

데이터베이스설계 (ICE4016)

실습 10주차

Web, Login

Prof. Wonik Choi

실습 목표

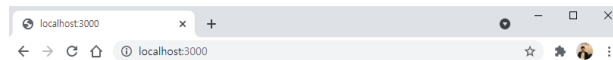
○ 웹 상에서 MySQL 사용하기

- 웹에서 delete, select 명령어 사용
- 라우터 구성
 - '/'(로그인 화면), '/select' (조회 화면),
 - '/delete' (데이터 삭제 화면)

localhost:3000/
(로그인 페이지)

localhost:3000/select
(전체 데이터 조회 페이지)

localhost:3000/delete
(데이터 선택 삭제 페이지)

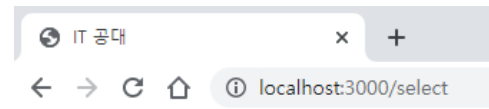


로그인

아이디

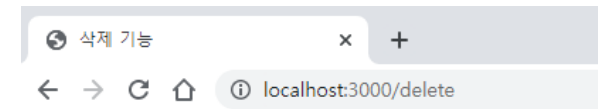
비밀번호

로그인



IT 공대

Dname	Dnumber
전기공학과	2
전자공학과	3
정보통신공학과	0
컴퓨터공학과	1



삭제 기능

Dname	Dnumber	
전기공학과	2	삭제
전자공학과	3	삭제
정보통신공학과	0	삭제
컴퓨터공학과	1	삭제

실습 10주차

○ 데이터베이스 생성

- 실습10주차 DB생성 파일 내부의 명령어 실행 후 아래 값 확인
 - MySQL에서 "week10" 데이터베이스를 생성한 후 "user", "department" 테이블 생성
 - 하단의 이미지와 같이 값 입력된 것 확인

```
mysql> select * from user;
+-----+-----+-----+
| Id      | Password | Role   |
+-----+-----+-----+
| 12000000 | test1234 | student |
| admin    | admin1234 | super  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

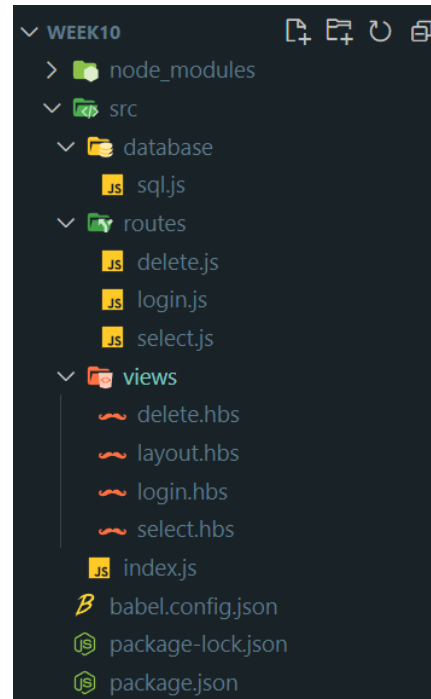
```
mysql> select * from department;
+-----+-----+-----+
| ID | Dname          | Dnumber |
+-----+-----+-----+
| 3  | 전기공학과     | 2       |
| 4  | 전자공학과     | 3       |
| 1  | 정보통신공학과 | 0       |
| 2  | 컴퓨터공학과   | 1       |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

실습 10주차

프로젝트 수정

- Week10 폴더 생성

- 내부 구조(파일 이름)는 아래 그림과 같게 수정



실습 10주차

프로젝트 수정

프로젝트 소스

index.js

```
index.js
src > index.js > ...
14
15 import express from "express";
16 import logger from "morgan";
17 import path from "path";
18
19 import loginRouter from "../routes/login";
20 import selectRouter from "../routes/select";
21 import deleteRouter from "../routes/delete";
22
23 const PORT = 3000;
24
25 const app = express();
26
27 app.use(express.urlencoded({ extended: false }));
28 app.use(express.json());
29
30 app.set("views", path.join(__dirname, "views"));
31 app.set("view engine", "hbs");
32
33 app.use(logger("dev"));
34
35 app.use("/", loginRouter);
36 app.use("/select", selectRouter);
37 app.use("/delete", deleteRouter);
38
39 app.listen(PORT, () => {
40   console.log(`Example app listening at http://localhost:${PORT}`);
41 });
42
```

sql.js

```
sql.js
src > database > sql.js > ...
1 import mysql from "mysql2";
2
3 // 데이터베이스 연결
4 const pool = mysql.createPool({
5   process.env.JAWSDB_URL ?? {
6     host: "localhost",
7     user: "root",
8     database: "week10",
9     password: "dila1234",
10    waitForConnections: true,
11    connectionLimit: 10,
12    queueLimit: 0,
13   }
14 });
15
16 // async / await 사용
17 const promisePool = pool.promise();
18
19 // select query
20 export const selectSql = {
21   getUsers: async () => {
22     const [rows] = await promisePool.query(`select * from user`);
23
24     return rows;
25   },
26   getDepartment: async () => {
27     const [rows] = await promisePool.query(`select * from department`);
28
29     return rows;
30   },
31 };
32
33 // delete query
34 export const deleteSql = {
35   deleteDepartment: async (data) => {
36     console.log("deleteSql.deleteDepartment:", data.Dnumber);
37     const sql = `delete from department where Dnumber=${data.Dnumber}`;
38
39     await promisePool.query(sql);
40   },
41 };
42
```



실습 10주차

프로젝트 수정

프로젝트 소스

login.js

```
login.js
src > routes > login.js > ...
14
15 import express from "express";
16 import { selectSql } from "../database/sql";
17
18 const router = express.Router();
19
20 router.get('/', (req, res) => {
21   res.render('login');
22 });
23
24 router.post('/', async (req, res) => {
25   const vars = req.body;
26   const users = await selectSql.getUsers();
27   let whoAmI = '';
28   let checkLogin = false;
```

```
users.map((user) => {
  console.log(user.Id);
  if (vars.id === user.Id && vars.password === user.Password) {
    console.log('login success!');
    checkLogin = true;
    if (vars.id === 'admin') {
      whoAmI = 'admin';
    } else {
      whoAmI = 'user';
    }
  }
});

if (checkLogin && whoAmI === 'admin') {
  res.redirect('/delete');
} else if (checkLogin && whoAmI === 'user') {
  res.redirect('/select');
} else {
  console.log('login failed!');
  res.send("<script>alert('로그인에 실패했습니다.');" location.href='/';</script>");
}

module.exports = router;
```

select.js

```
select.js
src > routes > select.js > router.get('/') callback
14
15 import express from "express";
16 import { selectSql } from "../database/sql";
17
18 const router = express.Router();
19
20 router.get('/', async function(req, res) {
21   const department = await selectSql.getDepartment();
22
23   res.render('select', {
24     title: 'IT 공대',
25     department
26   });
27 });
28
29 module.exports = router;
```

실습 10주차

프로젝트 수정

프로젝트 소스

delete.js

```
delete.js
src > routes > delete.js > ...
15 import express from "express";
16 import { selectSql, deleteSql } from "../database/sql";
17
18 const router = express.Router();
19
20
21 // 기존의 입력 값 불러오기
22 router.get('/', async (req, res) => {
23   const department = await selectSql.getDepartment();
24
25   res.render('delete', {
26     title: "삭제 기능",
27     department
28   });
29 });
30
31
32 // 삭제 버튼을 눌렀을 경우 update query를 실행하며 조회 페이지로 이동
33 router.post('/', async (req, res) => {
34   console.log('delete router:', req.body.delBtn);
35
36   const data = {
37     Dnumber: req.body.delBtn,
38   };
39
40   await deleteSql.deleteDepartment(data);
41
42   res.redirect('/delete');
43 });
44
45 module.exports = router;
```

login.hbs

```
<style>
  body {
    height: 75vh;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
  }
  .frame {
    border-radius: 15px;
    box-sizing: border-box;
    border: 1px solid #757575;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    width: 350px;
  }
  form {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
  }
  #id, #passwd, .btn {
    font-size: 25px;
  }
  .pwd {
    margin-top: 15px;
  }
  .btn {
    width: 322px;
    margin-top: 20px;
    margin-bottom: 20px;
  }
</style>

<div class="frame">
  <h1>로그인</h1>
  <form id="deparment" method="post" action="/">
    <div class="id">
      <input id="id" name="id" type="text" required placeholder="아이디">
    </div>
    <div class="pwd">
      <input id="passwd" name="password" type="password" required placeholder="비밀번호">
    </div>
    <button class="btn" type="submit">로그인</button>
  </form>
</div>
```



실습 10주차

○ 프로젝트 수정

- 프로젝트 소스

layout.hbs

```
<!DOCTYPE html>
<html>
  <head>
    <title>{{title}}</title>
    {{!-- <link rel='stylesheet' href='/stylesheets/style.css' /> --}}
    <style type="text/css">
      table{border-collapse:collapse}
      th,td{border:1px solid black; width: 150px; height: 30px}
    </style>
  </head>
  <body>
    {{{body}}}
  </body>
</html>
```

select.hbs

```
<h1>{{ title }}</h1>

<table>
  <tr>
    <td>Dname</td>
    <td>Dnumber</td>
  </tr>
  {{#each department}}
    <tr>
      <td>{{Dname}}</td>
      <td>{{Dnumber}}</td>
    </tr>
  {{/each}}
</table>
```

delete.hbs

```
<h1>{{ title }}</h1>

<table>
  <tr>
    <td>Dname</td>
    <td>Dnumber</td>
  </tr>
  {{#each department}}
    <form method="post">
      <tr>
        <td>{{Dname}}</td>
        <td>{{Dnumber}}</td>
        <td style="border: none; margin-left: 10px;">
          <button
            style="margin-left: 10px;"
            name='delBtn'
            type="submit"
            value="{{Dnumber}}"
            formaction="/delete"
          >
            삭제
          </button>
        </td>
      </tr>
    </form>
  {{/each}}
</table>
```


실습 10주차

○ 서버 실행

- 콘솔창에서 “npm run start” 명령어로 서버 실행하여 기능 확인

• 웹 브라우저에서 localhost:3000 접속

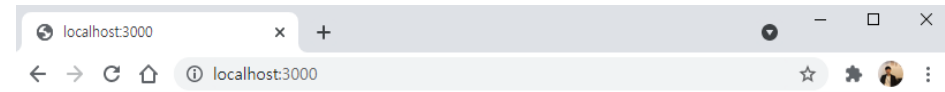
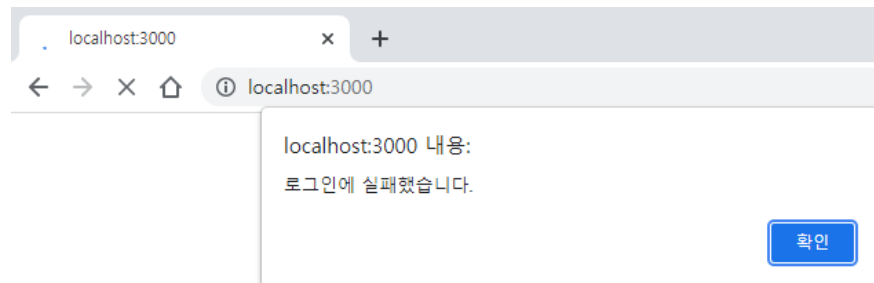
➤ 로그인 화면 확인

▪ DB에 저장된 유저 정보 확인

• Admin 계정 또는 일반 사용자 계정

▪ 로그인 성공 시 화면 이동

▪ 로그인 실패 시 경고 창 팝업

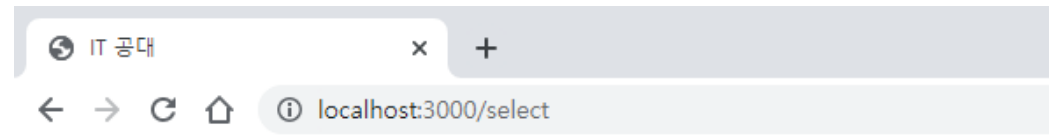


로그인

실습 10주차

○ 서버 실행

- 콘솔창에서 "npm run start" 명령어로 서버 실행하여 기능 확인
 - 웹 브라우저에서 localhost:3000/select 접속
 - Department 테이블 데이터 조회



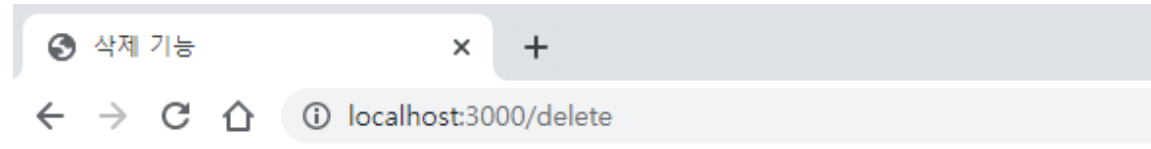
IT 공대

Dname	Dnumber
전기공학과	2
전자공학과	3
정보통신공학과	0
컴퓨터공학과	1

실습 10주차

○ 서버 실행

- 콘솔창에서 "npm run start" 명령어로 서버 실행하여 기능 확인
 - 웹 브라우저에서 localhost:3000/delete 접속
 - 삭제 버튼을 클릭하여 해당 row 데이터 삭제



삭제 기능

Dname	Dnumber	
전기공학과	2	삭제
전자공학과	3	삭제
정보통신공학과	0	삭제
컴퓨터공학과	1	삭제

실습 10주차 : 최종 목표

- STEP 1 : Inha DB에 로그인 기능 구현 적용
 - 학생의 학번 및 비밀번호로 로그인 하는 기능 구현할 것
 - 학생은 학생 정보를 조회할 수 있는 페이지로 이동
- STEP 2 : Class 테이블에 데이터 삽입하여 delete 결과 확인
 - 학생으로 로그인 한 경우에만 수강하는 수업(class)을 수강취소(삭제) 할 수 있도록 구현
- 보고서에 반드시 포함될 내용 (hwp/doc화일)
 - 소스코드 및 보고서는 하나의 폴더에 압축하여 업로드(.zip / .tar / .gz 등)