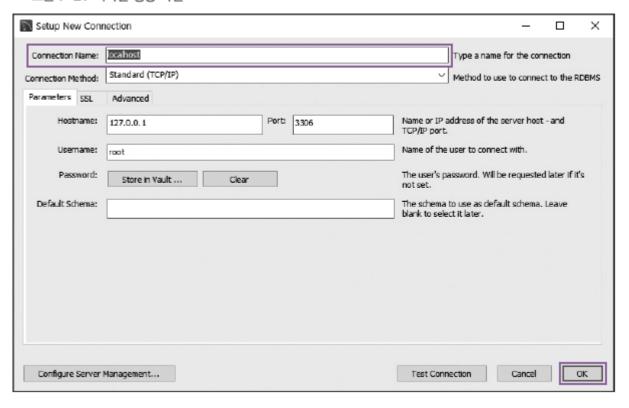
♥ 그림 7-20 워크벤치 실행 화면



12. MySQL 서버 주소 입력



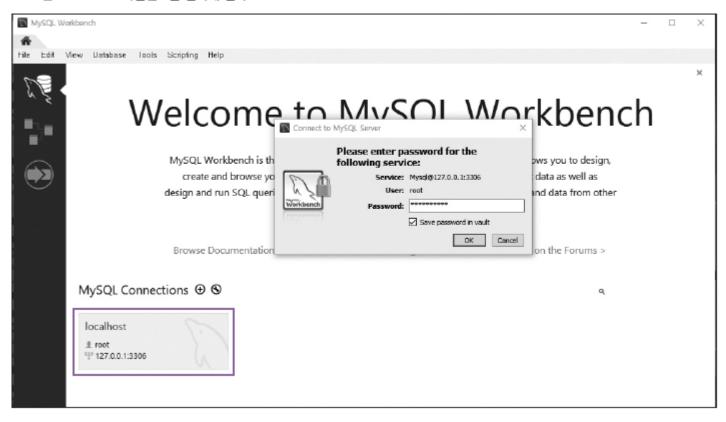
- >>> Connection Name에 localhost 적고 OK 입력
 - ▼ 그림 7-21 커넥션 생성 화면



13. localhost 서버 접속하기



- >> 아까 설정했던 비밀번호를 입력하여 접속
 - ▼ 그림 7-22 root 비밀번호를 입력해 접속



>> 접속이 되었다면 성공



1. 데이터베이스 생성하기



- >> 콘솔에서 MySQL 프롬프트에 접속
 - CREATE SCHEMA nodejs;로 nodejs 데이터베이스 생성
 - use nodejs;로 생성한 데이터베이스 선택

콘솔

mysql> CREATE SCHEMA nodejs;
Query OK, 1 row affected (0.01 sec)
mysql> use nodejs;
Database changed

2. 테이블 생성하기



- >> MySQL 프롬프트에서 테이블 생성
 - CREATE TABLE [데이터베이스명.테이블명]으로 테이블 생성
 - 사용자 정보를 저장하는 테이블

콘솔

```
mysql> CREATE TABLE nodejs.users (

→ id INT NOT NULL AUTO_INCREMENT,

→ name VARCHAR(20) NOT NULL,

→ age INT UNSIGNED NOT NULL,

→ married TINYINT NOT NULL,

→ comment TEXT NULL,

→ created_at DATETIME NOT NULL DEFAULT now(),

→ PRIMARY KEY(id),

→ UNIQUE INDEX name_UNIQUE (name ASC))

→ COMMENT = '사용자 정보'

→ DEFAULT CHARSET=utf8

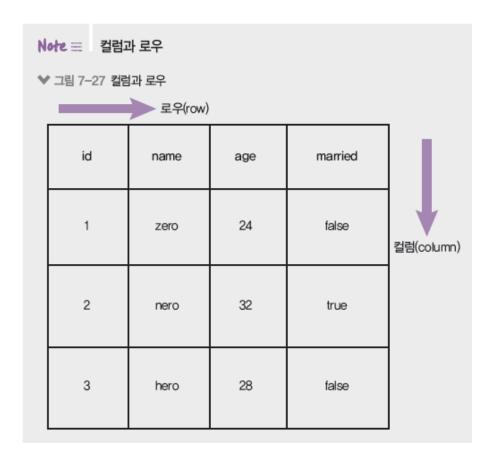
→ ENGINE=InnoDB;

Query OK, 0 row affected (0.09 sec)
```

3. 컬럼과 로우

Node.js

- >> 나이, 결혼 여부, 성별같은 정보가 컬럼
- >> 실제로 들어가는 데이터는 로우



4. 컬럼 옵션들



- >> id Int not null auto_increment
 - 컬럼명 옆의 것들은 컬럼에 대한 옵션들
 - INT: 정수 자료형(FLOAT, DOUBLE은 실수)
 - VARCHAR: 문자열 자료형, 가변 길이(CHAR은 고정 길이)
 - TEXT: 긴 문자열은 TEXT로 별도 저장
 - DATETIME: 날짜 자료형 저장
 - TINYINT: -127에서 128까지 저장하지만 여기서는 1 또는 0만 저장해 불 값 표현
 - NOT NULL: 빈 값은 받지 않는다는 뜻(NULL은 빈 값 허용)
 - AUTO_INCREMENT: 숫자 자료형인 경우 다음 로우가 저장될 때 자동으로 1 증가
 - UNSIGNED: 0과 양수만 허용
 - ZEROFILL: 숫자의 자리 수가 고정된 경우 빈 자리에 0을 넣음
 - DEFAULT now(): 날짜 컬럼의 기본값을 현재 시간으로

5. Primary Key, Unique Index



- >>> PRIMARY KEY(id)
 - id가 테이블에서 로우를 특정할 수 있게 해주는 고유한 값임을 의미
 - 학번, 주민등록번호같은 개념
- >> UNIQUE INDEX name_UNIQUE (name ASC)
 - 해당 컬럼(name)이 고유해야 함을 나타내는 옵션
 - name_UNIQUE는 이 옵션의 이름(아무거나 다른 걸로 지어도 됨)
 - ASC는 인덱스를 오름차순으로 저장함의 의미(내림차순은 DESC)

6. 테이블 옵션



- >> COMMENT: 테이블에 대한 보충 설명(필수 아님)
- >> DEFAULT CHARSET: utf8로 설정해야 한글이 입력됨
- >> ENGINE: InnoDB 사용(이외에 MyISAM이 있음, 엔진별로 기능 차이 존재)

7. 테이블 생성되었나 확인하기



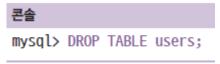
>> DESC 테이블명



♥ 그림 7-28 DESC 명령어 결과

nysal≻ DESC us	sers:		·		·		
Field	Туре	Null	Key	Default	Extra		
id name age married comment created_at	int(11) varchar(20) int(10) unsigned tinyint(4) text datetine	NO NO NO NO YES NO	PRI UNI	NULL NULL NULL NULL NULL CURRENT_T INESTANP	auto_increment		
rows in set (0.01 sec)							

>> 테이블 삭제하기: DROP TABLE 테이블명



8. 댓글 테이블 저장하기



>> comments 테이블 생성

```
콘솔
mysql> CREATE TABLE node;s.comments (
   → id INT NOT NULL AUTO_INCREMENT,
   → commenter INT NOT NULL,
   → comment VARCHAR(100) NOT NULL,
   → created_at DATETIME NOT NULL DEFAULT now(),
   → PRIMARY KEY(id),
   → INDEX commenter idx (commenter ASC),
   → CONSTRAINT commenter
   → FOREIGN KEY (commenter)
   → REFERENCES nodeis.users (id)
   → ON DELETE CASCADE
   → ON UPDATE CASCADE)
   → COMMENT = '댓글'
   → DEFAULT CHARSET=utf8
   → ENGINE=InnoDB;
Query OK, 0 row affected (0.09 sec)
```

9. 외래키



- >> 댓글 테이블은 사용자 테이블과 관계가 있음(사용자가 댓글을 달기 때문)
 - 외래키를 두어 두 테이블이 관계가 있다는 것을 표시
 - FOREIGN KEY (컬럼명) REFERENCES 데이터베이스.테이블명 (컬럼)
 - FOREIGN KEY (commenter) REFERENCES nodejs.users (id)
 - 댓글 테이블에는 commenter 컬럼이 생기고 사용자 테이블의 id값이 저장됨
 - ON DELETE CASCADE, ON UPDATE CASCADE
 - 사용자 테이블의 로우가 지워지고 수정될 때 댓글 테이블의 연관된 로우들도 같이 지워지고 수정됨
 - 데이터를 일치시키기 위해 사용하는 옵션(SET NULL과 NO ACTION도 있음)

10. 테이블 목록 보기



SHOW TABLES;

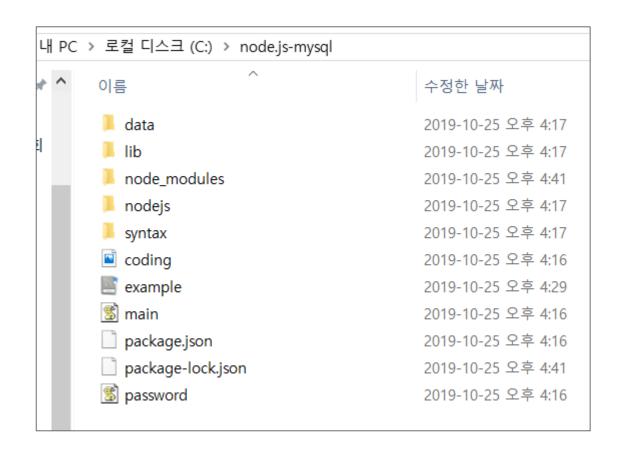


노드 & 데이터베이스 실습

실습파일 다운로드 후 압축풀기



node.js-mysql.zip



Opentutorials 데이터베이스 생성



C:\Users\kim>cd C:\Program Files\MySQL\MySQL Server 8.0\bin C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -h localhost -u root -p mysql> create database opentutorials;

mysql> show databases; mysql> use opentutorials;

```
:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -h localhost -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or #g.
Your MySQL connection id is 27
Server version: 8.0.11 MySQL Community Server - GPL
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names mav be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysgl> create database opentutorials
Query OK, 1 row affected (0.13 sec)
mysql> show database;
<u>FROR 1064 (42000): Y</u>ou have an error in your SQL syntax; check the manual that cor
for the right syntax to use near 'database' at line
nvsal> show databases;
  Database
  information schema
  mvsal
  nodebird
  node is
  opentutorials
  performance_schema
  rows in set (0.04 sec)
nysql> use opentutorials;
 atabase changed
```



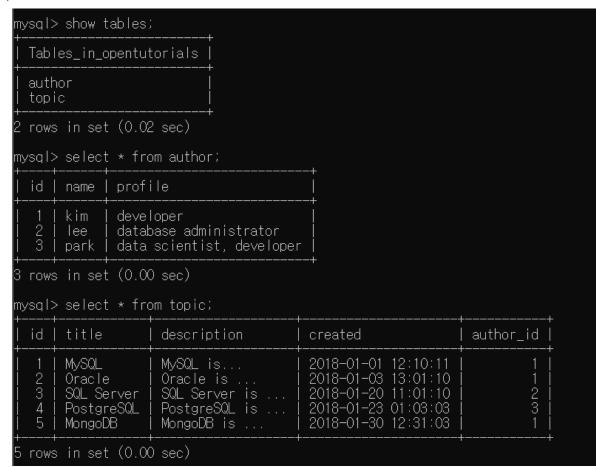


```
-- Table structure for table `author`
CREATE TABLE `author` (
  `id` int(11) NOT NULL AUTO INCREMENT,
 `name` varchar(20) NOT NULL,
 `profile` varchar(200) DEFAULT NULL,
 PRIMARY KEY ('id')
);
-- Dumping data for table `author`
INSERT INTO `author` VALUES (1,'kim','developer');
INSERT INTO `author` VALUES (2,'lee','database administrator');
INSERT INTO `author` VALUES (3,'park','data scientist, developer');
-- Table structure for table `topic`
CREATE TABLE `topic` (
  `id` int(11) NOT NULL AUTO INCREMENT,
 `title` varchar(30) NOT NULL,
 `description` text,
  `created` datetime NOT NULL,
  `author id` int(11) DEFAULT NULL,
 PRIMARY KEY ('id')
);
-- Dumping data for table `topic`
INSERT INTO `topic` VALUES (1, 'MySQL', 'MySQL is...', '2019-01-01 12:10:11',1);
INSERT INTO `topic` VALUES (2, 'Oracle', 'Oracle is ...', '2019-01-03 13:01:10',1);
INSERT INTO `topic` VALUES (3, 'SQL Server', 'SQL Server is ...', '2019-01-20 11:01:10',2);
INSERT INTO `topic` VALUES (4, 'PostgreSQL', 'PostgreSQL is ...', '2019-01-23 01:03:03',3);
INSERT INTO `topic` VALUES (5, 'MongoDB', 'MongoDB is ...', '2019-01-30 12:31:03',1);
```

생성 테이블 확인



```
mysql> show tables;
mysql> select * from author;
mysql> select * from topic;
```

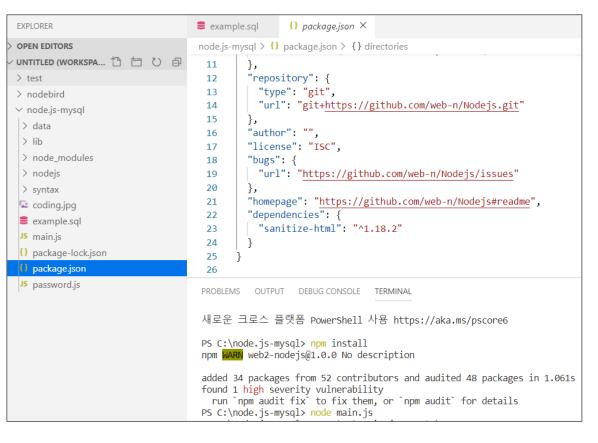


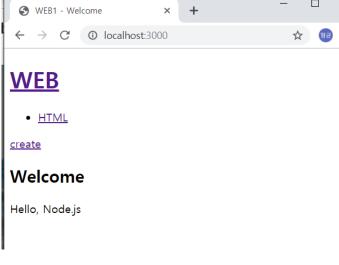
Node.js 세팅하기

Node.js 加山州

PS C:₩node.js-mysql> npm install

PS C:₩node.js-mysql> node main.js

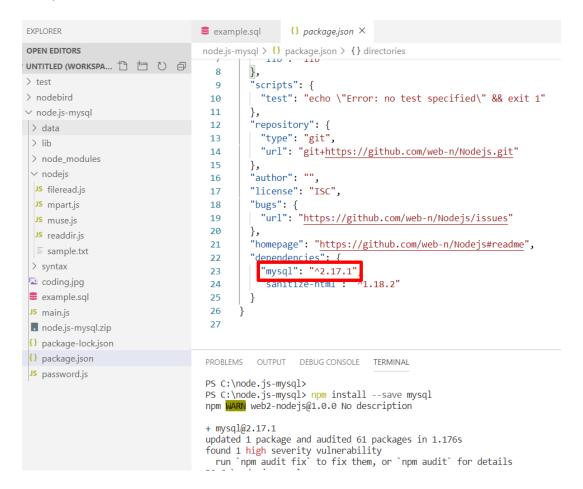




Node.js와 MySQL 연동



- 압축을 푼 폴더에서 mysql 모듈 설치
 - npm install --save mysql
 - packageg.json -> dependencies 변경 확인



Node.js와 MySQL 연동 테스트



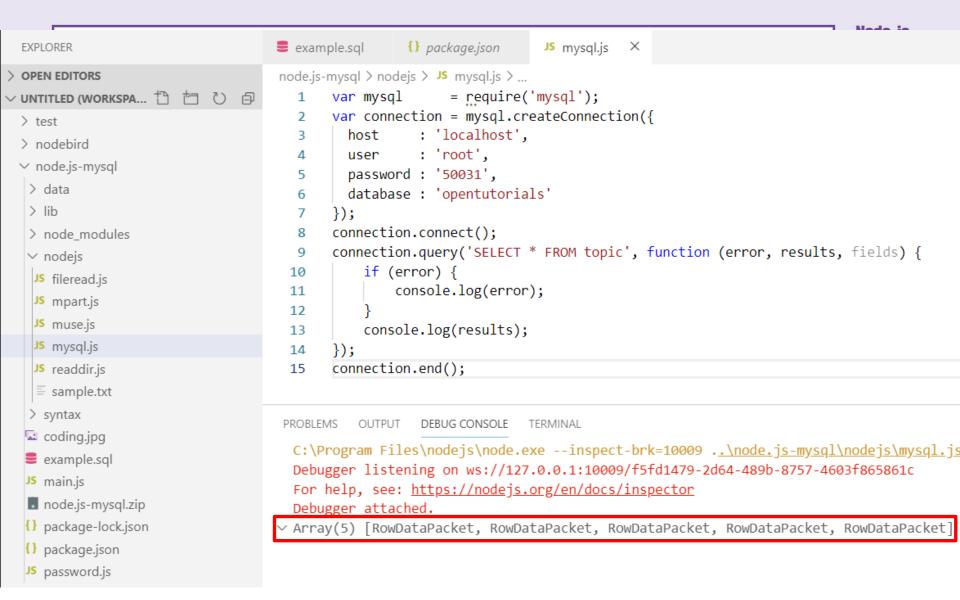
- createConnection 메소드
 - createConnection 메소드의 인자로 전달되는 객체에 자신의 데이터베이스 정보(유저명과 패스워드 등)를 입력

```
var mysql = require('mysql');
var connection = mysql.createConnection({
  host : 'localhost',
  user : ' < MySQL username > ',
  password : ' < MySQL password > ',
  port : < port >,
  database : 'my_db'
});
connection.connect();
connection.query('SELECT * from Persons', function(err, rows, fields) { });
connection.end();
```

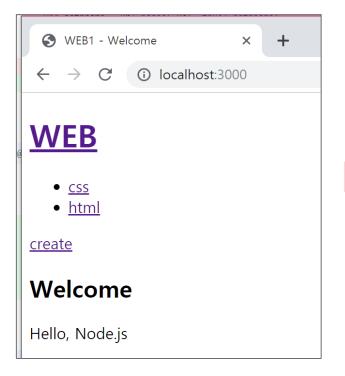


Node.js와 MySQL 연동 테스트

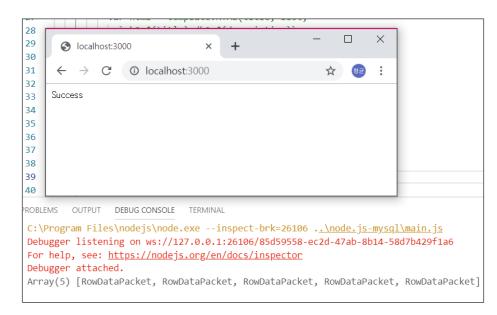
• 압축을 푼 폴더의 nodejs폴더 밑에 mysql.js 생성 var mysql = require('mysql'); var connection = mysql.createConnection({ host: 'localhost', user : 'root', password: ', database: 'opentutorials' }); connection.connect(); connection.query('SELECT * FROM topic', function (error, results, fields) { if (error) { console.log(error); console.log(results); }); connection.end();













>> main.js 변경하기

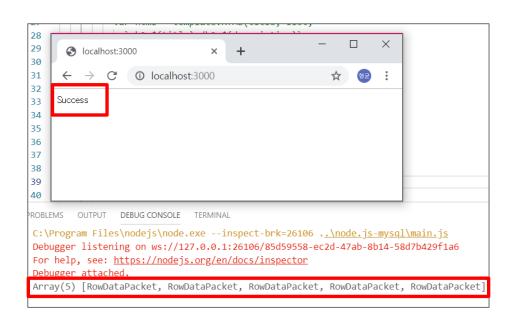
```
\times
example.sgl
                JS mysql.js
                                  Js main.js
node.js-mysql > JS main.js > ...
      var http = require('http');
      var fs = require('fs');
       var url = require('url');
       var qs = require('querystring');
       var template = require('./lib/template.js');
       var path = require('path');
       var sanitizeHtml = require('sanitize-html');
       var mysql = require('mysql');
  8
       var db = mysql.createConnection({
         host: 'localhost',
 10
         user: 'root',
 11
 12
         password: '50031',
         database: 'opentutorials'
 13
       });
 14
       db.connect();
 15
```



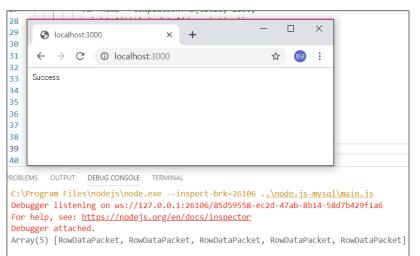
>> main.js 변경하기

```
var app = http.createServer(function(request, response){
   var url = request.url;
   var queryData = url.parse(_url, true).query;
   var pathname = url.parse( url, true).pathname;
    if(pathname === '/'){
      if(queryData.id === undefined){
      /* fs.readdir('./data', function(error, filelist){
          var title = 'Welcome';
          var description = 'Hello, Node.js';
          var list = template.list(filelist);
          var html = template.HTML(title, list,
            `<h2>${title}</h2>${description}`,
           `<a href="/create">create</a>`
          );
          response.writeHead(200);
          response.end(html);
        }): */
       db.query(`SELECT * FROM topic`, function(error, topics){
          console.log(topics);
          response.writeHead(200);
          response.end('Success');
```

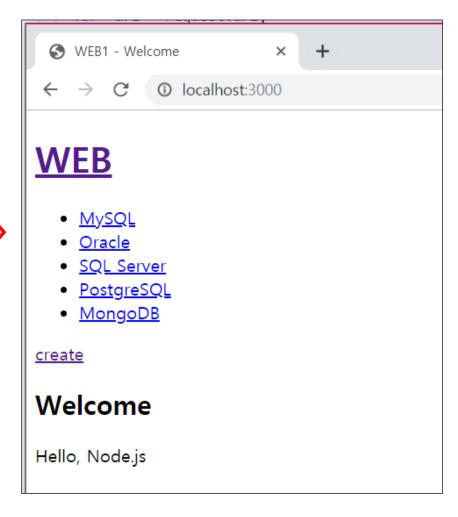














- main.js 변경
- 기존 data폴더의 메인 페이지를 보여주는 내용

```
if(pathname == '/'){
   if(queryData.id == undefined){
    fs.readdir('./data', function(error, filelist){
        var title = 'Home';
        var description = 'Hello, Node.js';
        var list = templateList(filelist);
        var template = templateHTML(title, list,`<h2>${title}</h2>${description}`,
        `<a href="/create">create</a>`);
        response.writeHead(200);
        response.end(template);
    })
```



```
if(pathname === '/'){
   if(queryData.id === undefined){
     db.query(`SELECT * FROM topic`, function(error,topics){
      console.log(topics);
      response.writeHead(200);
      response.end('Success');
   });
```



MySQL로 웹사이트 구현- 2단계 DB추출 ፟፟፟፟፟፟፟፟፟፟፟



- 기존 객체 모듈 is파일
- lib > template.js

fs.readdir('./data', function(error, filelist){

var description = 'Hello, Node.js'; var list = templateList(filelist);

`create`);

if(pathname == '/'){

if(queryData.id == undefined){

response.writeHead(200);

response.end(template);

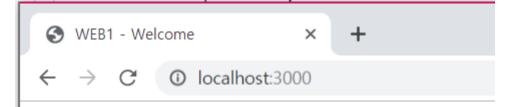
var title = 'Home';

```
node.js-mysql > lib > JS template.js > ...
                                                                                                                                                                                                                                      module.exports = {
                                                                                                                                                                                                                                               HTML:function(title, list, body, control){
                                                                                                                                                                                                                   3
                                                                                                                                                                                                                                                         return `
                                                                                                                                                                                                                                                         <!doctype html>
                                                                                                                                                                                                                  4
                                                                                                                                                                                                                   5
                                                                                                                                                                                                                                                         <html>
                                                                                                                                                                                                                                                          <head>
                                                                                                                                                                                                                  6
                                                                                                                                                                                                                                                                   <title>WEB1 - ${title}</title>
                                                                                                                                                                                                                                                                   <meta charset="utf-8">
                                                                                                                                                                                                                                                          </head>
                                                                                                                                                                                                                                                          <body>
                                                                                                                                                                                                                                                                  <h1><a href="/">WEB</a></h1>
                                                                                                                                                                                                                                                                   ${list}
var template = templateHTML(title, list, \( \hbar h2 > \frac{\h2 > \frac{
                                                                                                                                                                                                                                                                   ${control}
                                                                                                                                                                                                                                                                  ${body}
                                                                                                                                                                                                                                                          </body>
                                                                                                                                                                                                                                                          </html>
                                                                                                                                                                                                               16
                                                                                                                                                                                                              17
                                                                                                                                                                                                                                                },list:function(filelist){
                                                                                                                                                                                                              18
                                                                                                                                                                                                                                                         var list = '':
                                                                                                                                                                                                              19
                                                                                                                                                                                                                                                         var i = 0;
                                                                                                                                                                                                               20
                                                                                                                                                                                                                                                         while(i < filelist.length){</pre>
                                                                                                                                                                                                              21
                                                                                                                                                                                                                                                                 list = list + `<a href="/?id=${filelist[i]}">${filelist[i]}</a>;
                                                                                                                                                                                                              22
                                                                                                                                                                                                                                                                 i = i + 1;
                                                                                                                                                                                                              23
                                                                                                                                                                                                               24
                                                                                                                                                                                                                                                         list = list+'';
                                                                                                                                                                                                              25
                                                                                                                                                                                                                                                         return list;
                                                                                                                                                                                                              26
                                                                                                                                                                                                               27
                                                                                                                                                                                                               28
```

```
Node.js
元山村
```

```
module.exports = {
 HTML:function(title, list, body, control){
   return `
   <!doctype html>
   <html>
   <head>
     <title>WEB1 - ${title}</title>
     <meta charset="utf-8">
   </head>
   <bodv>
     <h1><a href="/">WEB</a></h1>
     ${list}
     ${control}
     ${body}
   </body>
   </html>
         쿼리 id 값을 DB 테이블 인덱스 번호와 연결
```





WEB

- MySQL
- Oracle
- <u>SQL Server</u>
- <u>PostgreSQL</u>
- MongoDB

<u>create</u>

Welcome

Hello, Node.js