Adzkia Husnul Abidat – 1506757680
APAP B

Write Up 05

Pada tutorial minggu ini saya mempelajari penggunaan database dengan query yang sedikit lebih kompleks, dimana ada relasi antara tabel student dengan tabel courses. Mempelajari bagaimana cara melakukan join antara tabel studentcourse dengan course berdasarkan id_course dan difilter hanya pada student dengan npm yang telah diberikan. Kemudian saya berlatih membuat method untuk mengubah SelectAllStudents dan menambahkan view pada Course. Dengan annotations Result dan Results saya dapat memetakan query ke class model.

Latihan

1. SelectAllStudents

Menghubungkan method selectAllStudent dengan method selectCourses pada **StudentMapper**

```
@Select("select npm, name, gpa from student")
@Results(value = {
        @Result(property="npm", column="npm"),
        @Result(property="name", column="name"),
        @Result(property="courses", column="npm",
        javaType = List.class,
        many=@Many(select="selectCourses"))
})
List<StudentModel> selectAllStudents ();
```

Menambahkan tampilan kuliah yang diambil pada viewall.html

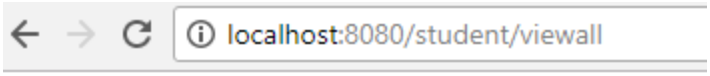
```
<div th:each="student,iterationStatus: ${students}">
    <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
    <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
    <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
    <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
    <a th:href="'/student/delete/' + ${student.npm}">Delete Data</a><br/>
    <a th:href="'/student/update/' + ${student.npm}">Update Data</a><br/>

    <h3>Kuliah yang diambil</h3>
    <ul th:each="course,iterationStatus: ${student.courses}">
    <li th:text="${course.name} + '-' + ${course.credits} + 'sks'">
    Nama kuliah-x SKS
    </li>
    </ul>

    <hr/>
</div>
```

Ketika dijalankan viewall akan tampil sebagai berikut

Adzkia Husnul Abidat – 1506757680
APAP B



No. 2

NPM = 123

Name = Martabak

GPA = 4.0

Delete Data
Update Data

Kuliah yang diambil

- MPKT-6sks

No. 3

NPM = 124

Name = Chanek Jr

GPA = 3.0

Delete Data
Update Data

Kuliah yang diambil

- PSP-4sks
- SDA-3sks

2. **Membuat View pada Course**

- Membuat class CourseModel seperti yang telah dilakukan pada step sebelumnya. Berisi atribut dan constructor yang dimiliki oleh suatu course

- Membuat CourseMapper untuk memetakan query database dengan Method SelectCourse untuk mengambil course sesuai ID dan selectParticipants yang mengembalikan list of student pada Course dengan ID tertentu. Methodnya adalah sebagai berikut:

```java
package com.example.dao;

import java.util.List;

import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Param;
import org.apache.ibatis.annotations.Select;
import org.apache.ibatis.annotations.Result;
import org.apache.ibatis.annotations.Results;
import org.apache.ibatis.annotations.Many;

import com.example.model.CourseModel;
import com.example.model.StudentModel;
@Mapper
public interface CourseMapper {
    @Select("select id_course, name, credits from course where id_course = #{id_course}")
  @Results(value = {
    @Result(property="id_course", column="id_course"),
    @Result(property="name", column="name"),
    @Result(property="credits", column="credits"),
    @Result(property="students", column="id_course",
    javaType = List.class,
    many = @Many(select = "selectParticipants"))
  })
  CourseModel selectCourse(@Param("id_course") String id_course);

  @Select("select student.name, studentcourse.npm " + "from studentcourse join student " +
  "on studentcourse.npm = student.npm " + "where studentcourse.id_course = #{id_course}")
  List <StudentModel> selectParticipants (@Param("id_course") String id_course);
}
```

- Membuat interface CourseService.java pada package **com.example.service**, dengan isi sebagai berikut

```java
package com.example.service;

import com.example.model.CourseModel;

public interface CourseService {
    CourseModel selectCourse(String id);
}
```

- Pada package yang sama, membuat class CourseServiceDatabase yang meng-*implements* CourseService dengan isi sebagai berikut

Adzkia Husnul Abidat – 1506757680
APAP B

```java
package com.example.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.example.dao.CourseMapper;
import com.example.model.CourseModel;

import lombok.extern.slf4j.Slf4j;

@Slf4j
@Service
public class CourseServiceDatabase implements CourseService{
    @Autowired
    private CourseMapper courseMapper;

    @Override
    public CourseModel selectCourse(String id) {
        log.info ("select course with id_course {}", id);
        return courseMapper.selectCourse(id);
    }

}
```

Membuat class **CourseController**, untuk melakukan requestMapping dari url yang dijalankan beserta keluarannya, dengan isi sebagai berikut :

Adzkia Husnul Abidat – 1506757680
APAP B

```java
package com.example.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;

import com.example.model.CourseModel;
import com.example.service.CourseService;
@Controller
public class CourseController {
    @Autowired
    CourseService courseDAO;

    @RequestMapping("/course/view/{id}")
    public String selectCourse (Model model,
            @PathVariable(value = "id") String id)
    {
        CourseModel course = courseDAO.selectCourse(id);

        if (course != null) {
            model.addAttribute ("course", course);
            return "viewcourses";
        } else {
            model.addAttribute ("id", id);
            return "nocourse";
        }
    }
}
```

Membuat tampilan pada viewcourses.html yang akan muncul apabila requestMapping dengan ID yang diminta berhasil diambil

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Courses by ID</title>
    </head>
    <body>
        <h3 th:text="'ID = ' + ${course.id_course}">Course ID</h3>
        <h3 th:text="'Nama = ' + ${course.name}">course Name</h3>
        <h3 th:text="'SKS = ' + ${course.credits}">credits</h3>

        <h3>Mahasiswa yang mengambil</h3>
        <ul th:each="student,iterationStatus: ${course.students}">
            <li th:text="${student.npm} + '-' + ${student.name}">
            Nama mahasiswa
            </li>
        </ul>
    </body>
</html>
```

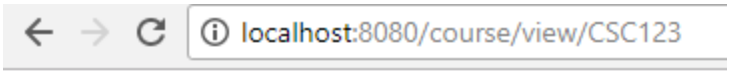
Membuat tampilan pada nocourse.html untuk menampilkan pesan bahwa id course yang dicari tidak ada

Adzkia Husnul Abidat – 1506757680
APAP B

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Course not found</title>
    </head>
    <body>
        <h1>Course not found</h1>
        <h3 th:text="'ID = ' + ${id}">Course ID</h3>
    </body>
</html>
```

Tampilan ketika dijalankan akan muncul sebagai berikut

- ID Course CSC123



ID = CSC123

Nama = PSP

SKS = 4

Mahasiswa yang mengambil

- 124-Chanek Jr

- ID course tidak ditemukan

Course not found

ID = CSC1230