

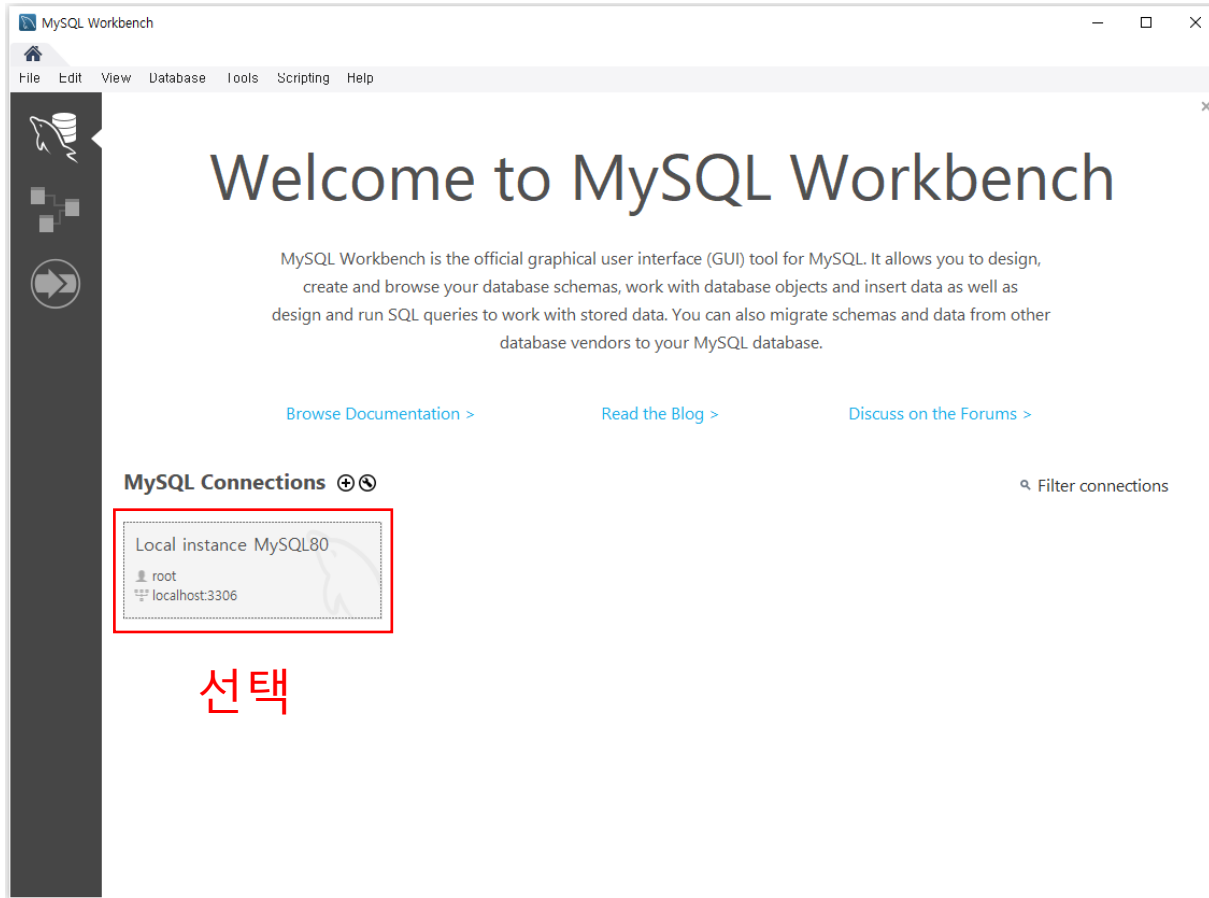
데이터베이스설계 (ICE4016)

실습 4주차

Node.js

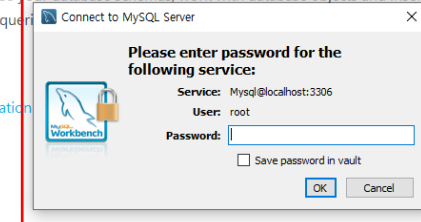
Prof. Wonik Choi

MySQL Workbench



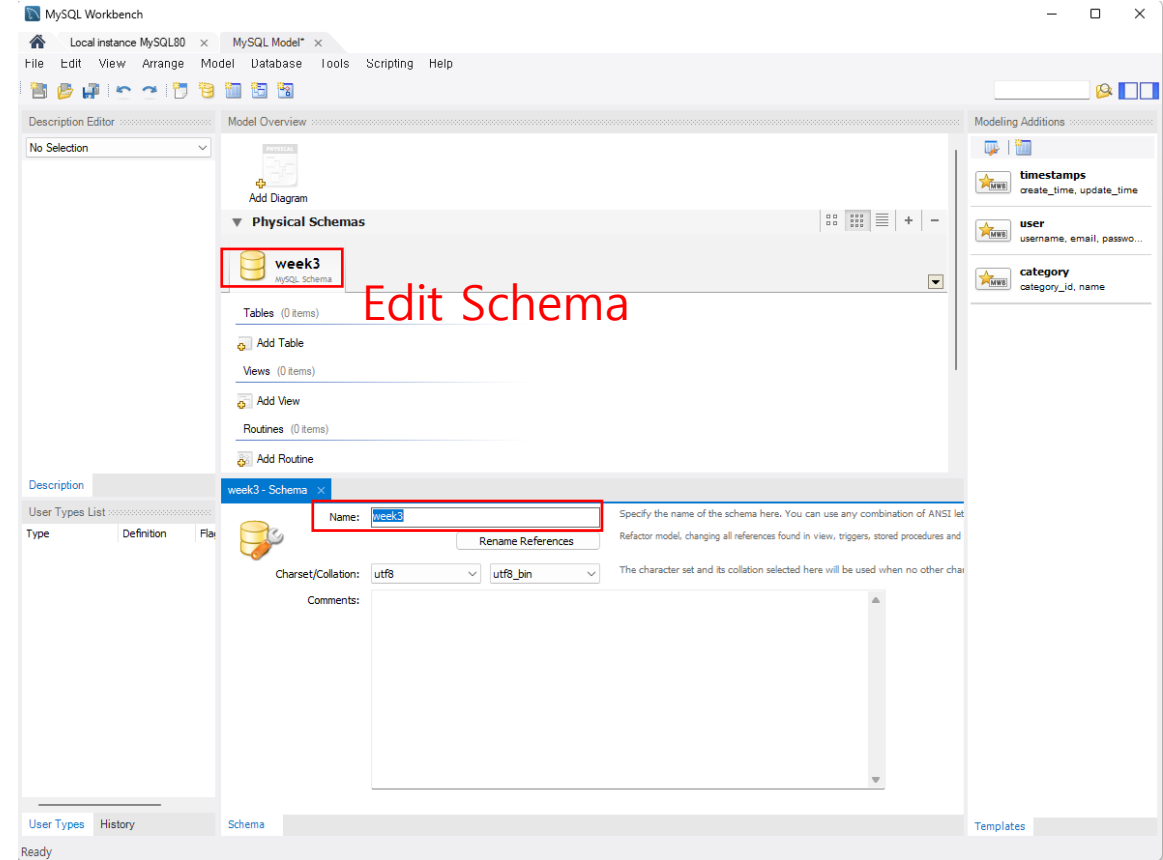
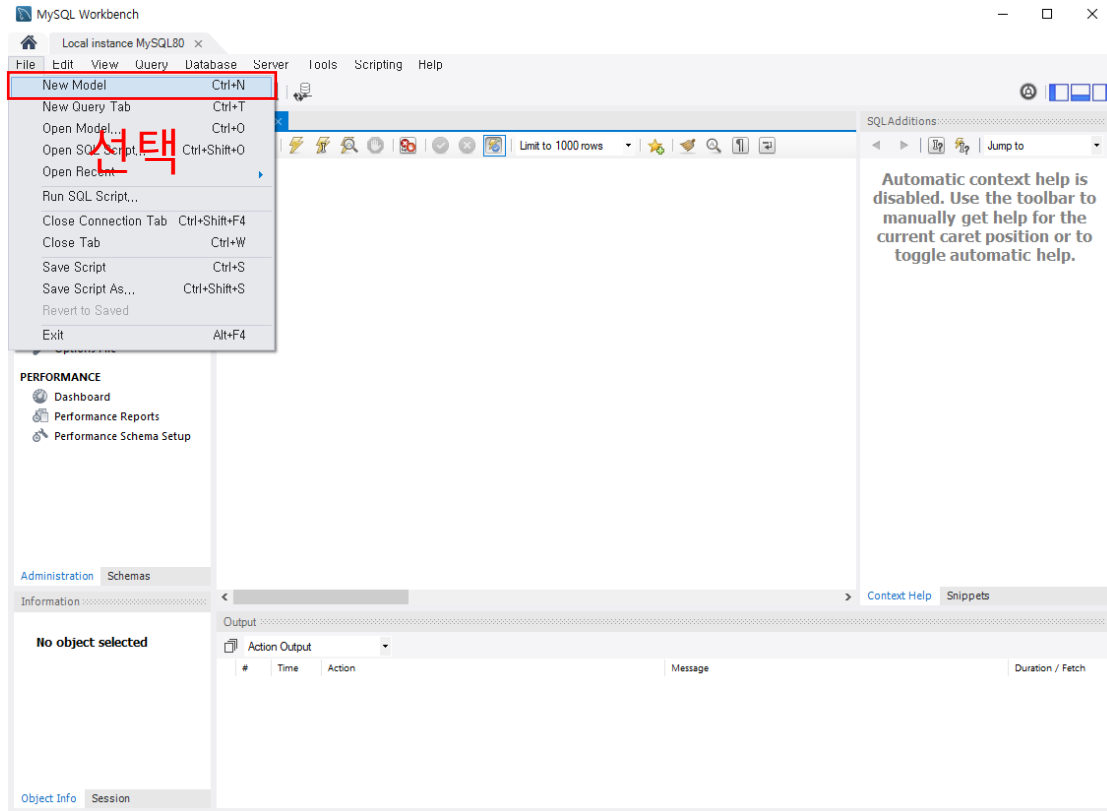
Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

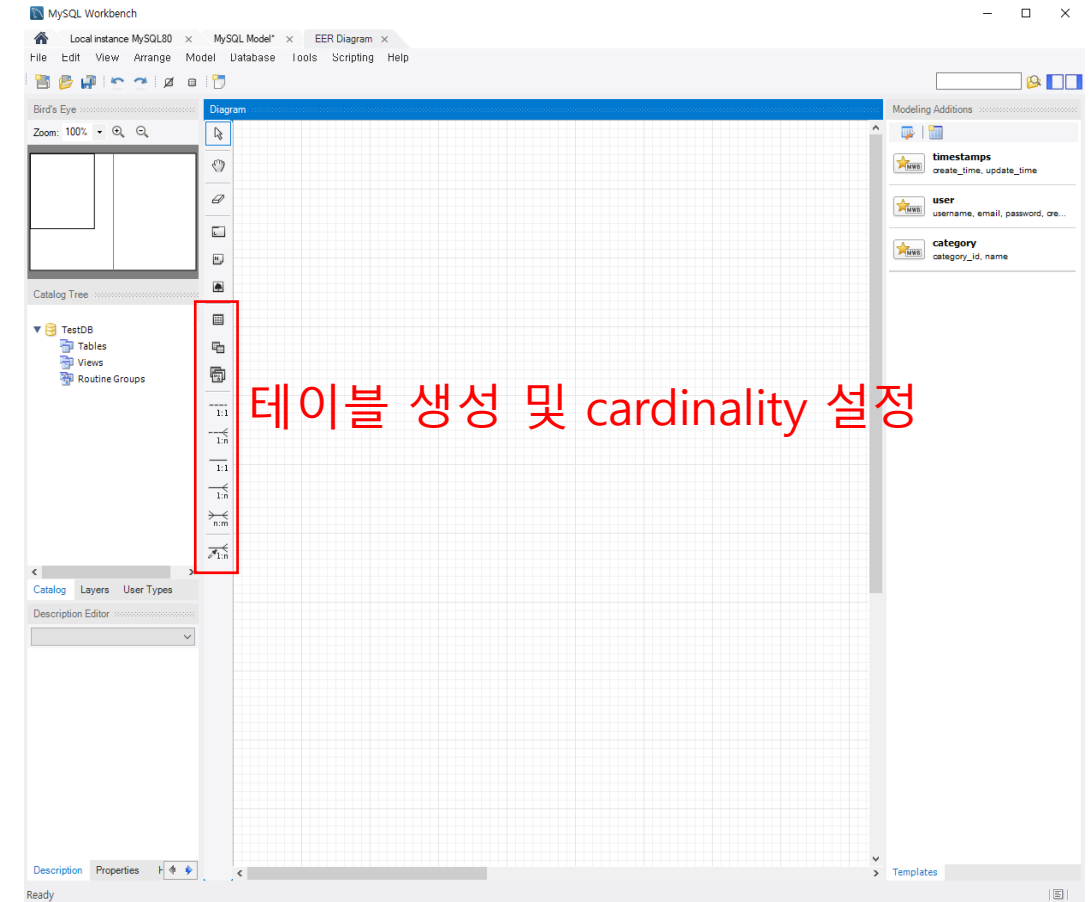
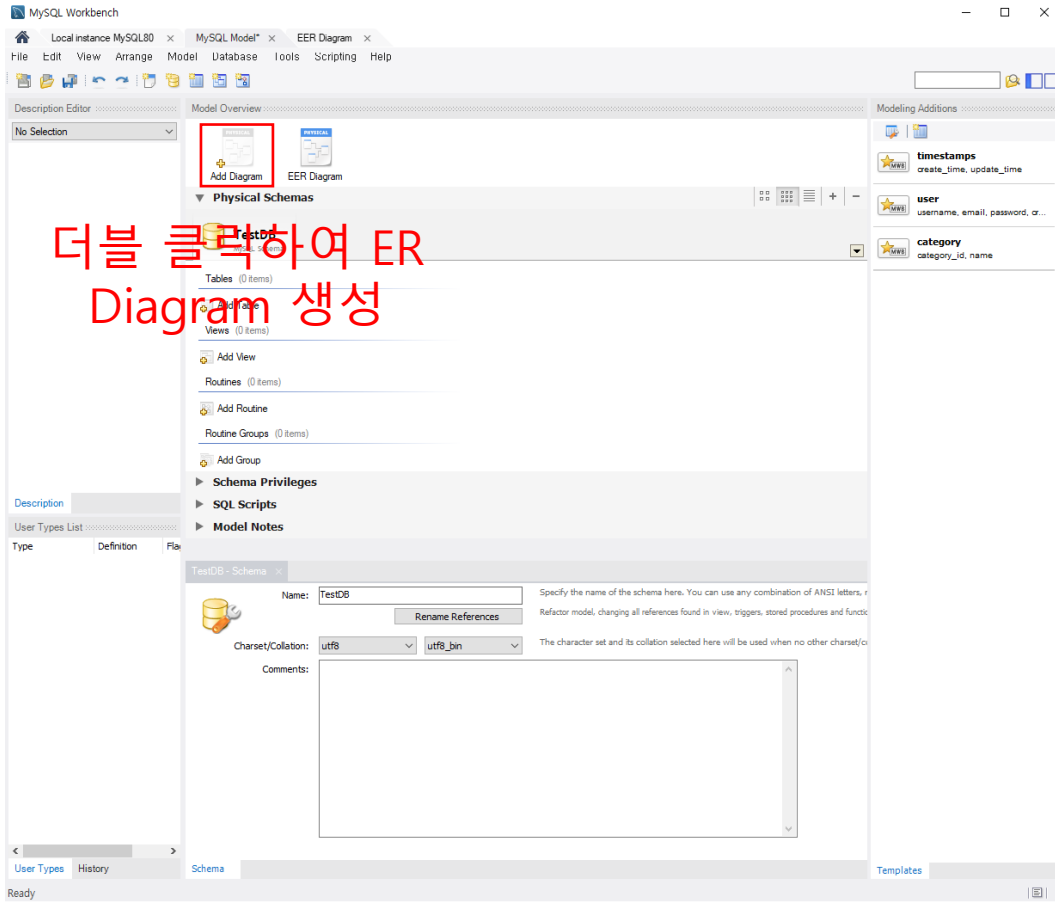


Password 입력하여 로그인

MySQL Workbench

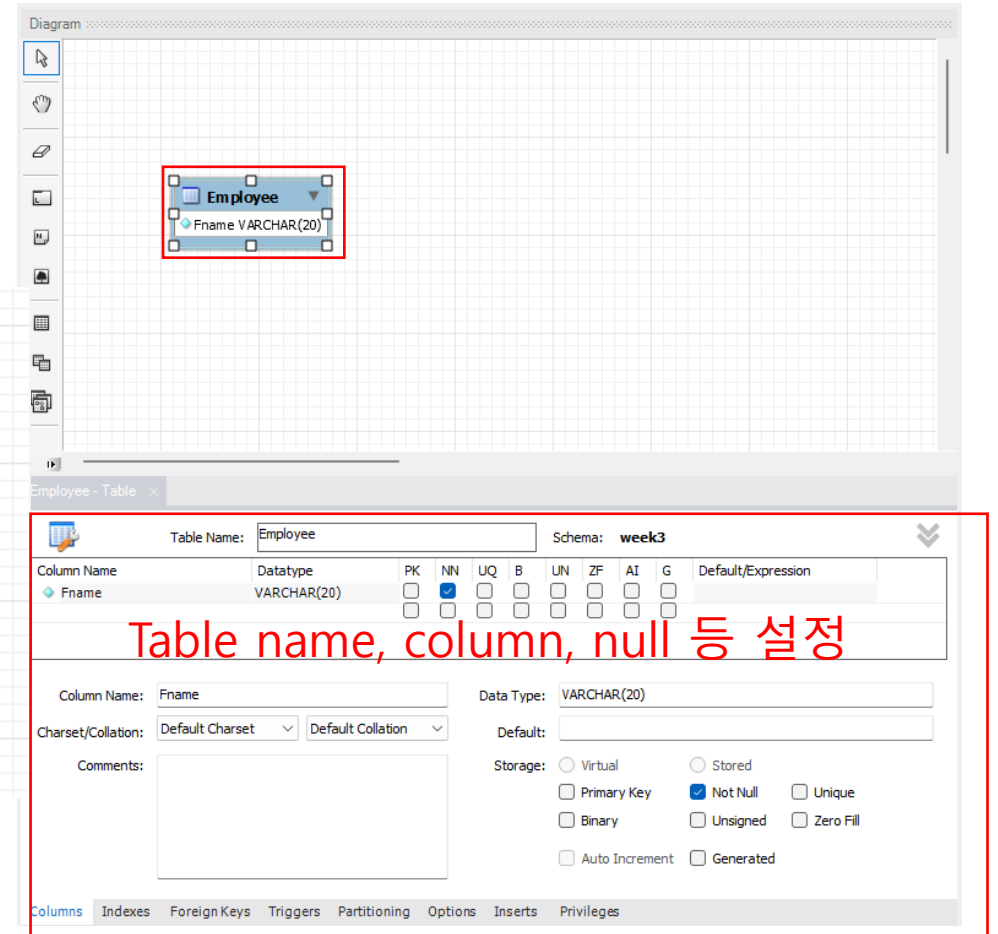
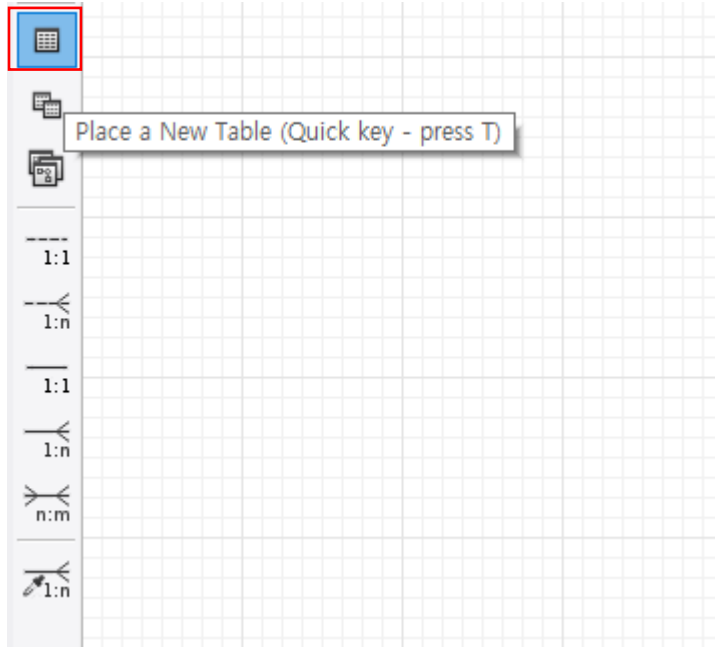


MySQL Workbench

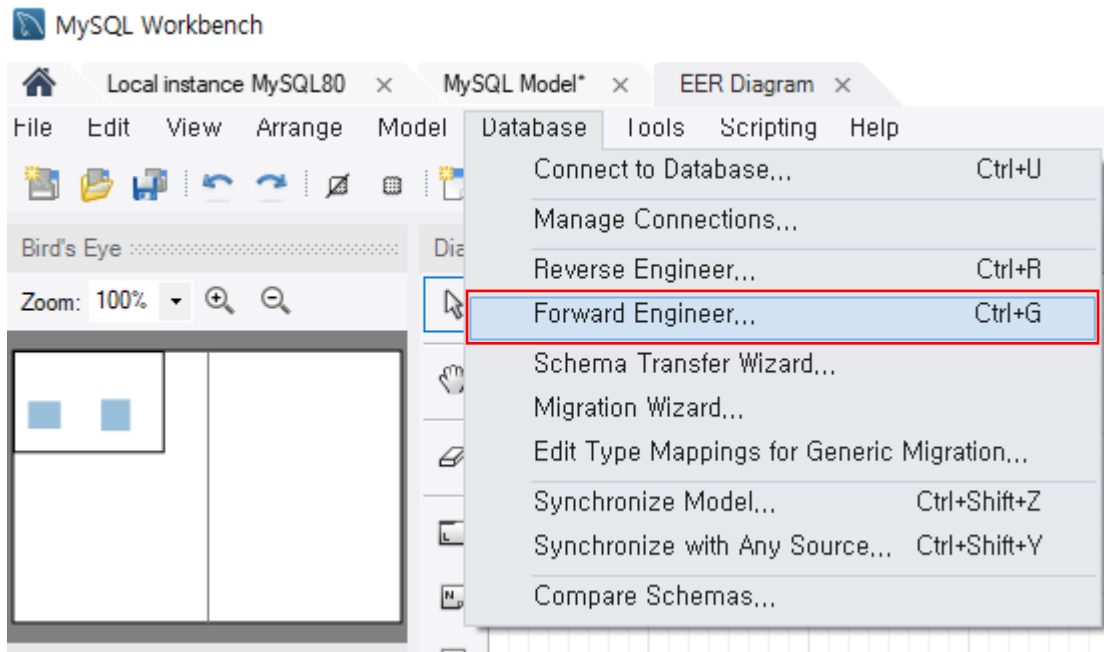


MySQL Workbench

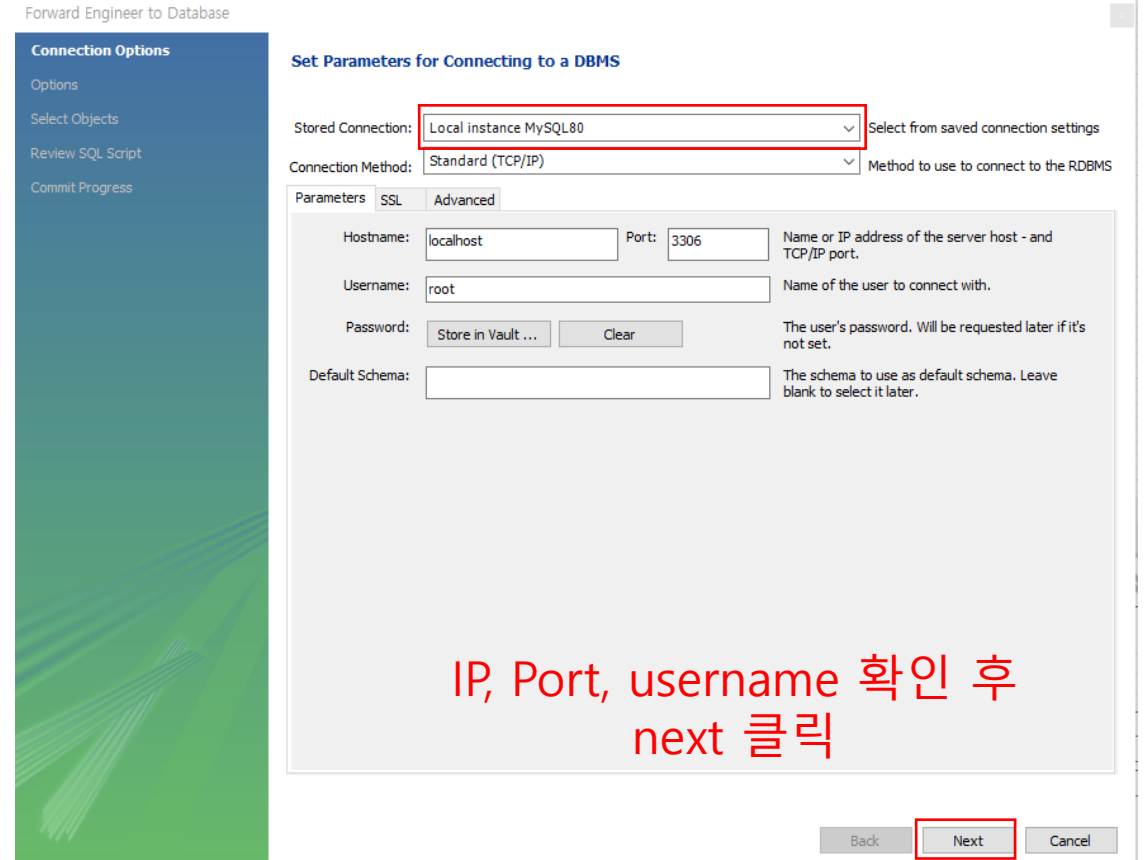
클릭 후 옆의 도면 아무 곳에 클릭



MySQL Workbench



다이어그램을 실제 DB에 반영



MySQL Workbench

Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

Set Options for Database to be Created

Tables

- ☐ Skip creation of FOREIGN KEYS
- ☐ Skip creation of FK Indexes as well
- ☐ Generate separate CREATE INDEX statements
- ☐ Generate INSERT statements for tables
- ☐ Disable FK checks for INSERTs

Other Objects

- ☐ Don't create view placeholder tables
- ☐ Do not create users. Only create privileges (GRANTS)

Code Generation

- ☐ DROP objects before each CREATE object
- ☐ Generate DROP SCHEMA
- ☐ Omit schema qualifier in object names
- ☐ Generate USE statements
- ☐ Add SHOW WARNINGS after every DDL statement
- ☒ Include model attached scripts

Back

Next

Cancel

옵션 선택(실습에서는 사용하지 않음)

Select Objects to Forward Engineer

To exclude objects of a specific type from the SQL Export, disable the corresponding checkbox. Press Show Filter and add objects or patterns to the ignore list to exclude them from the export.



☒ Export MySQL Table Objects

2 Total Objects, 2 Selected

Show Filter



☐ Export MySQL Views

0 Total Objects, 0 Selected

Show Filter



☐ Export MySQL Schemas

0 Total Objects, 0 Selected

Show Filter

☐ Export MySQL Users

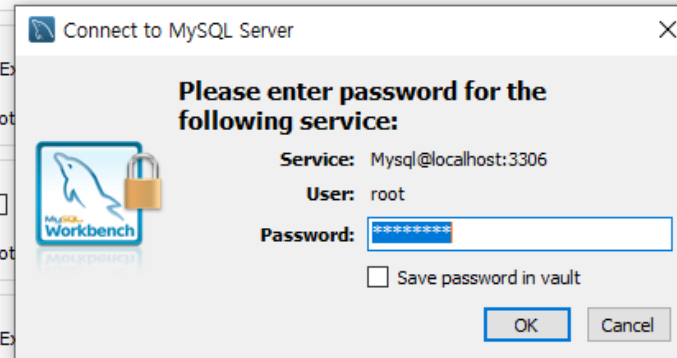
0 Total Objects, 0 Selected

Show Filter

☐ Export MySQL Procedures

0 Total Objects, 0 Selected

Show Filter



Password 입력 후 next

Back

Next

Cancel

MySQL Workbench

Forward Engineer to Database

Connection Options
Options
Select Objects
Review SQL Script
Commit Progress

Review the SQL Script to be Executed

This script will now be executed on the DB server to create your databases.
You may make changes before executing.

```
1  -- MySQL Workbench Forward Engineering
2
3  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
6
7  -----
8  -- Schema week4
9  -----
10
11 -----
12 -- Schema week4
13 -----
14 CREATE SCHEMA IF NOT EXISTS `week4` DEFAULT CHARACTER SET utf8mb4 ;
15 USE `week4` ;
16
17 -----
18 -- Table `week4`.`employee`
19 -----
20 CREATE TABLE IF NOT EXISTS `week4`.`employee` (
21   `Fname` VARCHAR(20) NOT NULL,
22   `Minit` CHAR(1) NULL,
23   `Lname` VARCHAR(20) NOT NULL,
24   `Ssn` CHAR(9) NOT NULL,
25   `Bdate` DATE NULL,
26   `Address` VARCHAR(30) NULL,
27   `Sex` CHAR(1) NULL,
28   `Salary` DECIMAL(5,0) NULL,
29   `Dno` INT NULL,
30   PRIMARY KEY (`Ssn`)
31 ) ENGINE = InnoDB;
32
33 -----
34 -- Table `week4`.`department`
35 -----
36
37 CREATE TABLE IF NOT EXISTS `week4`.`department` (
38   `Dname` CHAR(20) NOT NULL,
39   `Dnumber` INT NOT NULL,
40   `Mgr_Ssn` CHAR(9) NOT NULL,
41   `Mgr_start_date` DATE NULL
```

Save to File... Copy to Clipboard

Forward Engineering Progress

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

- ☒ Connect to DBMS
- ☐ Execute Forward Engineered Script
- ☐ Read Back Changes Made by Server
- ☐ Save Synchronizations

Connecting to DB

Please enter password for the following service:

Service: Mysql@localhost:3306

User: root

Password:

☐ Save password in vault

OK Cancel

Show Logs

Back

Close

Cancel

Back Next Cancel

DB 및 테이블 생성을 위한 script 생성됨

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.

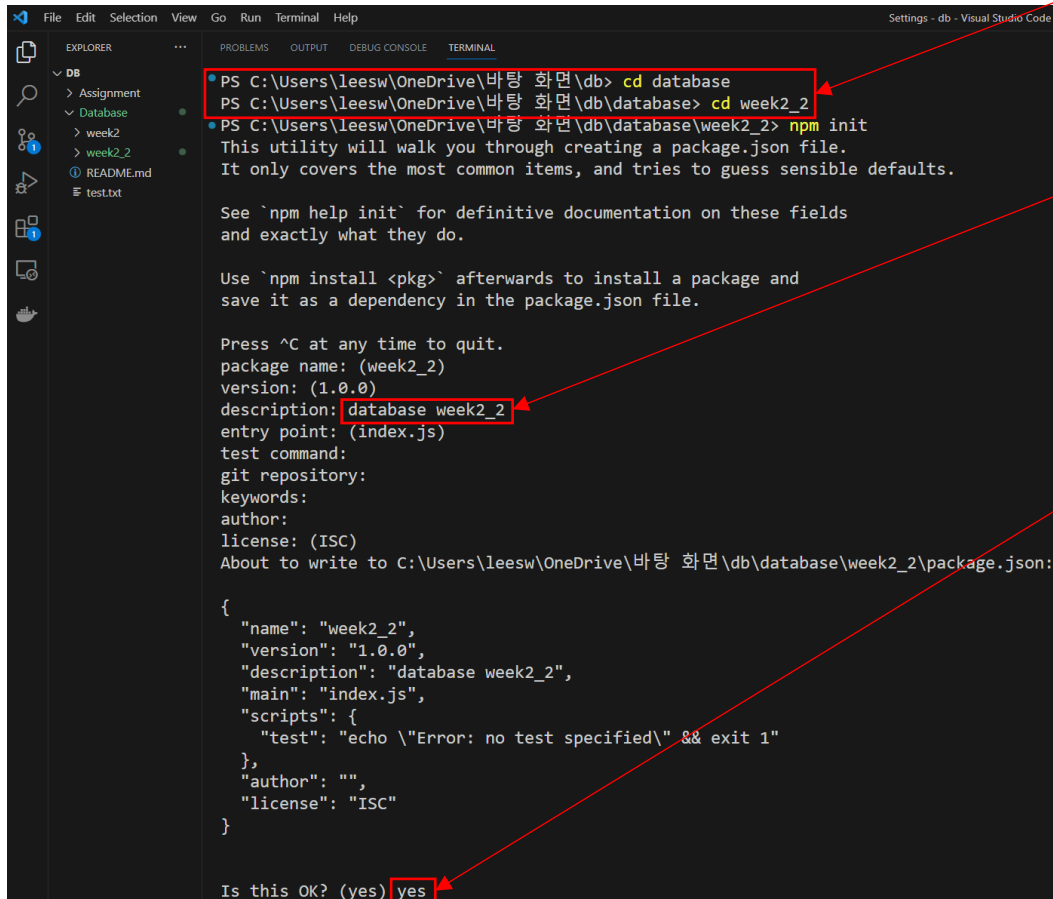
Installation complete. If you want to use this version, type

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



```
PS C:\Users\leesw\OneDrive\바탕 화면\db> cd database
PS C:\Users\leesw\OneDrive\바탕 화면\db\database> cd week2_2
PS C:\Users\leesw\OneDrive\바탕 화면\db\database\week2_2> 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 init` 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: (week2_2)
version: (1.0.0)
description: database week2_2
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\leesw\OneDrive\바탕 화면\db\database\week2_2\package.json:

{
  "name": "week2_2",
  "version": "1.0.0",
  "description": "database week2_2",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes) yes
```

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 transcompiler

○ 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

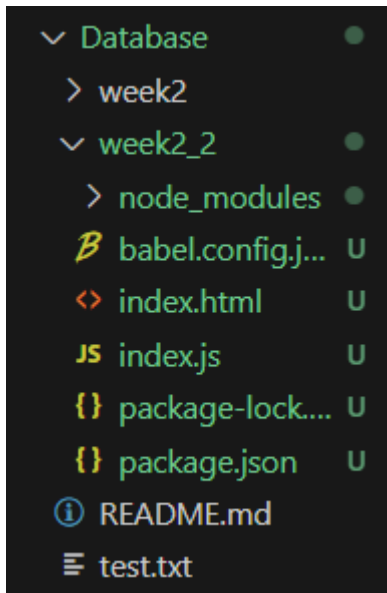
- package.json 파일 수정
 - "scripts" 부분 아래 구문 추가
 - "start": "nodemon --exec babel-node index.js"
 - babel-node: JS 최신 문법 적용
 - nodemon: 변경사항 탐지 및 자동 재시작

```
Database > week2_2 > {} package.json > ...
1  {
2    "name": "week2_2",
3    "version": "1.0.0",
4    "description": "database week2_2",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "start": "nodemon --exec babel-node index.js"
9    },
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "@babel/core": "^7.22.15",
14     "@babel/node": "^7.22.15",
15     "@babel/preset-env": "^7.22.15",
16     "body-parser": "^1.20.2",
17     "express": "^4.18.2",
18     "mysql2": "^3.6.0",
19     "nodemon": "^3.0.1"
20   }
21 }
```

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 > B babel.config.json > ...
1  {
2    |    "presets": ["@babel/preset-env"]
3  }
```

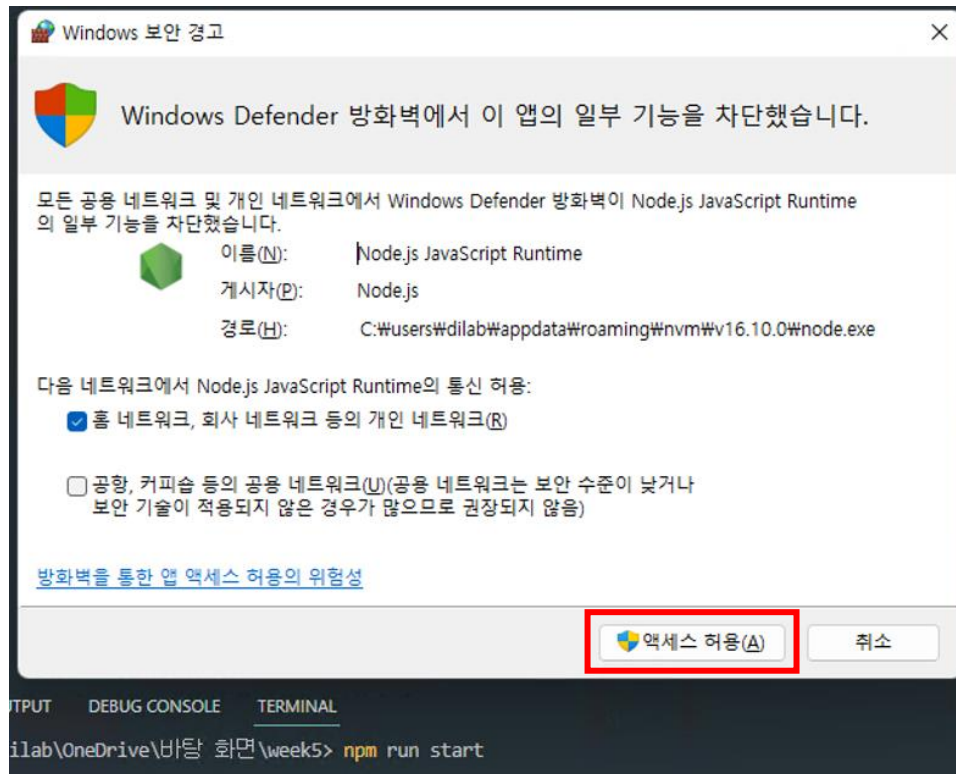
```
Database > week2_2 > <> index.html > html
1  <!DOCTYPE html>
2  <html>
3    <head>
4      |    <title>Database week2_2</title>
5    </head>
6    <body>
7      |    <h1>Hello Express!</h1>
8    </body>
9  </html>
```

```
Database > week2_2 > JS index.js > ...
1  import express from "express";
2  import bodyParser from "body-parser";
3  import path from 'path';
4
5  const app = express();
6
7  app.use(bodyParser.json());
8  app.use(bodyParser.urlencoded({ extended: true }));
9
10 app.get("/", (req, res)=>{
11   |    res.sendFile(path.join(__dirname+'/index.html'));
12   |  });
13
14 app.listen(3000, ()=>{
15   |    console.log("Server is running on port 3000.");
16   |  })
```

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

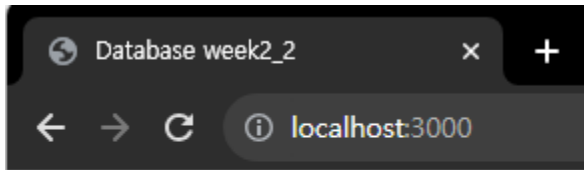
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `babel-node index.js`
Server is running on port 3000.
```


서버 구축

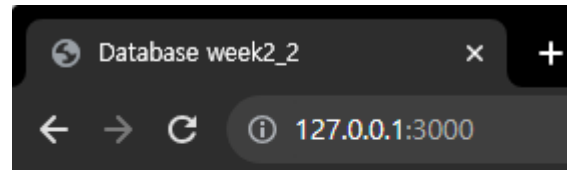
○ 프로젝트 생성 및 환경 세팅

- 서버 실행 확인

- <http://localhost:3000/> 또는 <http://127.0.0.1:3000/>



Hello Express!



Hello Express!

MySQL 연동

○ Express-Mysql 연동

- week2_2 Root 폴더에 파일 생성

- sql.js 파일 생성
- index.js 파일 내용 수정

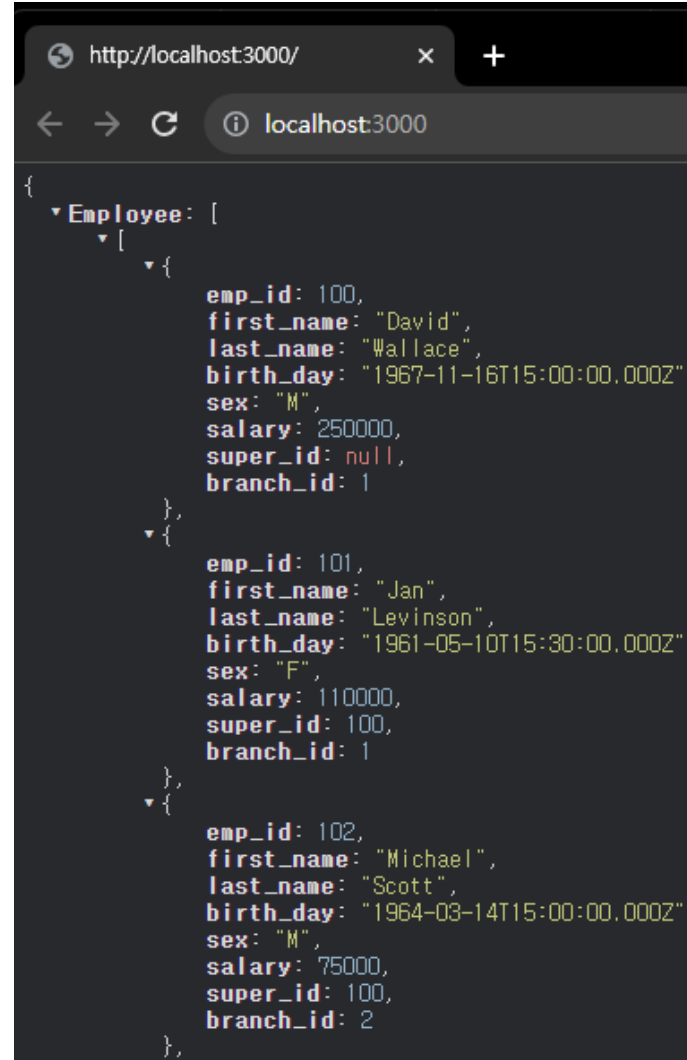
```
1 import mysql from 'mysql2';
2
3 const pool = mysql.createPool({
4   host: 'localhost',
5   port: 3306,
6   user: 'root',
7   password: '', Mysql pw
8   database: 'week3_company',
9 })
10
11 const promisePool = pool.promise();
12
13 const sql = {
14   getEmployee: async()=>{
15     const results = await promisePool.query(`
16       select * from employee
17     `)
18
19     return results;
20   },
21 };
22
23 export default sql;
```

```
1 import express from "express";
2 import bodyParser from "body-parser";
3 import path from 'path';
4 import sql from './sql';
5
6 const app = express();
7
8 app.use(bodyParser.json());
9 app.use(bodyParser.urlencoded({ extended: true }));
10
11 app.get("/", async (req, res)=>{
12   //res.sendFile(path.join(__dirname+'/index.html'));
13   const employee = await sql.getEmployee();
14   res.json({"Employee": employee})
15 });
16
17 app.listen(3000, ()=>{
18   console.log("Server is running on port 3000.");
19 });
```

MySQL 연동

○ Express-Mysql 연동

- 결과 확인



A screenshot of a web browser window with the address bar showing 'http://localhost:3000/'. The browser displays a JSON response from a server. The JSON is an array of employee objects. The first object is for David Wallace, born 1967-11-16, salary 250000, super_id null, branch_id 1. The second object is for Jan Levinson, born 1961-05-10, salary 110000, super_id 100, branch_id 1. The third object is for Michael Scott, born 1964-03-14, salary 75000, super_id 100, branch_id 2.

```
{
  "Employee": [
    {
      "emp_id": 100,
      "first_name": "David",
      "last_name": "Wallace",
      "birth_day": "1967-11-16T15:00:00.000Z",
      "sex": "M",
      "salary": 250000,
      "super_id": null,
      "branch_id": 1
    },
    {
      "emp_id": 101,
      "first_name": "Jan",
      "last_name": "Levinson",
      "birth_day": "1961-05-10T15:30:00.000Z",
      "sex": "F",
      "salary": 110000,
      "super_id": 100,
      "branch_id": 1
    },
    {
      "emp_id": 102,
      "first_name": "Michael",
      "last_name": "Scott",
      "birth_day": "1964-03-14T15:00:00.000Z",
      "sex": "M",
      "salary": 75000,
      "super_id": 100,
      "branch_id": 2
    }
  ]
}
```

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 직원수를 제시하라.
- 각 branch별 근무하는 사원의 수를 검색하여 branch 이름과 소속 직원수를 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