데이터베이스설계 (ICE4016)

실습 4주차

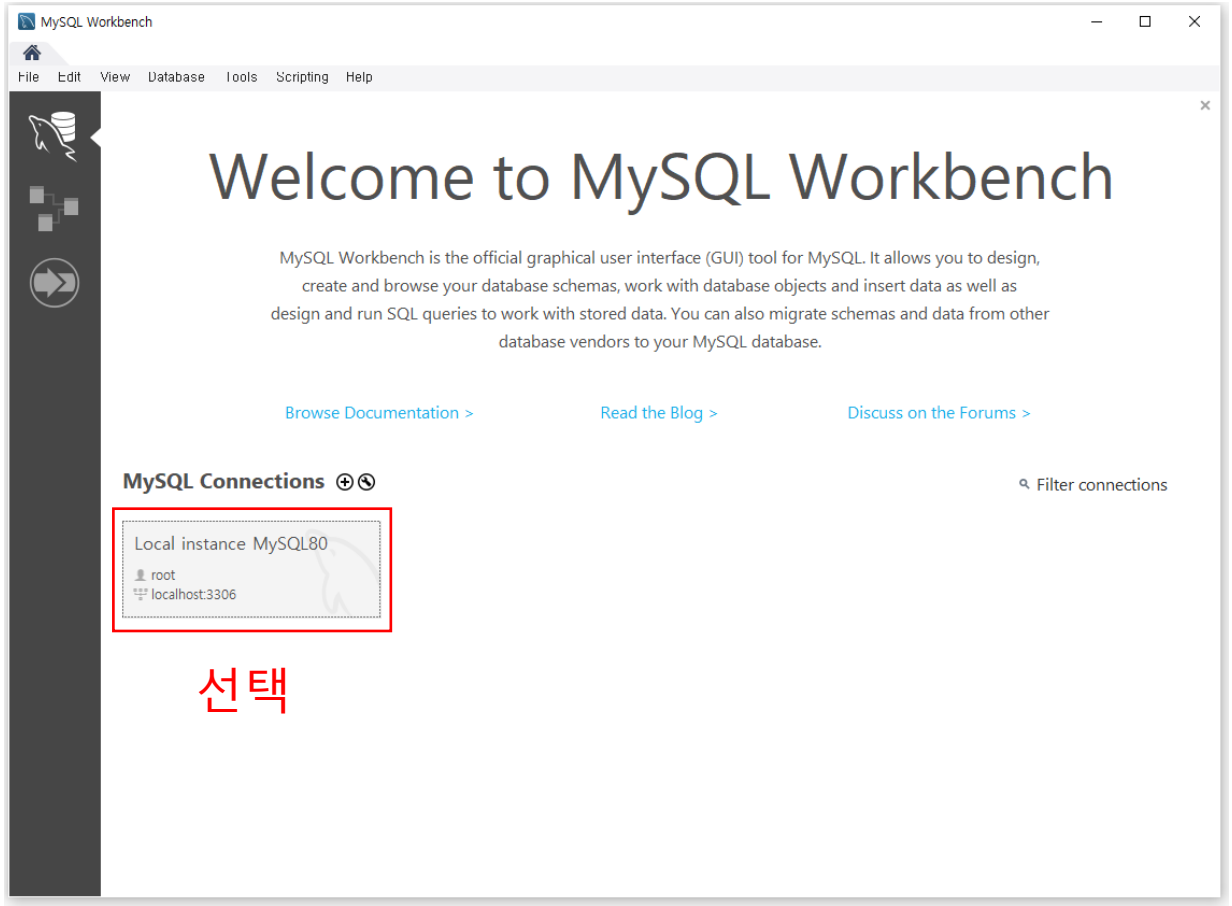
Node.js

Prof. Wonik Choi

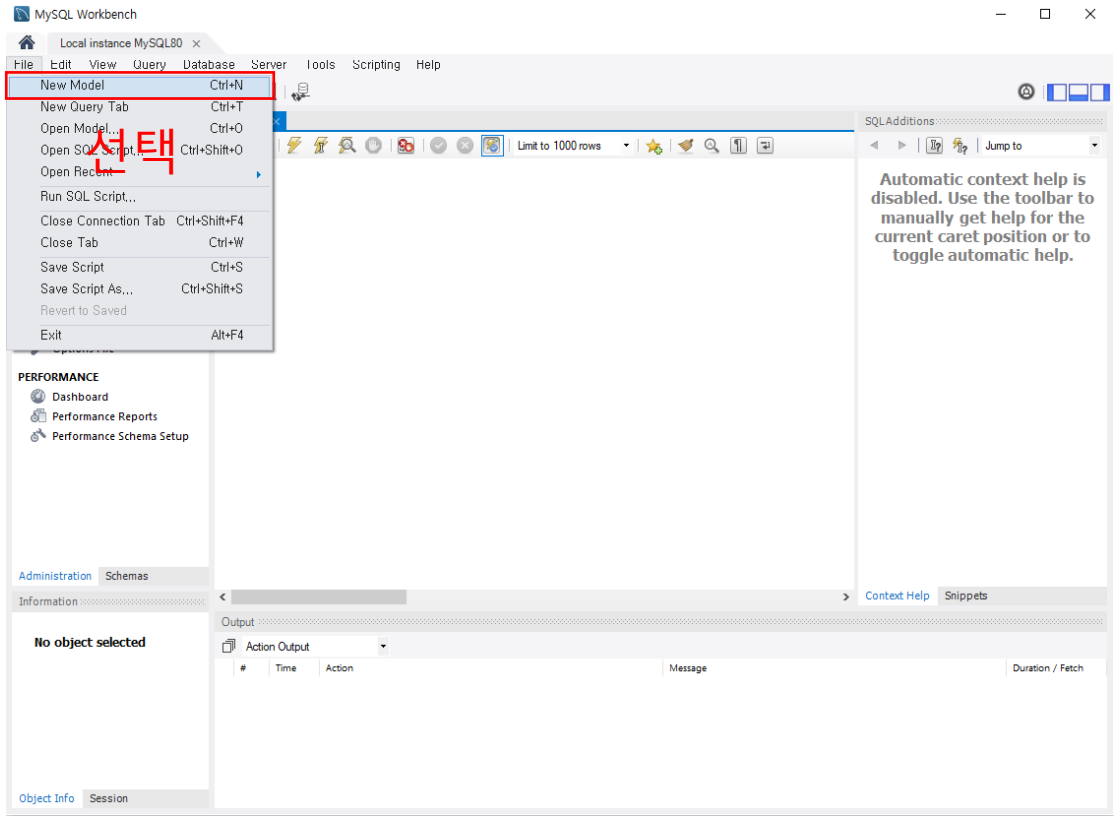
데이터 인텔리전스 연구실
Data Intelligence Laboratory

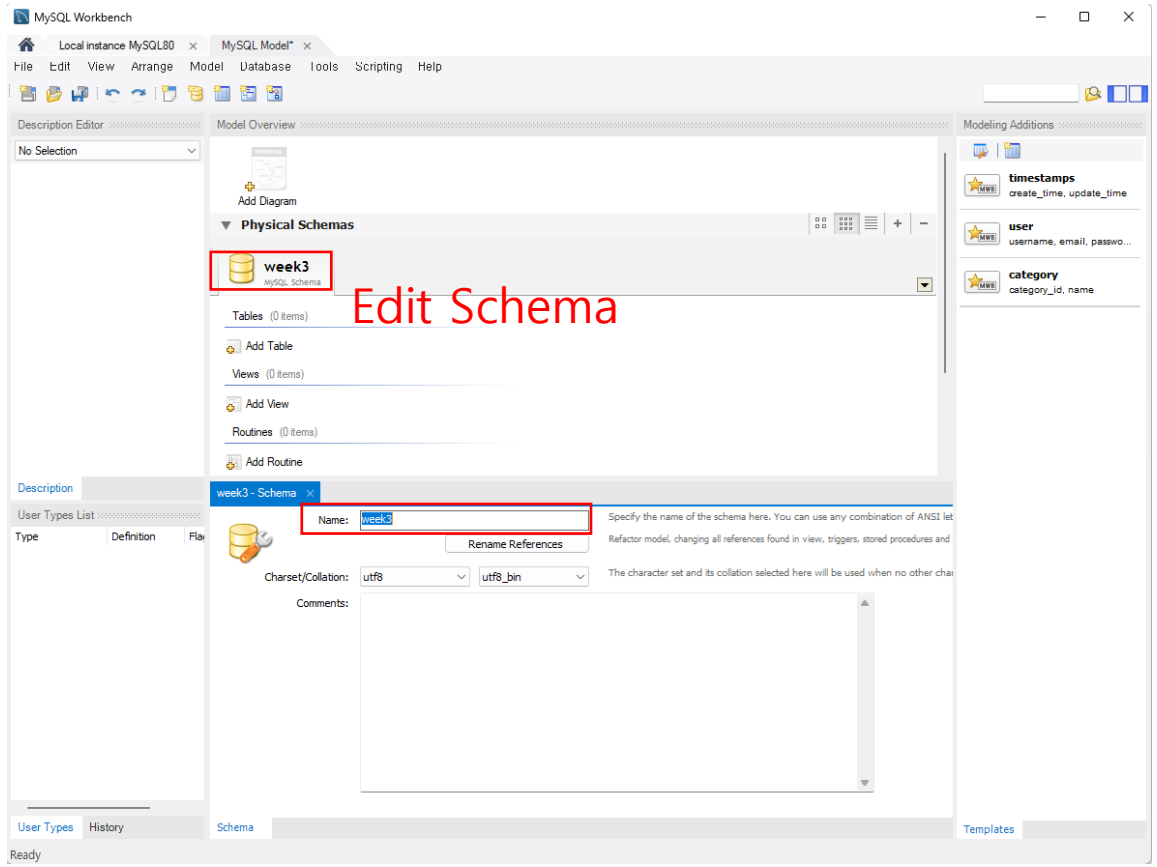
인하대학교
INHA UNIVERSITY

MySQL Workbench

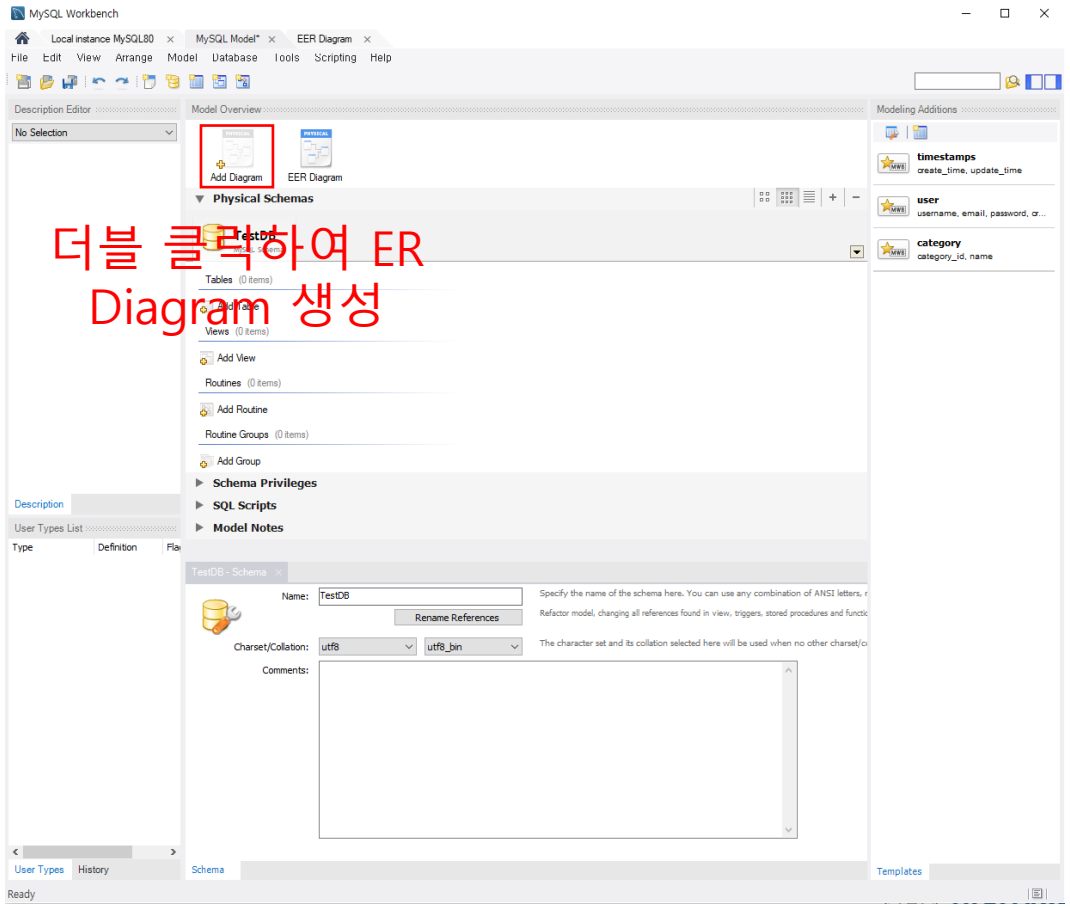


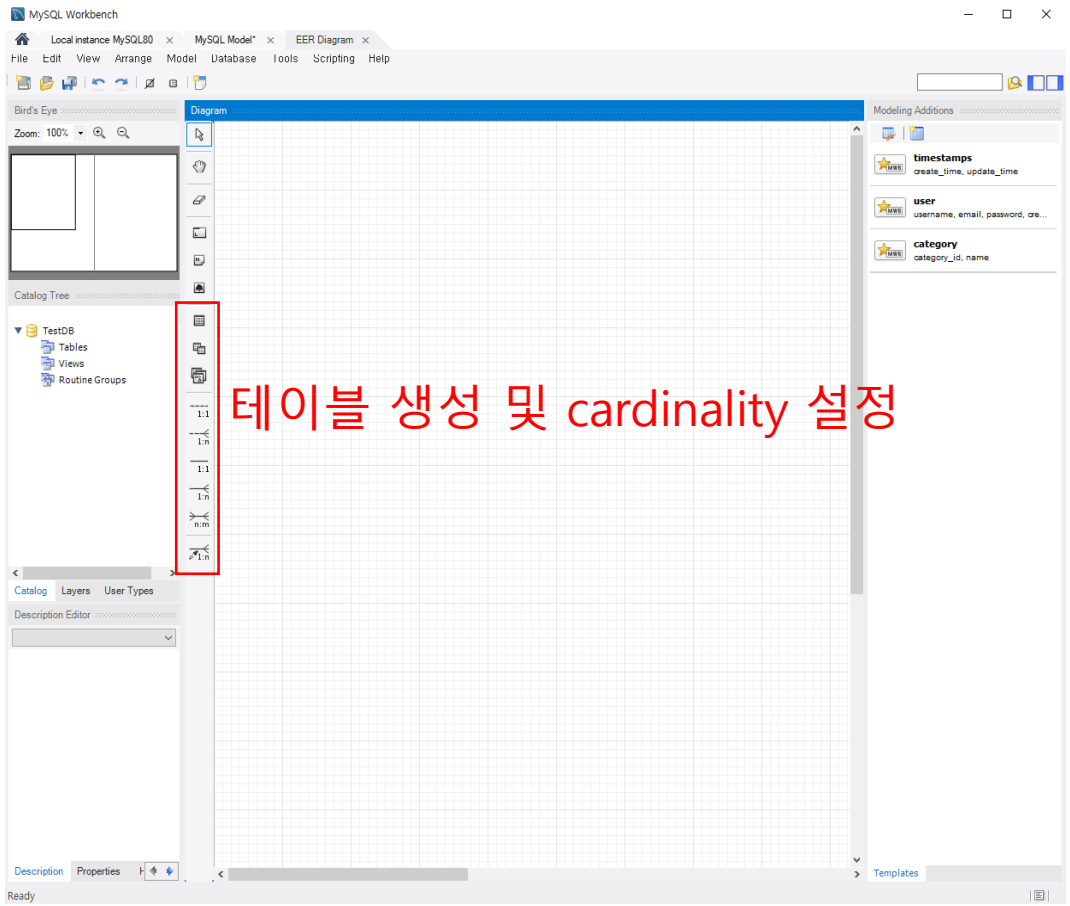
MySQL Workbench





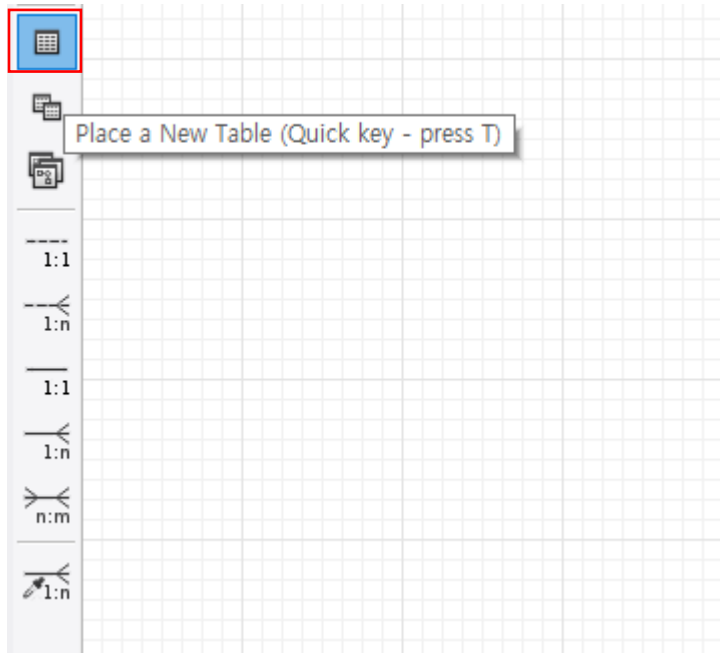
MySQL Workbench



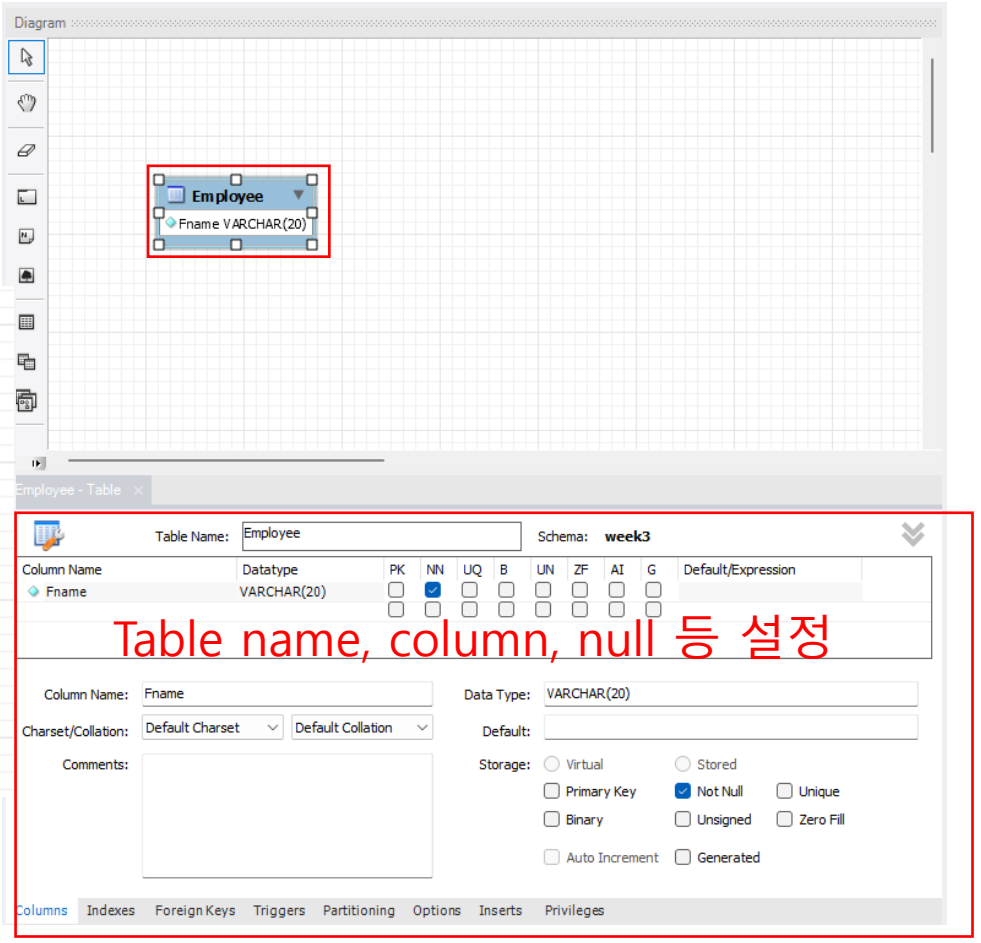


MySQL Workbench

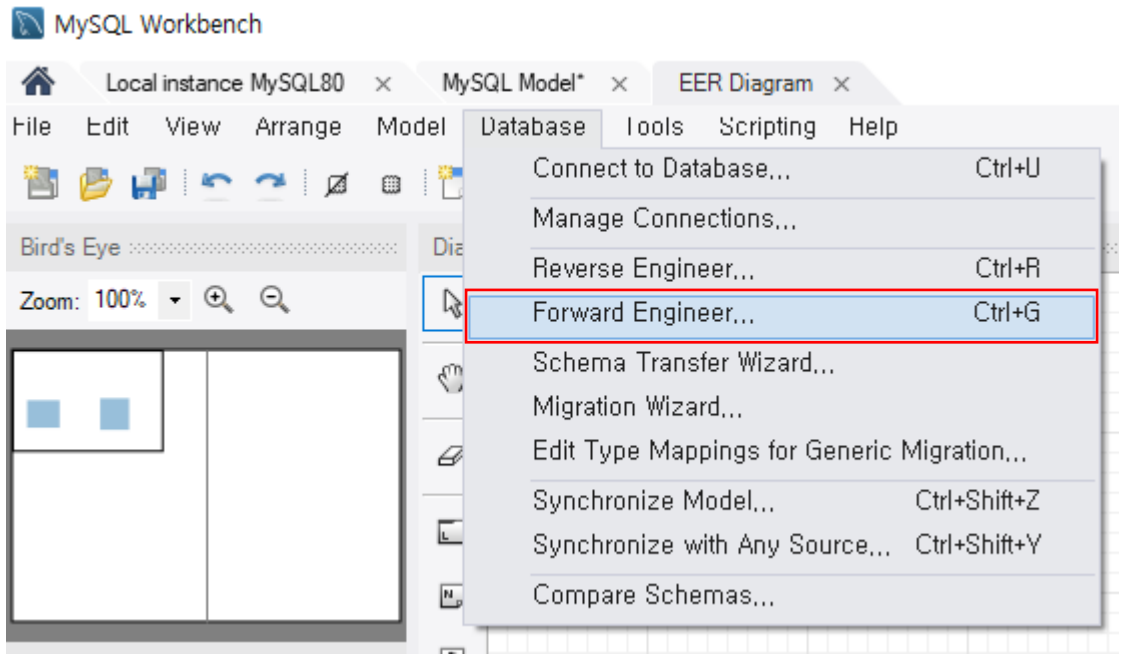
클릭 후 옆의 도면 아무 곳에 클릭



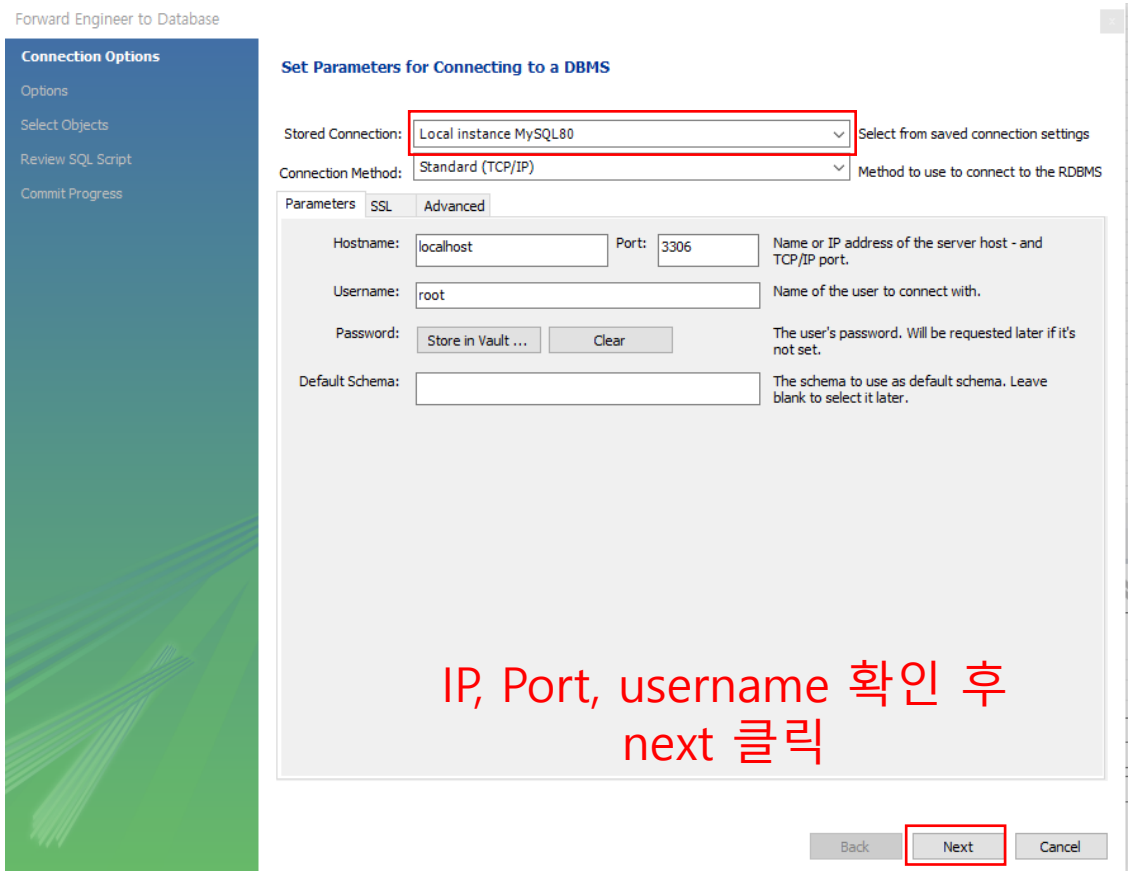
더블 클릭



MySQL Workbench

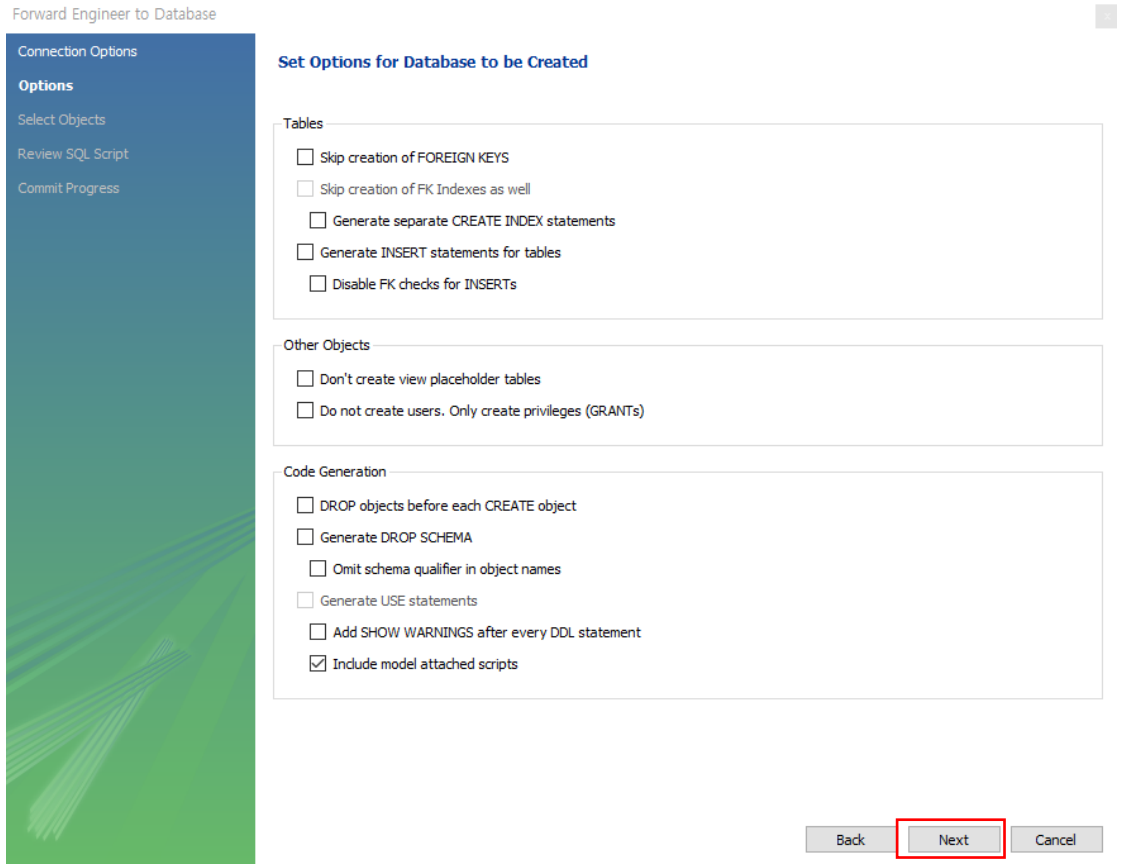


다이어그램을 실제 DB에 반영

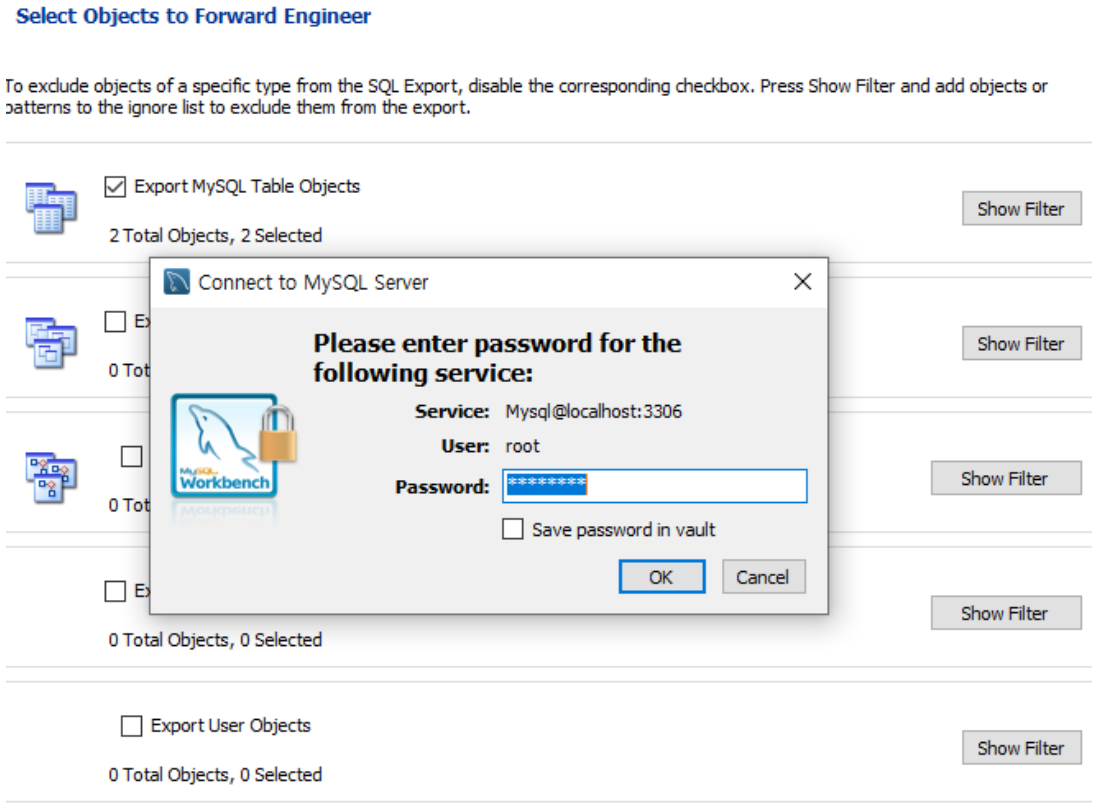


IP, Port, username 확인 후
next 클릭

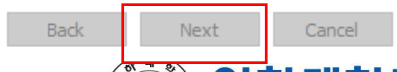
MySQL Workbench



옵션 선택(실습에서는 사용하지 않음)

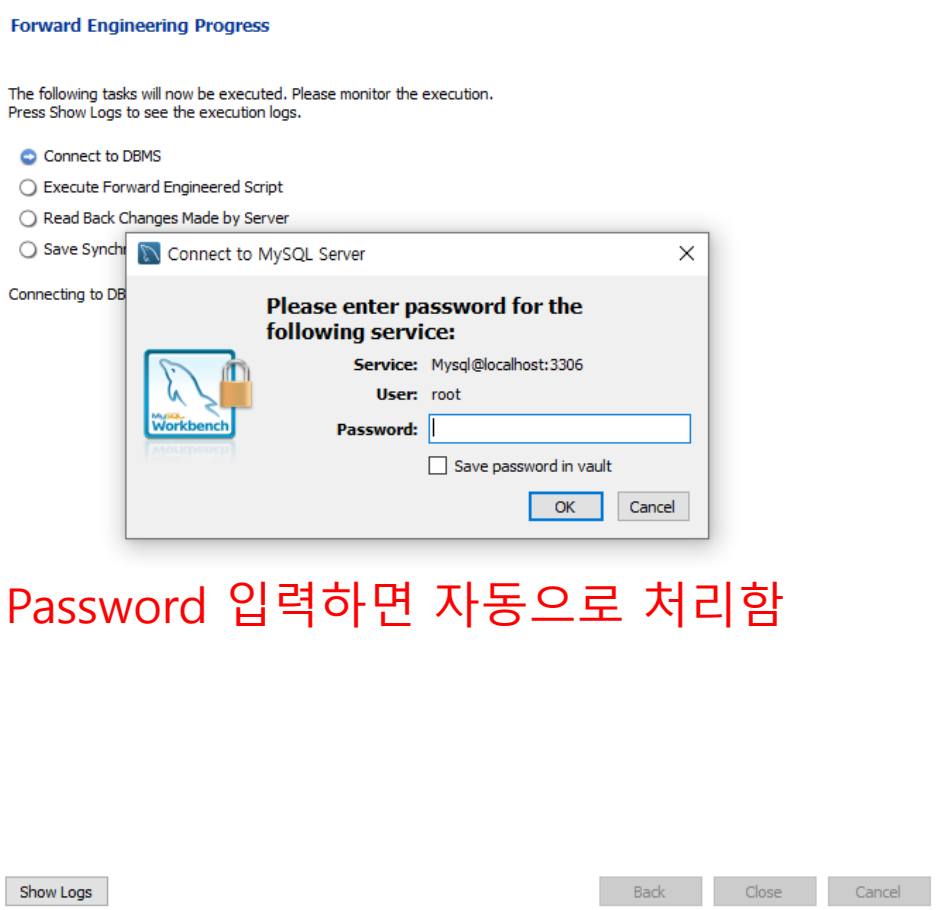


Password 입력 후 next



MySQL Workbench

DB 및 테이블 생성을 위한 script 생성됨



Password 입력하면 자동으로 처리함

MySQL Workbench

```
PS C:\Users\leesw> mysql -u root -p
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 69
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show Databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| week3              |
| world              |
+--------------------+
```

데이터베이스 생성 확인

```
mysql> use week3
Database changed
mysql> show tables;
+----------------+
| Tables_in_week3 |
+----------------+
| employee       |
+----------------+
1 row in set (0.00 sec)

mysql> desc employee;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| Fname | varchar(20) | NO   |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
1 row in set (0.00 sec)

mysql>
```

데이터베이스 지정

테이블 정보 확인

테이블 정보 확인

실습 환경 구축

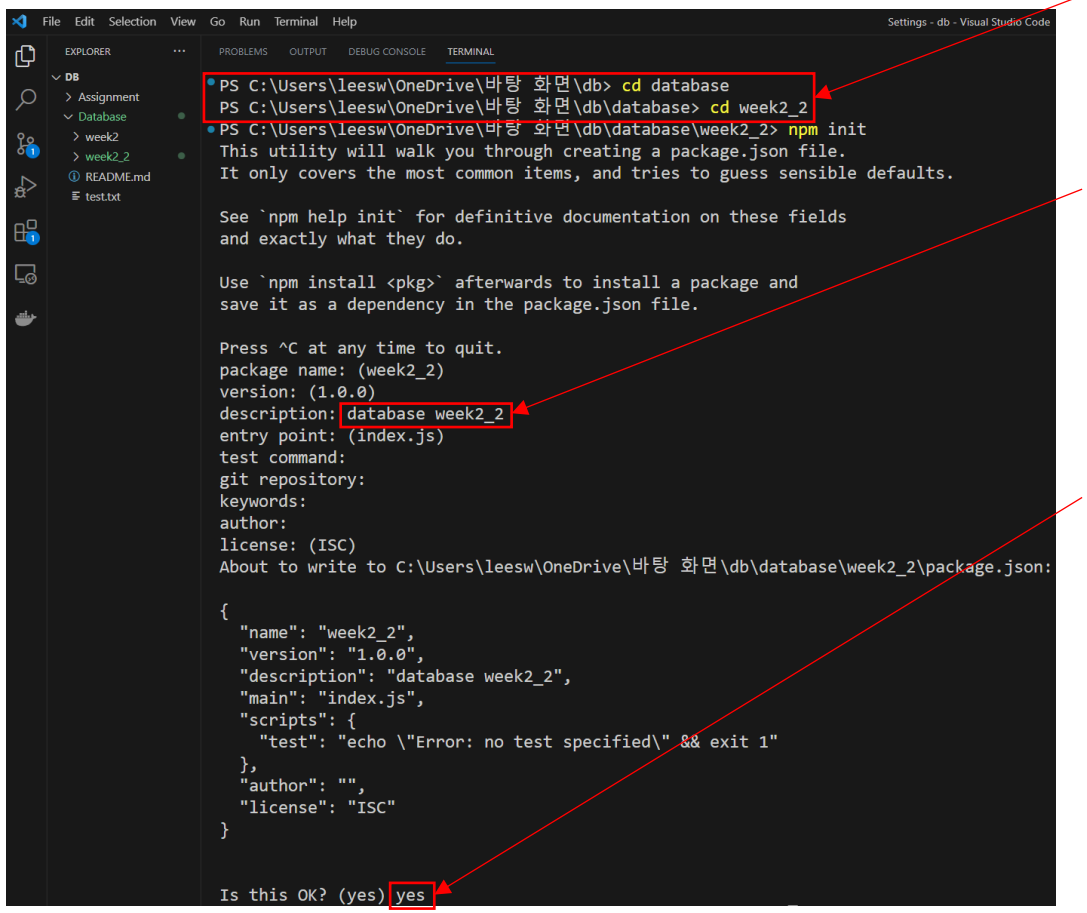
- Windows 10~
  - NVM 및 Node 설치 오류
    - 사용자 계정 이름이 한글이 경우 발생
      - 설치 경로 변경하여 해결
    - https://github.com/coreybutler/nvm-windows/releases
    - 1.1.11 (Latest) 하단의 nvm-setup.zip 파일 다운받아서 설치
    - Accept(next) -> next -> next -> install
    - 설치 후 windows terminal에서 nvm 버전 확인
      - nvm version
    - Node 설치 및 사용
      - nvm install 16.10.0
      - nvm use 16.10.0
      - node -v

```
PS C:₩Windows₩system32> nvm version
1.1.11
PS C:₩Windows₩system32> nvm install 16.10.0
Downloading node.js version 16.10.0 (64-bit)...
Extracting node and npm...
Complete
npm v7.24.0 installed successfully.


Installation complete. If you want to use this version, type

nvm use 16.10.0
PS C:₩Windows₩system32> nvm use 16.10.0
Now using node v16.10.0 (64-bit)
PS C:₩Windows₩system32> node -v
v16.10.0
```

서버 구축

○ 프로젝트 생성 및 환경 세팅

- npm init 명령어 실행



Github repository로 이동
week2_2 디렉토리 생성 후 이동

프로젝트 설명 작성
나머지 enter

yes 작성 후 enter

서버 구축

- Package.json
  - 의존중인 패키지들의 버전을 기록 및 관리
    - 프로젝트와 관련된 메타데이터를 기록하는 파일
- Express
  - Node.js의 핵심 모듈인 http와 Connect 컴포넌트를 기반으로 한 빠르고 간편한 웹 프레임워크
    - Node.js를 통해 다양한 자바스크립트 애플리케이션을 실행
    - 웹서버와 같이 확장성 있는 네트워크 프로그램을 제작하기 위해 만들어짐
- Babel
  - 최신 JS 문법을 모든 브라우저에서 사용할 수 있는 형태로 변환할 수 있는 transcomplier
- Nodemon
  - 웹개발 편의성을 높이기 위한 스크립트 모니터링 유틸리티
    - 어떤 파일이라도 수정될 경우, 이를 반영하여 자동으로 서버를 재시작

서버 구축

○ 프로젝트 생성 및 환경 세팅

- 환경 세팅을 위한 모듈 설치 대소문자 주의!!
  - npm install express mysql2 body-parser nodemon, morgan
  - npm install @babel/node @babel/core @babel/preset-env

```
PS C:\Users\leesw\OneDrive\바탕 화면\db\Database\week2_2> npm install express mysql2 body-parser nodemon

added 104 packages, and audited 105 packages in 6s

11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
PS C:\Users\leesw\OneDrive\바탕 화면\db\Database\week2_2> npm install @babel/node @babel/core @babel/preset-env

added 245 packages, and audited 322 packages in 11s

62 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

서버 구축

○ 프로젝트 생성 및 환경 세팅

- package.json 파일 수정
  - "scripts" 부분 아래 구문 추가
  - "start": "nodemon --exec babel-node index.js"
    - ➢ babel-node: JS 최신 문법 적용
    - ➢ nodemon: 변경사항 탐지 및 자동 재시작

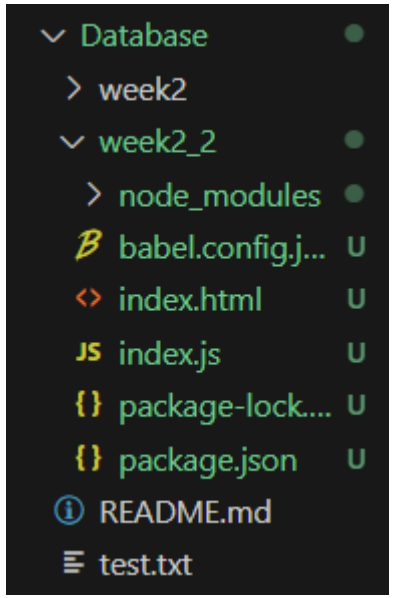
Database > week2_2 > {} package.json > ...

```
{
  "name": "week2_2",
  "version": "1.0.0",
  "description": "database week2_2",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "nodemon --exec babel-node index.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "@babel/core": "^7.22.15",
    "@babel/node": "^7.22.15",
    "@babel/preset-env": "^7.22.15",
    "body-parser": "^1.20.2",
    "express": "^4.18.2",
    "mysql2": "^3.6.0",
    "nodemon": "^3.0.1"
  }
}
```

서버 구축

○ 프로젝트 생성 및 환경 세팅

- week2_2 폴더에 파일 생성
  - babel.config.json, index.html, index.js 파일 생성
  - 각각의 파일에 아래의 내용 작성



Database > week2_2 > babel.config.json > ...

```
{
    "presets": ["@babel/preset-env"]
}
```

Database > week2_2 > index.html > html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Database week2_2</title>
    </head>
    <body>
        <h1>Hello Express!</h1>
    </body>
</html>
```

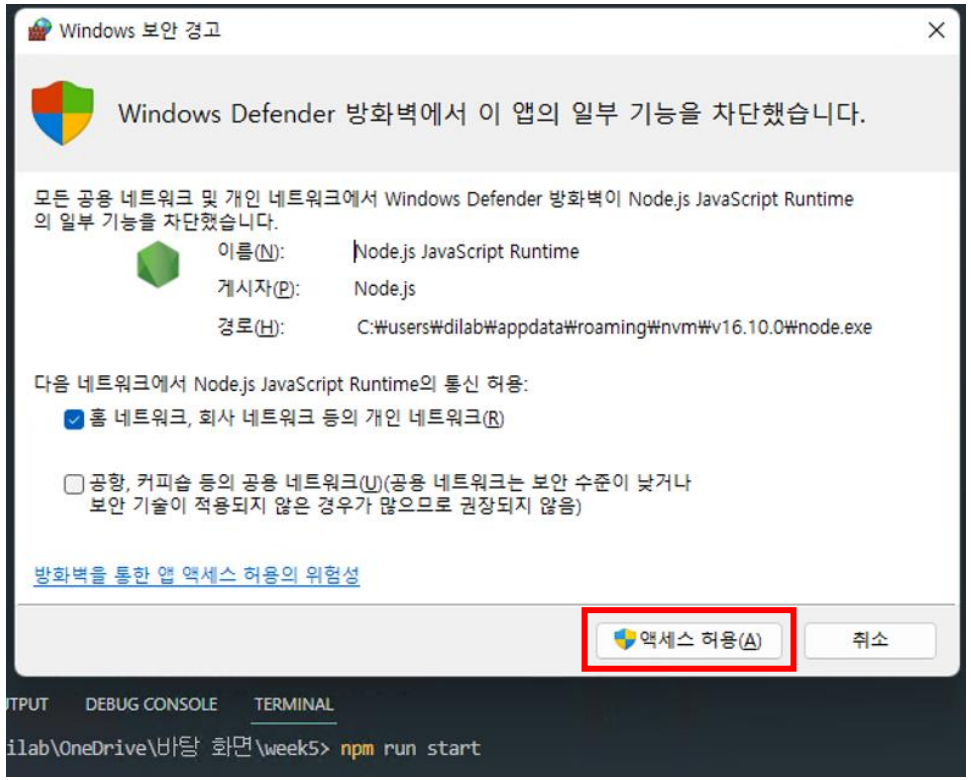
Database > week2_2 > index.js > ...

```
import express from "express";
import bodyParser from "body-parser";
import path from 'path';

const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

app.get("/", (req, res)=>{
    res.sendFile(path.join(__dirname+'/index.html'));
});

app.listen(3000, ()=>{
    console.log("Server is running on port 3000.");
})
```

서버 구축

○ 프로젝트 생성 및 환경 세팅

- 서버 실행 확인
  - npm run start

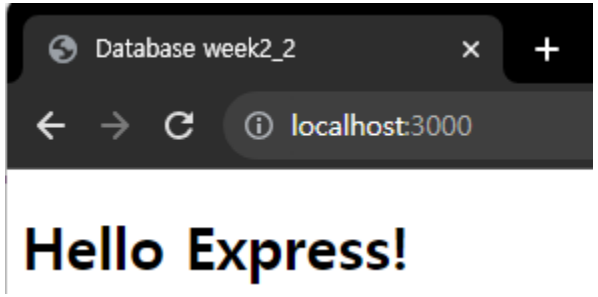


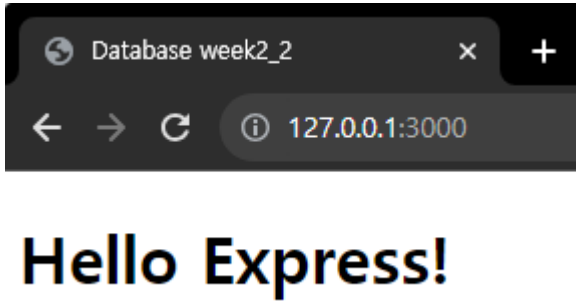

```
PS C:\Users\leesw\OneDrive\바탕 화면\db\Database\week2_2> npm run start

> week2_2@1.0.0 start
> nodemon --exec babel-node index.js

[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `babel-node index.js`
Server is running on port 3000.
```

서버 구축

○ 프로젝트 생성 및 환경 세팅

- 서버 실행 확인
  - http://localhost:3000/ 또는 http://127.0.0.1:3000/





MySQL 연동

○ Express-Mysql 연동

- week2_2 Root 폴더에 파일 생성
  - sql.js 파일 생성
  - index.js 파일 내용 수정

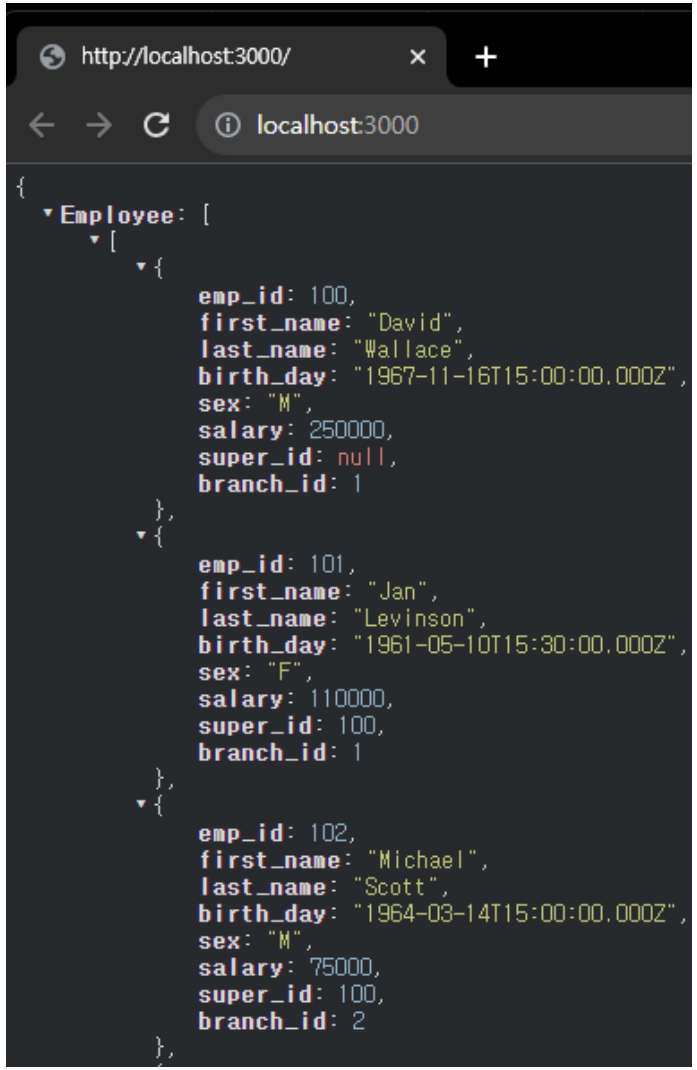
```sql.js
import mysql from 'mysql2';

const pool = mysql.createPool({
    host: 'localhost',
    port: 3306,
    user: 'root',
    password: '',
    database: 'week3_company',
})

const promisePool = pool.promise();

const sql = {
    getEmployee: async()=>{
        const results = await promisePool.query(`
        select * from employee
        `)

        return results;
    },
};

export default sql;
```

Mysql pw

```index.js
import express from "express";
import bodyParser from "body-parser";
import path from 'path';
import sql from './sql';

const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

app.get("/", async (req, res)=>{
    //res.sendFile(path.join(__dirname+'/index.html'));
    const employee = await sql.getEmployee();
    res.json({"Employee": employee})
});

app.listen(3000, ()=>{
    console.log("Server is running on port 3000.");
})
```

MySQL 연동

- Express-Mysql 연동
  - 결과 확인



실습 4주차 실습

- ‘Corporate’ branch에 해당하는 모든 사원의 이름, 기존 급여, 10% 증가된 급여를 출력하라.
- 급여가 60,000에서 80,000 사이에 있는 모든 남자 사원의 이름, 급여를 출력하라.
- 모든 사원을 1. branch_id(내림차순) 2. 급여(오름차순)으로 정렬하고, 이름, branch_id, 급여를 출력하라.
- ‘FedEx’와 일하는 급여 60,000 이상의 모든 사원의 이름, total_sales를 출력하라.
- 사원의 급여의 합, 최고 급여, 최저 급여, 평균 급여를 출력하라.
- 회사의 총 사원수를 제시하라.
- 각 branch별 근무하는 사원의 수를 검색하여 branch 이름과 소속 사원수를 출력하라

실습 4주차 실습

```
mysql> source week3.sql
+------------+-----------+--------+-------------------+
| first_name | last_name | salary | increased_salary  |
+------------+-----------+--------+-------------------+
| David      | Wallace   | 250000 |         275000.00 |
| Jan        | Levinson  | 110000 |         121000.00 |
+------------+-----------+--------+-------------------+
2 rows in set (0.00 sec)

+------------+-----------+--------+
| first_name | last_name | salary |
+------------+-----------+--------+
| Michael    | Scott     |  75000 |
| Stanley    | Hudson    |  69000 |
| Josh       | Porter    |  78000 |
| Andy       | Bernard   |  65000 |
| Jim        | Halpert   |  71000 |
+------------+-----------+--------+
5 rows in set (0.00 sec)

+------------+-----------+-----------+--------+
| first_name | last_name | branch_id | salary |
+------------+-----------+-----------+--------+
| Andy       | Bernard   |         3 |  65000 |
| Jim        | Halpert   |         3 |  71000 |
| Josh       | Porter    |         3 |  78000 |
| Kelly      | Kapoor    |         2 |  55000 |
| Angela     | Martin    |         2 |  63000 |
| Stanley    | Hudson    |         2 |  69000 |
| Michael    | Scott     |         2 |  75000 |
| Jan        | Levinson  |         1 | 110000 |
| David      | Wallace   |         1 | 250000 |
+------------+-----------+-----------+--------+
9 rows in set (0.00 sec)

+------------+-----------+-------------+
| first_name | last_name | total_sales |
+------------+-----------+-------------+
| Jim        | Halpert   |       22500 |
| Michael    | Scott     |       15000 |
| Stanley    | Hudson    |      130000 |
+------------+-----------+-------------+
3 rows in set (0.00 sec)

+--------------+-------------+-------------+-------------+
| total_salary | max_salary | min_salary | avg_salary  |
+--------------+-------------+-------------+-------------+
|       836000 |      250000 |       55000 | 92888.8889 |
+--------------+-------------+-------------+-------------+
1 row in set (0.00 sec)

+-----------------+
| total_employees |
+-----------------+
|               9 |
+-----------------+
1 row in set (0.00 sec)

+-------------+---------------------+
| branch_name | employees_in_branch |
+-------------+---------------------+
| Corporate   |                   2 |
| Scranton    |                   4 |
| Stamford    |                   3 |
+-------------+---------------------+
3 rows in set (0.00 sec)

mysql> Student ID:22211039, Name:Sangwon Lee
```

Week3.sql 실행