

mybatis resultMap

학습목표

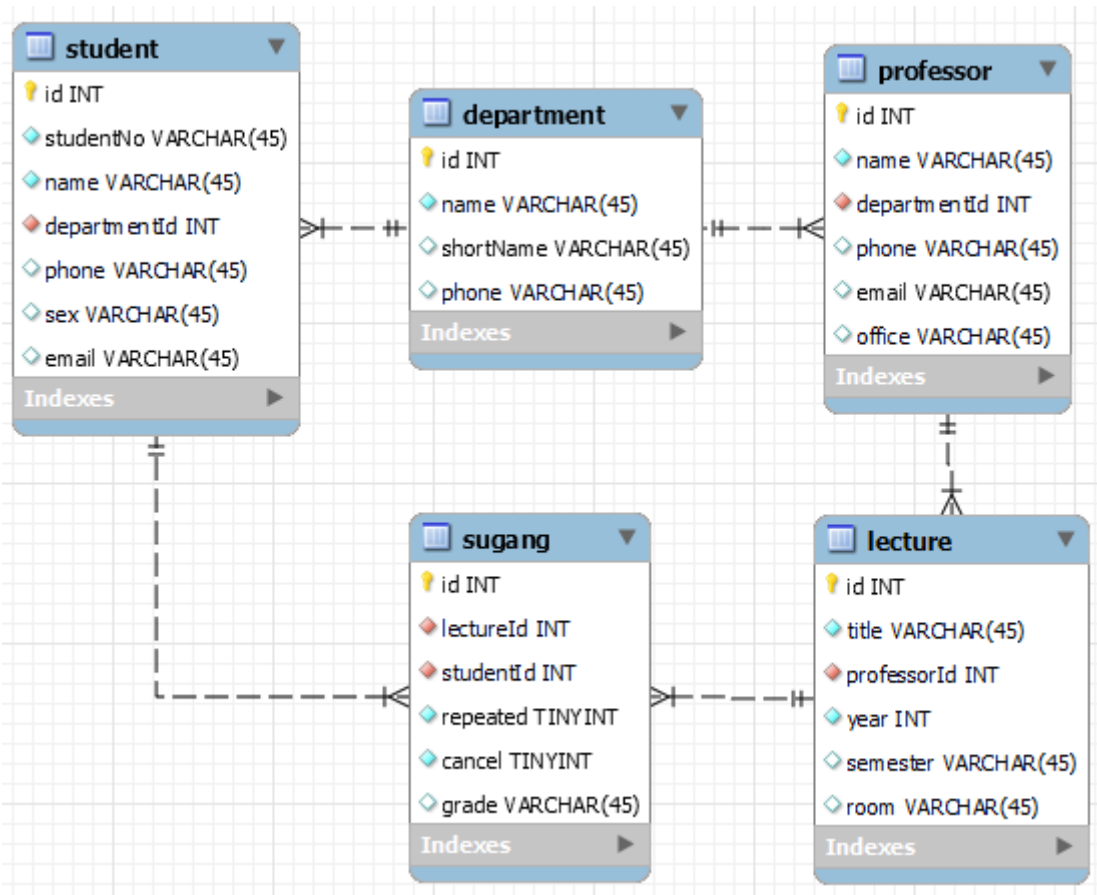
mybatis resultMap 기능을 학습한다

목차

1. 배경 지식	2
1) student2 데이터베이스	2
2) SQL 명령	3
3) mysql true/false 필드의 타입	3
2. mybatis7 프로젝트	4
1) 개요	4
2) 프로젝트 생성	4
3) application.properties	6
4) Sugang DTO 클래스	7
5) SugangMapper.java	8
6) SugangController.java 구현	9
7) common.css	10
8) sugnag/list.html	11
9) 실행	12
10) 평가	12
3. mybatis8 프로젝트	13
1) mybatis resultMap 기능	13
2) mybatis mapper 구현 방법	13
3) FQN (Full Qualified Name)	13
4) 프로젝트 생성	14
5) application.properties	16
6) Student DTO 클래스	17
7) Lecture DTO 클래스	17
8) Sugang DTO 클래스	18
9) SugangMapper.java	19
10) SugangMapper.xml	20
11) SugangController.java	23
12) common.css	24
13) sugnag/list.html	25
4. 연습문제	26
1) 답	27

1. 배경 지식

1) student2 데이터베이스



sugang 테이블

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
lectureId	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
studentId	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
repeated	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cancel	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
grade	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

수강신청 데이터가 들어있다.

필드명	설명
id	자동으로 일련번호가 부여되는 (auto increment) 기본키(primary key)
lectureId	강좌 id 외래키
studentId	학생 id 외래키
repeated	재수강
cancel	수강 취소
grade	학점

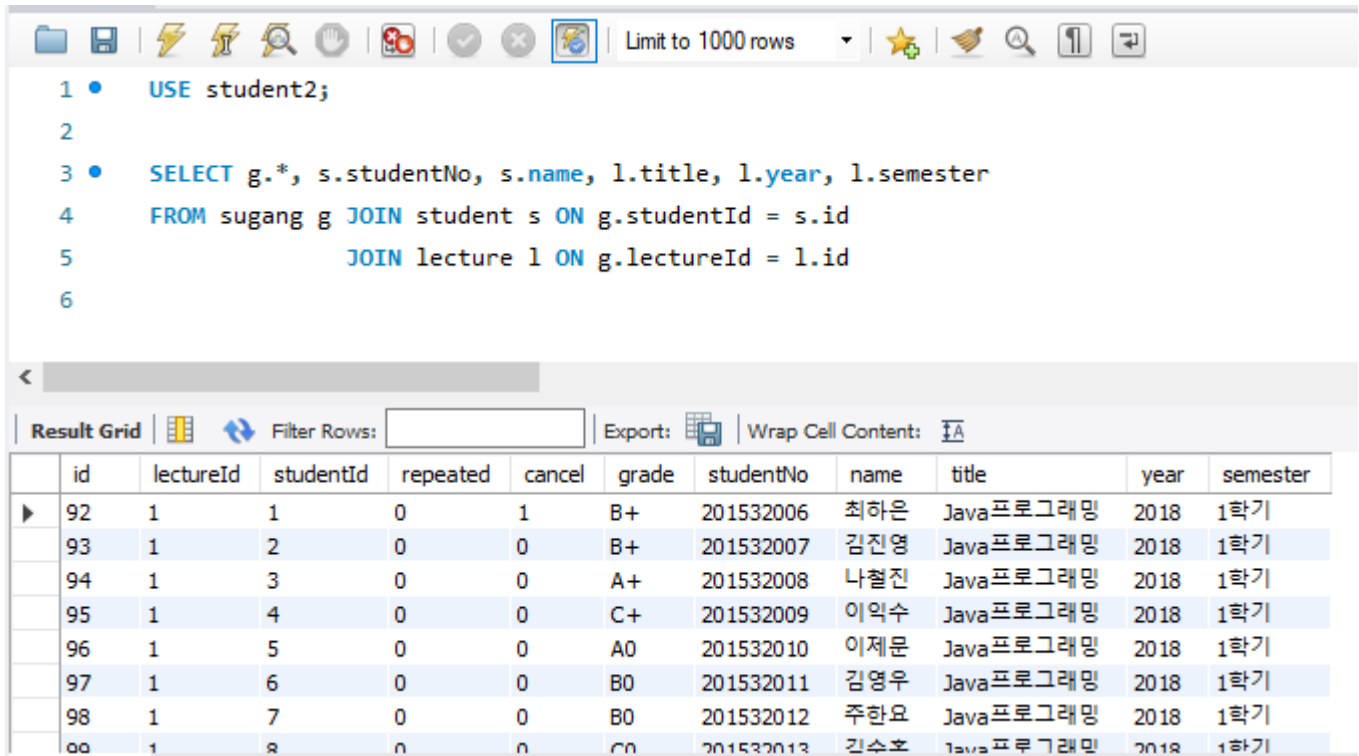
sugang 테이블의 레코드 한 개는

- studentId 외래키가 가르키는 한 학생이
- lectureId 외래키가 가르키는 한 강좌를 수강하고 있음을 뜻 함.

2) SQL 명령

```
USE student2;
```

```
SELECT g.*, s.studentNo, s.name, l.title, l.year, l.semester  
FROM sugang g JOIN student s ON g.studentId = s.id  
      JOIN lecture l ON g.lectureId = l.id
```



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • USE student2;  
2  
3 • SELECT g.*, s.studentNo, s.name, l.title, l.year, l.semester  
4     FROM sugang g JOIN student s ON g.studentId = s.id  
5         JOIN lecture l ON g.lectureId = l.id  
6
```

Below the editor is a 'Result Grid' tab. It shows a table with 12 columns: id, lectureId, studentId, repeated, cancel, grade, studentNo, name, title, year, semester. The data is as follows:

	id	lectureId	studentId	repeated	cancel	grade	studentNo	name	title	year	semester
▶	92	1	1	0	1	B+	201532006	최하은	Java프로그래밍	2018	1학기
	93	1	2	0	0	B+	201532007	김진영	Java프로그래밍	2018	1학기
	94	1	3	0	0	A+	201532008	나철진	Java프로그래밍	2018	1학기
	95	1	4	0	0	C+	201532009	이익수	Java프로그래밍	2018	1학기
	96	1	5	0	0	A0	201532010	이제문	Java프로그래밍	2018	1학기
	97	1	6	0	0	B0	201532011	김영우	Java프로그래밍	2018	1학기
	98	1	7	0	0	B0	201532012	주한요	Java프로그래밍	2018	1학기
	99	1	8	0	0	C0	201532013	김수호	Java프로그래밍	2018	1학기

조회 결과 컬럼

sugang 테이블 (id, lectureId, studentId, repeated, cancel, grade)

student 테이블 (studentNo, name)

lecture 테이블 (title, year, semester)

위 SQL 명령을 mybatis mapper 메소드에 구현하자.

위 조회 결과 데이터를 채울 DTO 클래스를 구현하자.

3) mysql true/false 필드의 타입

true/false를 저장하는 boolean 타입이 mysql에 없다.

그래서 boolean 타입 대신 TINYINT 타입을 사용하여, true/false 대신 1/0을 저장해야 한다.

TINYINT 타입은 8bit 정수이다.

sugang 테이블의 repeated 필드, cancel 필드의 DB 타입은 TINYINT 이다.

이 필드 값이 1 이면 true 이고, 0 이면 false 이다.

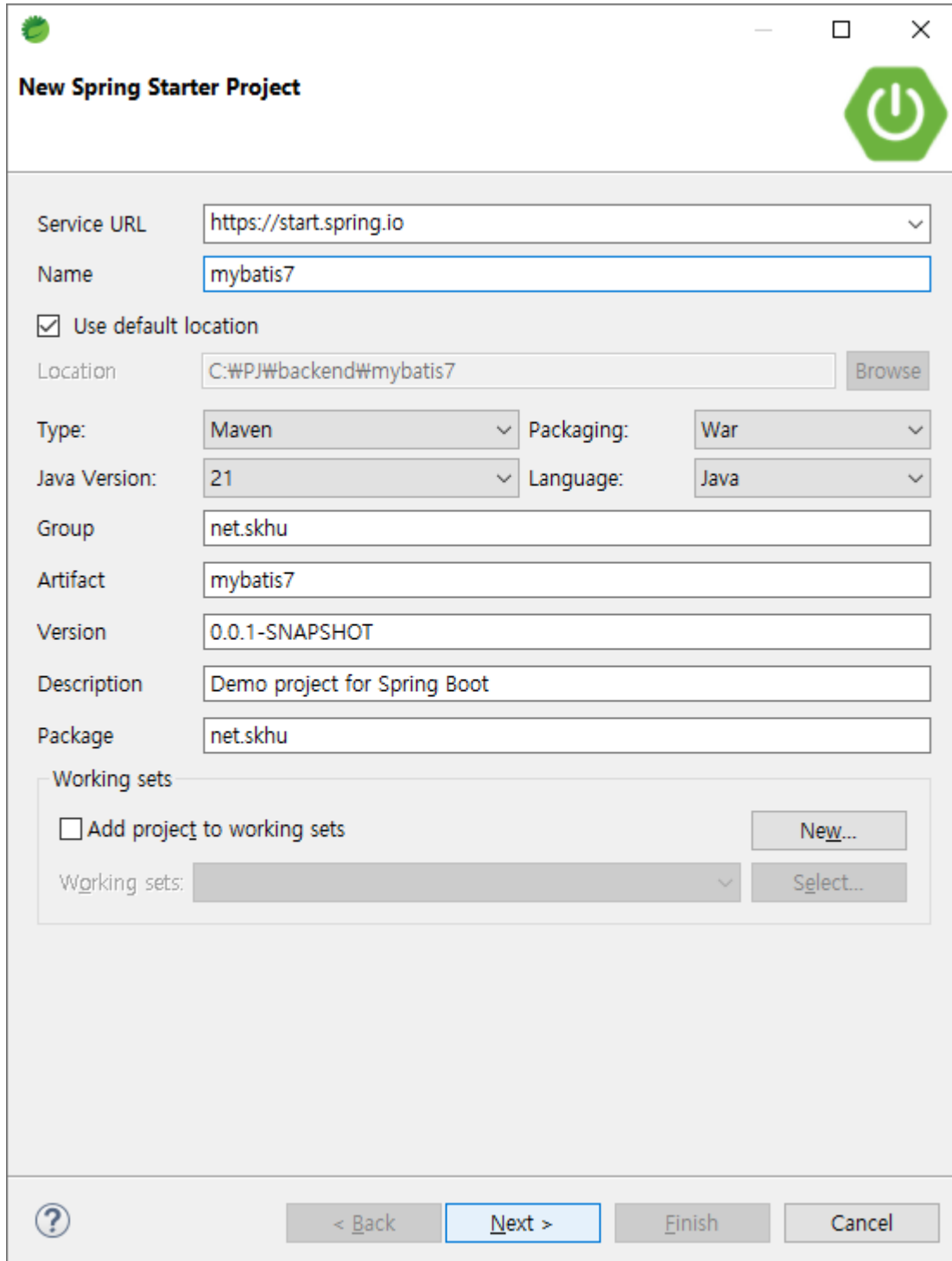
TINYINT 타입의 값을 Java에서 조회할 때, int 타입으로 가져와도 되고, boolean 타입으로 가져올 수도 있다.

2. mybatis7 프로젝트

1) 개요

전체 수강신청 목록을 조회하는 기능을 mybatis resultMap 기능을 사용하지 않고 구현하자.
평범한 DTO 클래스로 구현하자.

2) 프로젝트 생성



New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:



Package:

Working sets

☐ Add project to working sets

Working sets:

Name	mybatis7
Type	maven


New Spring Starter Project Dependencies


Spring Boot Version: 3.3.5

Frequently Used:

☒ Lombok
☒ MyBatis Framework
☒ MySQL Driver

☒ Spring Boot DevTools
☐ Spring Data JPA
☐ Spring Security

☒ Spring Web
☒ Thymeleaf
☐ Validation

Available:

Selected:


Type to search dependencies

- ▶ AI
- ▶ Developer Tools
- ▶ Google Cloud
- ▶ I/O
- ▶ Messaging
- ▶ Microsoft Azure
- ▶ NoSQL
- ▶ Observability
- ▶ Ops
- ▶ SQL
- ▶ Security
- ▶ Spring Cloud
- ▶ Spring Cloud Circuit Breaker
- ▶ Spring Cloud Config
- ▶ Spring Cloud Discovery

- X Spring Boot DevTools
- X Lombok
- X MyBatis Framework
- X MySQL Driver
- X Thymeleaf
- X Spring Web

Make Default

Clear Selection



< Back

Next >

Finish

Cancel

Spring BootDevTools
 Lombok
 MyBatis Framework
 MySQL Driver
 Thymeleaf
 Spring Web

3) application.properties

src/main/resources/application.properties

```
spring.application.name=mybatis7
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/student2?useUnicode=yes&characterEncoding=UTF-8&allowMultiQueries=true&serverTimezone=Asia/Seoul
spring.datasource.username=user1
spring.datasource.password=skhuA+4.5
server.port=8088
logging.level.net.skhu=DEBUG
```

4) Sugang DTO 클래스

src/main/java/net/skhu/dto/Sugang.java

```
1 package net.skhu.dto;
2
3 import lombok.Data;
4
5 @Data
6 public class Sugang {
7     int id;
8     int lectureId;
9     int studentId;
10    boolean repeated;
11    boolean cancel;
12    String grade;
13
14    String studentNo;
15    String name;
16
17    String title;
18    int year;
19    String semester;
20 }
```

sugang 테이블의 repeated 필드, cancel 필드의 DB 타입은 TINYINT 이지만 이 값을 Java에서 boolean 타입으로 조회할 수 있다.

어색함 점

sugang 테이블의 필드에 해당하는 속성은

id, lectureId, studentId, repeated, cancel, grade 6개이다.

studentNo, name, title, year, semester 5개 속성은

student 테이블이나 lecture 테이블의 필드에 해당한다.

이 DTO 클래스 이름이 Sugang 인 것이 맞나?

5) SugangMapper.java

src/main/java/net/skhu/mapper/SugangMapper.java

```
1 package net.skhu.mapper;
2
3 import java.util.List;
4
5 import org.apache.ibatis.annotations.Mapper;
6 import org.apache.ibatis.annotations.Select;
7
8 import net.skhu.dto.Sugang;
9
10 @Mapper
11 public interface SugangMapper {
12
13     @Select("""
14         SELECT g.*, s.studentNo, s.name, l.title, l.year, l.semester
15         FROM sugang g
16             JOIN student s ON g.studentId = s.id
17             JOIN lecture l ON g.lectureId = l.id
18         ORDER BY s.studentNo
19         """)
20     List<Sugang> findAll();
21 }
```


6) SugangController.java 구현

src/main/java/net/skhu/controller/SugangController.java

```
1 package net.skhu.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.stereotype.Controller;
5 import org.springframework.ui.Model;
6 import org.springframework.web.bind.annotation.GetMapping;
7 import net.skhu.mapper.SugangMapper;
8
9 @Controller
10 public class SugangController {
11
12     @Autowired SugangMapper sugangMapper;
13
14     @GetMapping("sugang/list")
15     public String list(Model model) {
16         model.addAttribute("sugangs", sugangMapper.findAll());
17         return "sugang/list";
18     }
19
20 }
```

7) common.css

src/main/resources/static/common.css

```
1  div.container { width: 800px; margin: 10px auto; font-size: 11pt; }
2  .shadow { box-shadow: 2px 2px 5px gray; }
3
4  .btn { padding: 0.4em 1em; border: 1px solid gray;
5         border-radius: 0.5em; background: linear-gradient(#fff, #ddd);
6         text-decoration: none; color: black;
7         display: inline-block; }
8  .btn:active {
9         -ms-transform: translateY(2px);
10        -webkit-transform: translateY(2px);
11        transform: translateY(2px);
12        background: #ccc; }
13
14  table.list { border-collapse: collapse; width: 100%; }
15  table.list td { padding: 4px; border: 1px solid gray; }
16  table.list th { padding: 4px; border: 1px solid gray; background-color: #eee; }
17
18  div.row { margin-bottom: 0.8em; }
19  div.row > label:nth-child(1) { display: inline-block; min-width: 4em; text-align: right; }
20
21  input[type=text], input[type=number] { padding: 0.3em; }
22  select { padding: 0.3em; }
23  h1 { margin: 0.5em 0; }
24  hr { margin: 1em 0; }
25  .error { color: red; padding: 0.2em; margin-left: 1em; }
26
27  div.pagination table { border-collapse: collapse; margin-top: 1em; }
28  div.pagination table td { border: 1px solid #ccc; text-align: center; }
29  div.pagination table td.active { background-color: #48f; }
30  div.pagination table a { text-decoration: none; color: blue; display: inline-block;
31                          width: 25px; padding: 6px; }
32  div.pagination table td.active a { color: white; }
```

8) sugnag/list.html

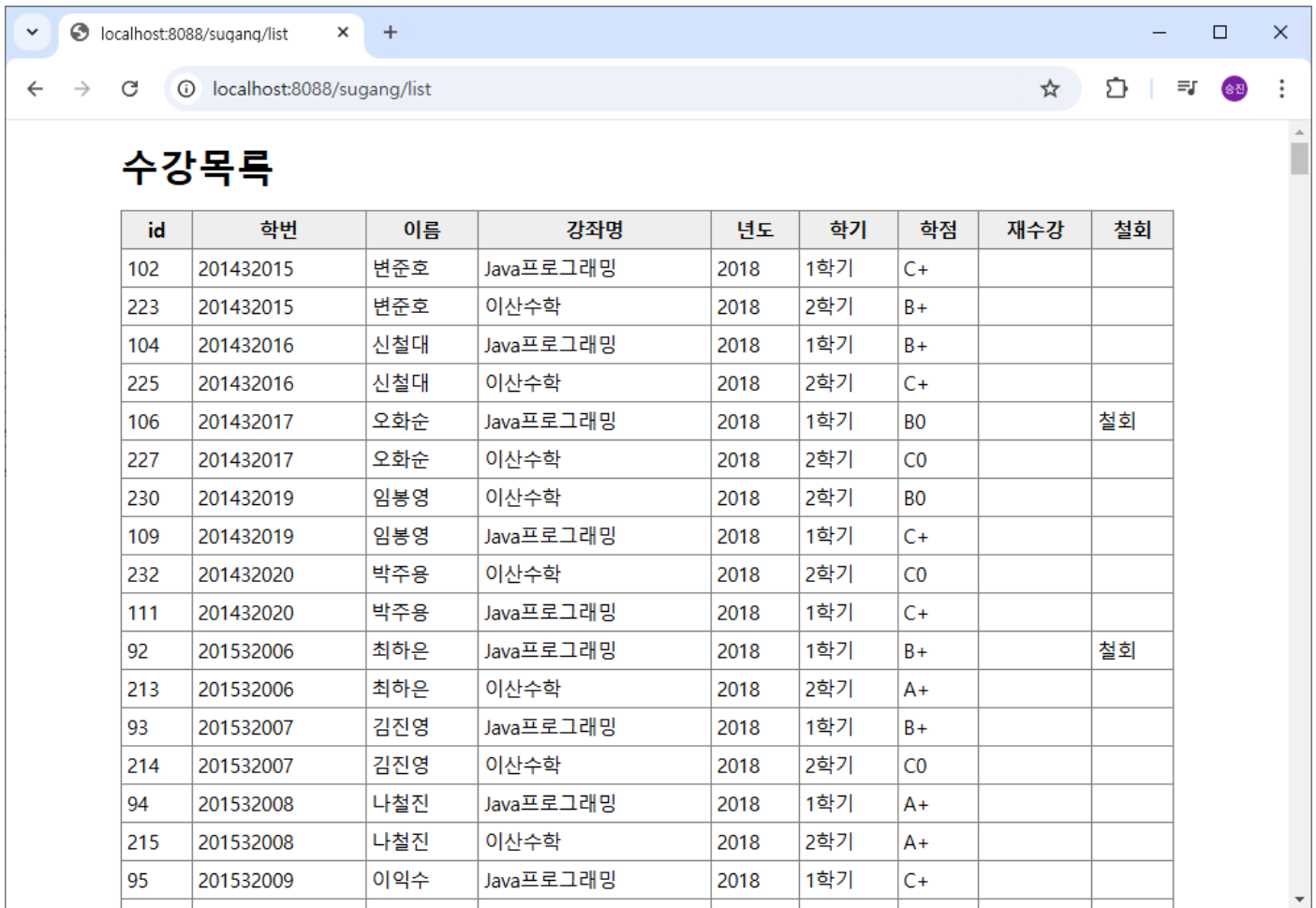
src/main/resources/templates/sugang/list.html

```
1 <!DOCTYPE html>
2 <html lang="ko" xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="utf-8">
5   <link rel="stylesheet" type="text/css" th:href="@{/common.css}" />
6 </head>
7 <body>
8 <div class="container">
9   <h1>수강목록</h1>
10  <table class="list">
11    <thead>
12      <tr>
13        <th>id</th>
14        <th>학번</th>
15        <th>이름</th>
16        <th>강좌명</th>
17        <th>년도</th>
18        <th>학기</th>
19        <th>학점</th>
20        <th>재수강</th>
21        <th>철회</th>
22      </tr>
23    </thead>
24    <tbody>
25      <tr th:each="s : ${sugangs}">
26        <td th:text="${s.id}"></td>
27        <td th:text="${s.studentNo}"></td>
28        <td th:text="${s.name}"></td>
29        <td th:text="${s.title}"></td>
30        <td th:text="${s.year}"></td>
31        <td th:text="${s.semester}"></td>
32        <td th:text="${s.grade}"></td>
33        <td th:text="${s.repeated ? '재수강' : ''}"></td>
34        <td th:text="${s.cancel ? '철회' : ''}"></td>
35      </tr>
36    </tbody>
37  </table>
38 </div>
39 </body>
40 </html>
```

`${s.repeated ? '재수강' : ''}`

s.repeated 값이 true 이면 '재수강' 문자열을 출력하고
false 이면 '' 빈 문자열을 출력한다.

9) 실행



The screenshot shows a web browser window with the address bar displaying 'localhost:8088/sugang/list'. The page title is '수강목록' (Lecture List). The table contains 20 rows of lecture data with columns for id, 학번 (Student ID), 이름 (Name), 강좌명 (Lecture Name), 년도 (Year), 학기 (Semester), 학점 (Credits), 재수강 (Retake), and 철회 (Withdrawal).

id	학번	이름	강좌명	년도	학기	학점	재수강	철회
102	201432015	변준호	Java프로그래밍	2018	1학기	C+		
223	201432015	변준호	이산수학	2018	2학기	B+		
104	201432016	신철대	Java프로그래밍	2018	1학기	B+		
225	201432016	신철대	이산수학	2018	2학기	C+		
106	201432017	오화순	Java프로그래밍	2018	1학기	B0		철회
227	201432017	오화순	이산수학	2018	2학기	C0		
230	201432019	임봉영	이산수학	2018	2학기	B0		
109	201432019	임봉영	Java프로그래밍	2018	1학기	C+		
232	201432020	박주용	이산수학	2018	2학기	C0		
111	201432020	박주용	Java프로그래밍	2018	1학기	C+		
92	201532006	최하은	Java프로그래밍	2018	1학기	B+		철회
213	201532006	최하은	이산수학	2018	2학기	A+		
93	201532007	김진영	Java프로그래밍	2018	1학기	B+		
214	201532007	김진영	이산수학	2018	2학기	C0		
94	201532008	나철진	Java프로그래밍	2018	1학기	A+		
215	201532008	나철진	이산수학	2018	2학기	A+		
95	201532009	이익수	Java프로그래밍	2018	1학기	C+		

10) 평가

이렇게 구현해도 큰 문제는 없다.
그렇지만...

sugang 테이블의 필드에 해당하는 속성은
id, lectureId, studentId, repeated, cancel, grade 6개이다.

studentNo, name, title, year, semester 5개 속성은
student 테이블이나 lecture 테이블의 필드에 해당한다.

이 DTO 클래스 이름이 Sugang 인 것이 맞나?

프로젝트를 개발하다보면, 이런 애마한 DTO 클래스를 매우 많이 만들게 된다. (몇십개)
Sugang DTO 클래스에 학번, 이름 등의 속성이 들어있고
Student DTO 클래스에도 학점, 수강 과목 등의 속성이 들어있게 된다.
그러다 보면, 이 두 클래스의 차이가 뭔가 헷갈리게 되고, 뭐가 어색하고 불편해진다.

3. mybatis8 프로젝트

1) mybatis resultMap 기능

mybatis는 ORM 기술 중의 하나이다.

ORM 이란 객체지향 언어의 객체와 관계형 데이터를 서로 변환해 준다는 것이다.

DB 조회 결과를, DTO 객체 하나에 넣는 것이 아니고, 여러 객체에 나누어 넣을 때, mybatis의 resultMap 기능을 사용한다.

2) mybatis mapper 구현 방법

annotation으로 구현하기

지난 시간에 구현한 StudentMapper는 annotation을 사용하여 mapper를 구현하였다.
간단한 SQL을 구현할 때, annotation을 이용하여 구현하는 것이 편하다.

mapper XML 파일에 구현하기

mybatis resultMap 기능을 구현하려면 mapper XML 파일에 구현하는 것이 편하다.
또는 mapper에 여러 설정을 할 때도 XML 파일이 편하다.

위 두 방법을 혼합해도 된다.

간단한 SQL의 메소드는 annotation으로 구현하고

복잡한 SQL의 메소드는 mapper XML 파일에 구현할 수 있다.

3) FQN (Full Qualified Name)

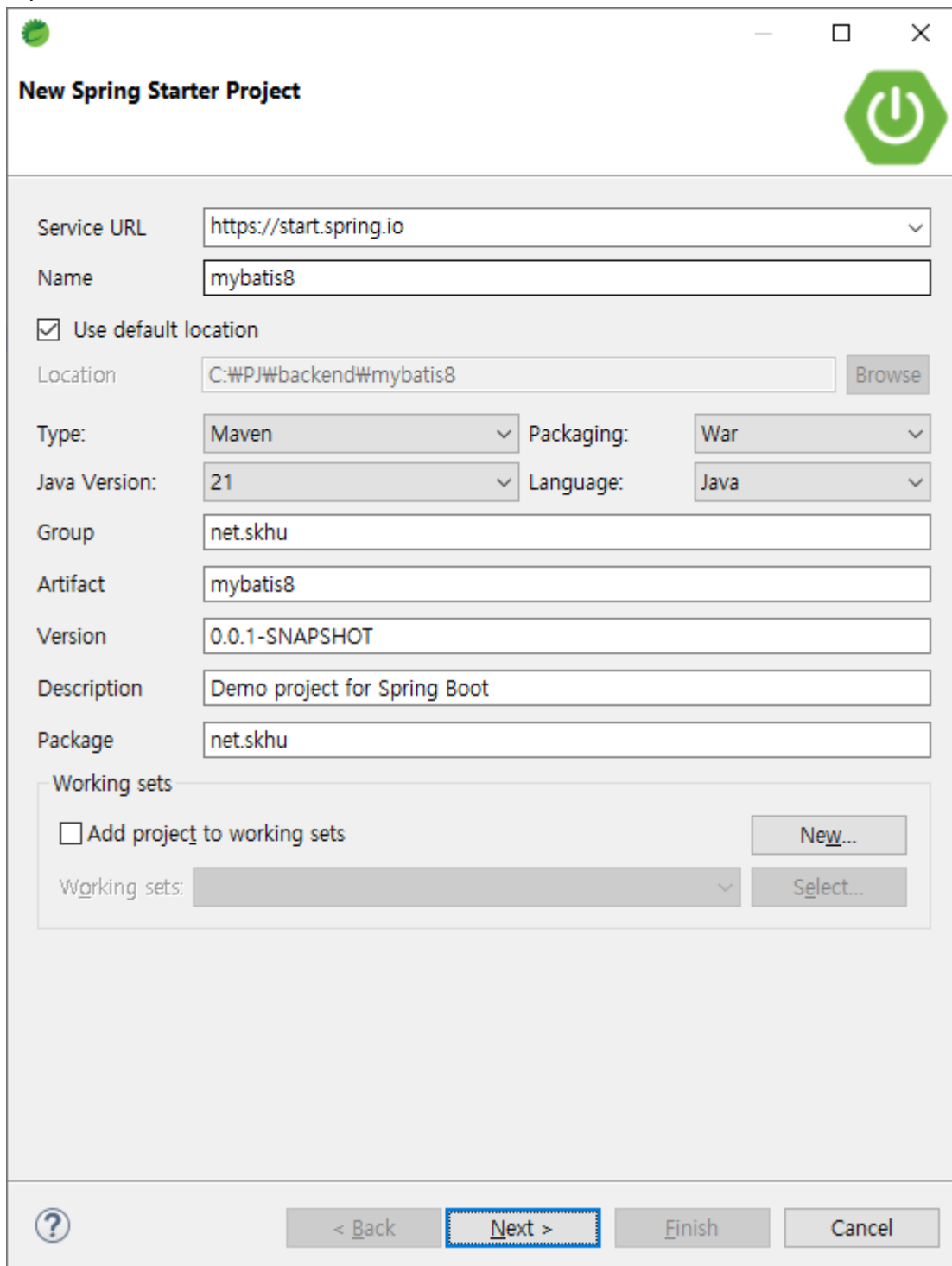
자바 클래스 이름에 소속 패키지까지 붙인 것이 FQN 이다.

예) StudentController 자바 클래스

이름: StudentController

FQN: net.skhu.controller.StudentController

4) 프로젝트 생성



The image shows a 'New Spring Starter Project' dialog box. It contains several input fields and checkboxes for configuring a new project. The 'Name' field is set to 'mybatis8'. The 'Type' is set to 'Maven' and 'Packaging' is set to 'War'. The 'Java Version' is set to '21' and 'Language' is set to 'Java'. The 'Group' is set to 'net.skhu'. The 'Artifact' is set to 'mybatis8'. The 'Version' is set to '0.0.1-SNAPSHOT'. The 'Description' is set to 'Demo project for Spring Boot'. The 'Package' is set to 'net.skhu'. The 'Working sets' section has an unchecked checkbox 'Add project to working sets' and a 'New...' button. The 'Working sets' list is empty, and there is a 'Select...' button. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted with a blue border.

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:



Package:

Working sets

☐ Add project to working sets

Working sets:

Name	mybatis8
Type	maven

New Spring Starter Project Dependencies

Spring Boot Version: 3.3.5

Frequently Used:

☒ Lombok

☒ MyBatis Framework

☒ MySQL Driver

☒ Spring Boot DevTools

☐ Spring Data JPA

☐ Spring Security

☒ Spring Web

☒ Thymeleaf

☐ Validation

Available:

Type to search dependencies

▶ AI

▶ Developer Tools

▶ Google Cloud

▶ I/O

▶ Messaging

▶ Microsoft Azure

▶ NoSQL

▶ Observability

▶ Ops

▶ SQL

▶ Security

▶ Spring Cloud

▶ Spring Cloud Circuit Breaker

▶ Spring Cloud Config

▶ Spring Cloud Discovery

Selected:

X Spring Boot DevTools

X Lombok

X MyBatis Framework


X MySQL Driver

X Thymeleaf

X Spring Web

Make Default

Clear Selection



< Back

Next >

Finish

Cancel

Spring BootDevTools
Lombok
MyBatis Framework
MySQL Driver
Thymeleaf
Spring Web

5) application.properties

src/main/resources/application.properties

```
spring.application.name=mybatis8
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/student2?useUnicode=yes&characterEncoding=UTF-8&allowMultiQueries=true&serverTimezone=Asia/Seoul
spring.datasource.username=user1
spring.datasource.password=skhuA+4.5
server.port=8088
logging.level.net.skhu=DEBUG
```


6) Student DTO 클래스

src/main/java/net/skhu/dto/Student.java

```
1 package net.skhu.dto;
2
3 import lombok.Data;
4
5 @Data
6 public class Student {
7     int id;
8     String studentNo;
9     String name;
10    int departmentId;
11    String phone;
12    String sex;
13    String email;
14 }
```

student 테이블의 필드에 해당하는 속성만 구현된 단순한 DTO 클래스이다.

7) Lecture DTO 클래스

src/main/java/net/skhu/dto/Lecture.java

```
1 package net.skhu.dto;
2
3 import lombok.Data;
4
5 @Data
6 public class Lecture {
7     int id;
8     String title;
9     int profesorId;
10    int year;
11    String semester;
12    String room;
13 }
```

lecture 테이블의 필드에 해당하는 속성만 구현된 단순한 DTO 클래스이다.

8) Sugang DTO 클래스

src/main/java/net/skhu/dto/Sugang.java

```
1 package net.skhu.dto;
2
3 import lombok.Data;
4
5 @Data
6 public class Sugang {
7     int id;
8     int lectureId;
9     int studentId;
10    boolean repeated;
11    boolean cancel;
12    String grade;
13
14    Lecture lecture;
15    Student student;
16 }
```

sugang 테이블을 조회할 때,
강좌 정보, 학생 정보도 같이 조회해야 하기 때문에
Sugang DTO 클래스에 Lecture, Student 객체 속성을 추가하였다.

9) SugangMapper.java

src/main/java/net/skhu/mapper/SugangMapper.java

```
1 package net.skhu.mapper;
2
3 import java.util.List;
4
5 import org.apache.ibatis.annotations.Mapper;
6 import org.apache.ibatis.annotations.ResultMap;
7 import org.apache.ibatis.annotations.Select;
8
9 import net.skhu.dto.Sugang;
10
11 @Mapper
12 public interface SugangMapper {
13
14     @ResultMap("findAll")
15     @Select("""
16         SELECT g.*, s.studentNo, s.name, l.title, l.year, l.semester
17         FROM sugang g
18         JOIN student s ON g.studentId = s.id
19         JOIN lecture l ON g.lectureId = l.id
20         ORDER BY s.studentNo
21     """)
22     List<Sugang> findAll();
23 }
```

@ResultMap("findAll")

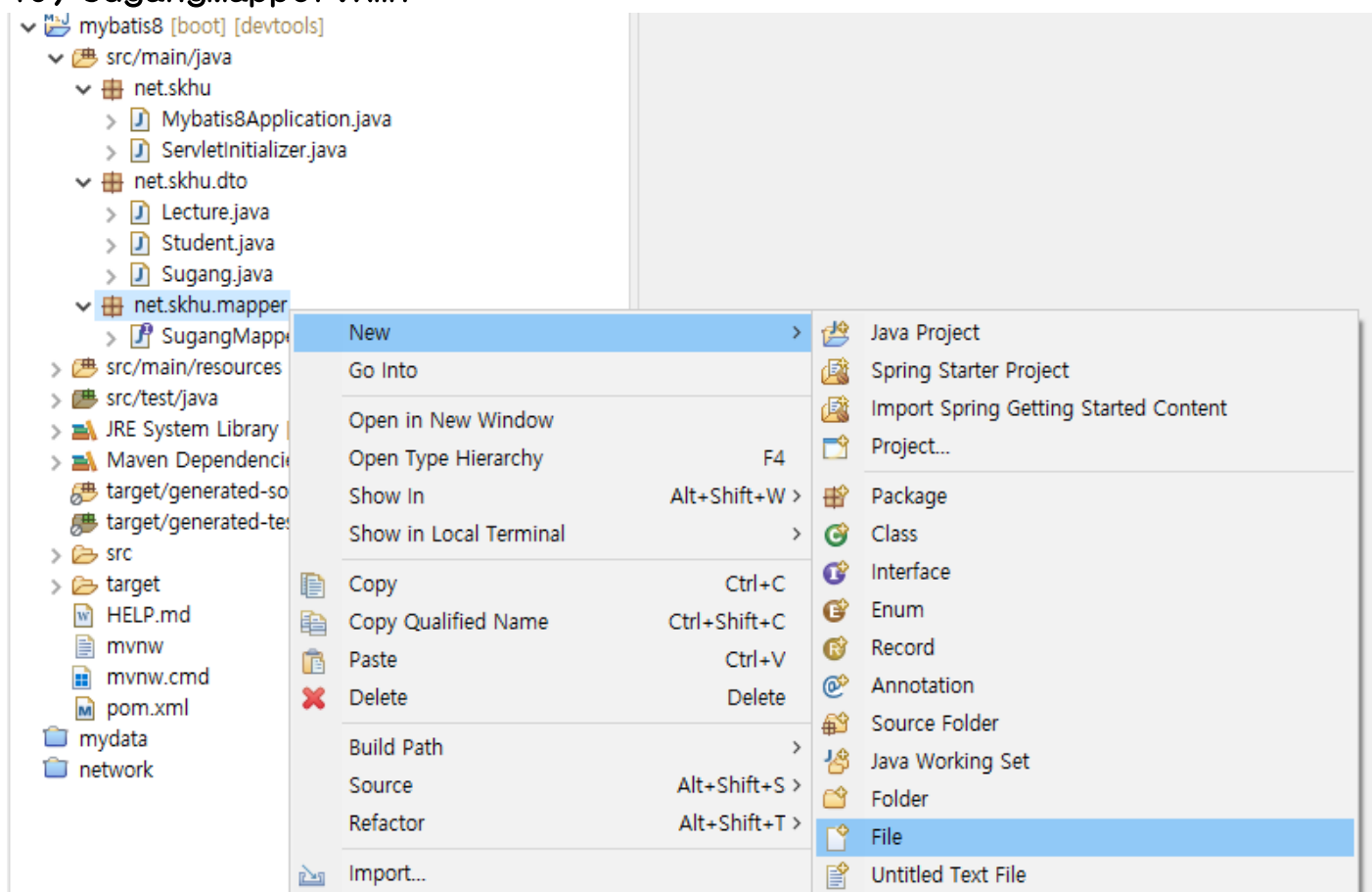
SugangMapper.xml 파일의 id="findAll" resultMap 방법으로
조회 결과를 DTO 객체에 채워서 리턴하라는 설정이다.

조회 결과 컬럼을 다음과 같이 여러 객체로 쪼개서 채워야 한다.

조회 결과 컬럼	Sugang DTO 객체의 속성
id	id
lectureId	lectureId
studentId	studentId
repeated	repeated
cancel	cancel
grade	grade
studentId	student.id
studentNo	student.studentNo
name	student.name
lectureId	lecture.id
title	lecture.title
year	lecture.year
semester	lecture.semester

위와 같이 쪼개 넣는 방법을 SugangMapper.xml 파일에 정의해야한다.

10) SugangMapper.xml



SugangMapper.xml

src/main/java/net/skhu/mapper/SugangMapper.xml

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
3     "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4
5 <mapper namespace="net.skhu.mapper.SugangMapper">
6
7     <resultMap id="findAll" type="net.skhu.dto.Sugang">
8         <id property="id" column="id" />
9         <result property="lectureId" column="lectureId" />
10        <result property="studentId" column="studentId" />
11        <result property="repeated" column="repeated" />
12        <result property="cancel" column="cancel" />
13        <result property="grade" column="grade" />
14        <association property="lecture" javaType="net.skhu.dto.Lecture">
15            <id property="id" column="studentId" />
16            <result property="title" column="title" />
17            <result property="year" column="year" />
18            <result property="semester" column="semester" />
19        </association>
20        <association property="student" javaType="net.skhu.dto.Student">
21            <id property="id" column="lectureId" />
22            <result property="studentNo" column="studentNo" />
23            <result property="name" column="name" />
24        </association>
25    </resultMap>
26
27 </mapper>
```

```
<mapper namespace="net.skhu.mapper.SugangMapper">
```

SugangMapper 자바 인터페이스 FQN을 정확하게 적어야 한다.

```
<resultMap id="findAll" type="net.skhu.dto.Sugang">
```

SugangMapper.java 파일의 @ResultMap("findAll") 부분과 일치해야 한다.

type="net.skhu.dto.Sugang" 메소드가 리턴하는 DTO 클래스의 FQN을 정확하게 적어야 한다.

```
<result property="속성명" column="컬럼명" />
```

```
<id property="속성명" column="컬럼명" />
```

조회 결과 컬럼과, 그 값을 채울 DTO 객체 속성명을 적는다.

평범한 컬럼인 경우 <result> 태그를 사용하고

기본키(primary key) 컬럼인 경우, <id> 태그를 사용해야 한다.

```
<association property="lecture" javaType="net.skhu.dto.Lecture">
```

```
<id property="id" column="studentId" />
```

```
<result property="title" column="title" />
```

```
<result property="year" column="year" />
```

```
<result property="semester" column="semester" />
```

```
</association>
```

Sugang DTO 클래스의 lecture 객체 속성에 채울 정보

SugangMapper.xml 수정

src/main/java/net/skhu/mapper/SugangMapper.xml

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
3      "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4  <mapper namespace="net.skhu.mapper.SugangMapper">
5
6      <resultMap id="findAll" type="net.skhu.dto.Sugang" autoMapping="true">
7          <id property="id" column="id" />
8          <association property="lecture" javaType="net.skhu.dto.Lecture" autoMapping="true">
9              <id property="id" column="studentId" />
10         </association>
11         <association property="student" javaType="net.skhu.dto.Student" autoMapping="true">
12             <id property="id" column="lectureId" />
13         </association>
14     </resultMap>
15
16 </mapper>
```

일단 실행해 본 후 수정하자.

`autoMapping="true"` 을 사용하면,

조회 결과 컬럼과, 그 값을 채울 DTO 객체 속성명이 같은 `<result>` 태그 항목을들 생략할 수 있다.
기본키(primary key) `<id>` 태그는 생략할 수 없다.

11) SugangController.java

src/main/java/net/skhu/controller/SugangController.java

```
1  package net.skhu;
2
3  import org.springframework.beans.factory.annotation.Autowired;
4  import org.springframework.stereotype.Controller;
5  import org.springframework.ui.Model;
6  import org.springframework.web.bind.annotation.GetMapping;
7  import net.skhu.mapper.SugangMapper;
8
9  @Controller
10 public class SugangController {
11
12     @Autowired SugangMapper sugangMapper;
13
14     @GetMapping("sugang/list")
15     public String list(Model model) {
16         model.addAttribute("sugangs", sugangMapper.findAll());
17         return "sugang/list";
18     }
19
20 }
```

12) common.css

src/main/resources/static/common.css

```
1  div.container { width: 800px; margin: 10px auto; font-size: 11pt; }
2  .shadow { box-shadow: 2px 2px 5px gray; }
3
4  .btn { padding: 0.4em 1em; border: 1px solid gray;
5         border-radius: 0.5em; background: linear-gradient(#fff, #ddd);
6         text-decoration: none; color: black;
7         display: inline-block; }
8  .btn:active {
9         -ms-transform: translateY(2px);
10        -webkit-transform: translateY(2px);
11        transform: translateY(2px);
12        background: #ccc; }
13
14  table.list { border-collapse: collapse; width: 100%; }
15  table.list td { padding: 4px; border: 1px solid gray; }
16  table.list th { padding: 4px; border: 1px solid gray; background-color: #eee; }
17
18  div.row { margin-bottom: 0.8em; }
19  div.row > label:nth-child(1) { display: inline-block; min-width: 4em; text-align: right; }
20
21  input[type=text], input[type=number] { padding: 0.3em; }
22  select { padding: 0.3em; }
23  h1 { margin: 0.5em 0; }
24  hr { margin: 1em 0; }
25  .error { color: red; padding: 0.2em; margin-left: 1em; }
26
27  div.pagination table { border-collapse: collapse; margin-top: 1em; }
28  div.pagination table td { border: 1px solid #ccc; text-align: center; }
29  div.pagination table td.active { background-color: #48f; }
30  div.pagination table a { text-decoration: none; color: blue; display: inline-block;
31                          width: 25px; padding: 6px; }
32  div.pagination table td.active a { color: white; }
```


13) sugnag/list.html

src/main/resources/templates/sugang/list.html

```
1 <!DOCTYPE html>
2 <html lang="ko" xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="utf-8">
5   <link rel="stylesheet" type="text/css" th:href="@{/common.css}" />
6 </head>
7 <body>
8 <div class="container">
9   <h1>수강목록</h1>
10  <table class="list">
11    <thead>
12      <tr>
13        <th>id</th>
14        <th>학번</th>
15        <th>이름</th>
16        <th>강좌명</th>
17        <th>년도</th>
18        <th>학기</th>
19        <th>학점</th>
20        <th>재수강</th>
21        <th>철회</th>
22      </tr>
23    </thead>
24    <tbody>
25      <tr th:each="s : ${sugangs}">
26        <td th:text="${s.id}"></td>
27        <td th:text="${s.student.studentNo}"></td>
28        <td th:text="${s.student.name}"></td>
29        <td th:text="${s.lecture.title}"></td>
30        <td th:text="${s.lecture.year}"></td>
31        <td th:text="${s.lecture.semester}"></td>
32        <td th:text="${s.grade}"></td>
33        <td th:text="${s.repeated ? '재수강' : ''}"></td>
34        <td th:text="${s.cancel ? '철회' : ''}"></td>
35      </tr>
36    </tbody>
37  </table>
38 </div>
39 </body>
40 </html>
```

4. 연습문제

@Data

```
public class Student {  
    int id;  
    String studentNo;  
    String name;  
    int departmentId;  
    String phone;  
    String sex;  
    String email;  
  
    Department department;  
}
```

student 테이블을 조회할 때, department 소속 학과 정보도 같이 조회하자.
mybatis resultMap 기능으로 구현하자.

1) 답

Department.java

```
package net.skhu.dto;

import lombok.Data;

@Data
public class Department {
    int id;
    String name;
    String shortName;
    String phone;
}
```

Student.java

```
package net.skhu.dto;

import lombok.Data;

@Data
public class Student {
    int id;
    String studentNo;
    String name;
    int departmentId;
    String phone;
    String sex;
    String email;

    Department department;
}
```

StudentMapper.java

```
package net.skhu.mapper;

import java.util.List;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.ResultMap;
import org.apache.ibatis.annotations.Select;
import net.skhu.dto.Student;

@Mapper
public interface StudentMapper {

    @ResultMap("findAll")
    @Select("""
        SELECT s.*,
               d.name departmentName, d.shortName, d.phone departmentPhone
        FROM student s
        JOIN department d ON s.departmentId = d.id """)
    List<Student> findAll();
}
```

student 테이블에서 name, phone 필드가 있으므로,

조회 결과 컬럼명의 중복을 막기 위해서

조회 결과 학과명 컬럼명을 departmentName 으로 수정하고

학과전화번호 컬럼명을 departmentPhone 으로 수정한다.

id	studentNo	name	departmentId	phone	sex	email	departmentName	shortName	departmentPhone
1	201532006	최하은	1	010-4361-1145	남	choi064@skhu.ac.kr	소프트웨어공학과	소프	02-2610-9563
5	201532010	이제문1111	1	010-7700-1333	남	lee107@skhu.ac.kr	소프트웨어공학과	소프	02-2610-9563
6	201532011	김영우	1	010-2090-1421	남	kim112@skhu.ac.kr	소프트웨어공학과	소프	02-2610-9563
10	201532015	박원종	1	010-7655-1724	남	park157@skhu.ac.kr	소프트웨어공학과	소프	02-2610-9563
11	201432015	변준호	1	010-2245-1750	남	b@skhu.net	소프트웨어공학과	소프	02-2610-9563
16	201532018	박재인	1	010-2616-2144	남	park182@skhu.ac.kr	소프트웨어공학과	소프	02-2610-9563
23	201532023	성희지	1	010-4267-2555	남	seong234@skhu.ac.kr	소프트웨어공학과	소프	02-2610-9563

StudentMapper.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="net.skhu.mapper.StudentMapper">

    <resultMap id="findAll" type="net.skhu.dto.Student" autoMapping="true">
        <id property="id" column="id" />
        <association property="department" javaType="net.skhu.dto.Department" autoMapping="true">
            <id property="id" column="departmentId" />
            <result property="name" column="departmentName" />
            <result property="phone" column="departmentPhone" />
        </association>
    </resultMap>

</mapper>
```

학과명 조회 결과 컬럼명이 departmentName 이고,

학과전화 번호 조회 결과 컬럼명이 departmentPhone 이다.

이 컬럼의 값을 각각 Department 객체의 name, phone 속성에 채워야 한다.

StudentController.java

```
package net.skhu;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import net.skhu.mapper.StudentMapper;

@Controller
public class StudentController {

    @Autowired StudentMapper studentMapper;

    @GetMapping("student/list")
    public String list(Model model) {
        model.addAttribute("students", studentMapper.findAll());
        return "student/list";
    }
}
```

student/list.html

```
<!DOCTYPE html>
<html lang="ko" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" th:href="@{/common.css}" />
</head>
<body>
<div class="container">
    <h1>학생목록</h1>
    <table class="list">
        <thead>
            <tr>
                <th>id</th>
                <th>학번</th>
                <th>이름</th>
                <th>강좌명</th>
                <th>단축명</th>
            </tr>
        </thead>
        <tbody>
            <tr th:each="s : ${students}">
                <td th:text="${s.id}"></td>
                <td th:text="${s.studentNo}"></td>
                <td th:text="${s.name}"></td>
                <td th:text="${s.department.name}"></td>
                <td th:text="${s.department.shortName}"></td>
            </tr>
        </tbody>
    </table>
</div>
</body>
</html>
```

localhost:8088/student/list

localhost:8088/student/list

학생목록

id	학번	이름	강좌명	단축명
1	201532006	최하은	소프트웨어공학과	소프
5	201532010	이제문1111	소프트웨어공학과	소프
6	201532011	김영우	소프트웨어공학과	소프
10	201532015	박원종	소프트웨어공학과	소프
11	201432015	변준호	소프트웨어공학과	소프
16	201532018	박재인	소프트웨어공학과	소프
23	201532023	성희지	소프트웨어공학과	소프
30	201532030	김만기	소프트웨어공학과	소프