# MySQL 연동

## MySQL

- MySQL은 오픈 소스로 제공되는 관계형 데이터베이스 관리 시스템(RDBMS)이다.
- APM(Apache-PHP-MySQL) 플랫폼의 데이터베이스 구성체로 작동된다.
- MySQL은 GPL/Commercial License의 듀얼 라이선스가 적용된다.

✓ 본 챕터에서는 mysql의 CRUD(Create, Read, Update, Delete) 작업은 다루지 않는다.
만약 CRUD작업을 이해하지 못했더라면 먼저, 그 부분을 구글링 등을 통해 이해 및 습득한 후에 해당 챕터를 진행해 주길 바란다.
(참조 : https://dejavuqa.tistory.com/317)

## 예제1) MySQL모듈 설치 및 데이터 입·출력

1. mkdir 명령어로 새로운 디렉터리를 생성한다.

2. 생성된 디렉터리로 이동 후 npm init를 실행하여 해당 디렉터리를 초기화한다.

```
wlgns12www@nodejs-test:~/nodejs/8_mysql$ ls
wlgns12www@nodejs-test:~/nodejs/8_mysql$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (8_mysql)
version: (1.0.0)
description: connect to nodejs to mysql
entry point: (index.js) main.js
test command:
git repository:
keywords:
author: INLAB Ji-Hoon
license: (ISC)
About to write to /home/wlgns12www/nodejs/8_mysql/package.json:

{
  "name": "8_mysql",
  "version": "1.0.0",
  "description": "connect to nodejs to mysql",
  "main": "main.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "INLAB Ji-Hoon",
  "license": "ISC"
}


Is this ok? (yes) yes
```

: entry point는 main.js로 기입
: 나머지 항목은 자유롭게 기입

3. mysql모듈 설치 (npm init된 디렉터리에서 명령어를 진행)

   $ npm install mysql

4. 데이터베이스에 테이블 및 데이터 생성

<테이블 생성>

```
CREATE TABLE User (
        id VARCHAR(45) NOT NULL,
        pwd VARCHAR(45) NOT NULL,
        PRIMARY KEY(id)
);
```

<생성된 테이블로 접근>

```
use User;
```

<테이블내에 데이터 삽입>

```
INSERT INTO User(id, pwd) VALUES("testuser1", "1234");
INSERT INTO User(id, pwd) VALUES("testuser2", "5678");
```

<테이블이 생성되었는지 확인>

```
show tables;
```

```
mysql> show tables;
+------------------+
| Tables_in_testdb |
+------------------+
| User             |
+------------------+
```

<데이터가 삽입되었는지 확인>

```
SELECT * FROM User;
```

```
mysql> select * from User;
+-----------+------+
| id        | pwd  |
+-----------+------+
| testuser1 | 1234 |
| testuser2 | 5678 |
+-----------+------+
```

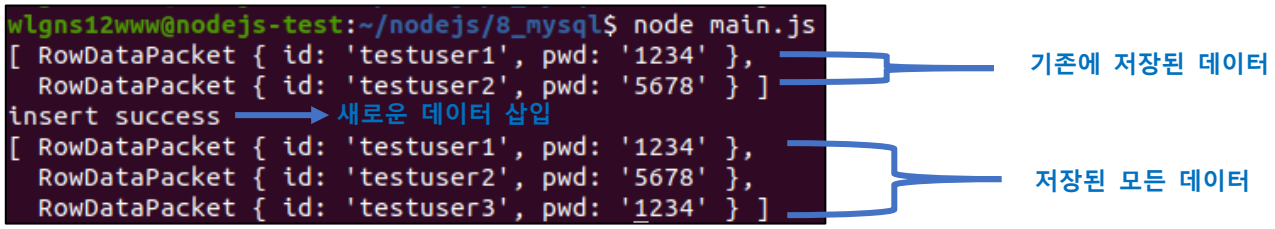
5. main.js에 아래의 내용과 같이 입력

   $ vi main.js

```
"use strict";
const mysql = require("mysql");
const connection = mysql.createConnection({ //연결할 DB정의
        host     : "localhost", // 연결할 host ip 주소
        user     : "testuser", // DB 사용자 이름
        password : "inlab1234", // 사용자 패스워드
        database : "testdb" // 사용할 DB 이름
});
let sql; // DBMS로 전송할 SQL문 (CRUD작업)
let insert_params;

connection.connect();

// 기존에 User테이블에 저장된 데이터 확인
sql = "SELECT * FROM User";
connection.query(sql, function(err, rows, fields){
        if(err) console.log(err);
        console.log(rows);
});

// 새로운 데이터 삽입
sql = "INSERT INTO User(id, pwd) VALUES(?,?)"
insert_params = ["testuser3", "1234"];
connection.query(sql, insert_params, function(err, rows, fields){
        if(err) console.log(err);
        else console.log("insert success");
});

// User테이블에 저장된 모든 데이터 확인
sql = "SELECT * FROM User";
connection.query(sql, function(err, rows, fields){
        if(err) console.log(err);
        console.log(rows);
});

connection.end();
```

✓ main.js 실행화면

$ node main.js




✓ DB에 새롭게 저장된 testuser3 데이터 확인


## 참고문헌


- Ung-mo Lee. https://poiemaweb.com/nodejs-mysql. Express, 10.4
- Tongchun. https://dejavuqa.tistory.com/317. Database/MYSQL, mysql설치와 기본설정(on Ubuntu)