

```

1  HOL : Spring and MyBatis
2  -----
3  Task1. MySQL의 World Database의 City Table 가져오기
4  1. MariaDB 설치하기
5      1)https://mariadb.org
6      2)https://mariadb.org/download/ MariaDB Server 10.6.4
7
8  2. MyBatisDemo project 생성
9      1)In Package Explorer > right-click > New > Java Project
10     2)Project name : MybatisDemo
11
12 3. src > right-click > New > Package
13     1)Package name : com.example
14
15 4. Java Project를 Spring Project로 변환
16     1)MybatisDemo Project > right-click > Configure > Convert to Maven Project
17     2)Project : /MybatisDemo
18     3)Group Id : MybatisDemo
19     4)Artifact Id : MybatisDemo
20     5)version : 0.0.1-SNAPSHOT
21     6)Packaging : jar
22     7)Finish
23
24     8)MybatisDemo Project > right-click > Spring > Add Spring Project Nature
25
26     9)pom.xml 파일에 Spring Context Dependency 추가하기
27         <version>0.0.1-SNAPSHOT</version>
28         <dependencies>
29             <dependency>
30                 <groupId>org.springframework</groupId>
31                 <artifactId>spring-context</artifactId>
32                 <version>5.3.100</version>
33             </dependency>
34         </dependencies>
35
36     10)pom.xml > right-click > Run As > Maven install
37         [INFO] BUILD SUCCESS 확인
38
39
40 5. MyBatis library 검색 및 설치
41     1)Maven Repository에서 'mybatis'로 검색
42
43         <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->
44         <dependency>
45             <groupId>org.mybatis</groupId>
46             <artifactId>mybatis</artifactId>
47             <version>3.5.7</version>
48         </dependency>
49
50     2)pom.xml에 등록 및 설치
51
52
53 6. MyBatis-Spring library 검색 및 설치
54     1)Maven Repository에서 'mybatis spring'으로 검색
55
56         <!-- https://mvnrepository.com/artifact/org.mybatis-spring -->
57         <dependency>
58             <groupId>org.mybatis</groupId>
59             <artifactId>mybatis-spring</artifactId>
60             <version>2.0.6</version>
61         </dependency>
62
63     2)pom.xml에 등록 및 설치
64
65
66 7. Jdbc Driver 설치
67     1)Oracle 12C 이후 version일 경우, mvnrepository에서 oralc로 검색후, Ojdbc8 설치
68         <dependency>
69             <groupId>com.oracle.database.jdbc</groupId>
70             <artifactId>ojdbc8</artifactId>
71             <version>21.3.0.0</version>
72         </dependency>
73
74     2)Oracle 11g version일 경우

```

```

75     -pom.xml에 붙여 넣고 Maven Install 하기
76     <dependency>
77         <groupId>com.oracle</groupId>
78         <artifactId>ojdbc6</artifactId>
79         <version>11.2</version>
80     </dependency>
81
82 3)MySQL의 경우, MySQL Connector/J로 들어가서
83     <dependency>
84         <groupId>mysql</groupId>
85         <artifactId>mysql-connector-java</artifactId>
86         <version>8.0.26</version>
87     </dependency>
88
89 4)MariaDB의 경우, MariaDB Java Client로 들어가서
90     <dependency>
91         <groupId>org.mariadb.jdbc</groupId>
92         <artifactId>mariadb-java-client</artifactId>
93         <version>2.7.4</version>
94     </dependency>
95
96 5)pom.xml > right-click > Run As > Maven install
97     [INFO] BUILD SUCCESS 확인
98
99
100 8. Spring JDBC 설치
101 1)JdbcTemplate를 사용하기 위해 pom.xml에 다음 dependency를 추가해야 함.
102
103     <dependency>
104         <groupId>org.springframework</groupId>
105         <artifactId>spring-jdbc</artifactId>
106         <version>5.3.100</version>
107     </dependency>
108
109 2)pom.xml에 붙여 넣고 Maven Install 하기
110
111
112 9. JUnit5 Library 설치
113 1)http://mvnrepository.com에 접근
114 2)junit으로 검색
115 3)JUnit Jupiter API click
116 4)JUnit Jupiter API 5.8.1 version을 pom.xml에 추가
117
118     <dependency>
119         <groupId>org.junit.jupiter</groupId>
120         <artifactId>junit-jupiter-api</artifactId>
121         <version>5.8.1</version>
122         <scope>test</scope>
123     </dependency>
124
125 5)pom.xml > right-click > Run As > Maven Install
126
127
128 10. Lombok library 추가
129 1)https://mvnrepository.com/에서 'lombok'으로 검색
130 2)'Project Lombok' click
131 3)1.18.20 click
132 4)dependency copy해서 pom.xml에 붙여넣기
133
134     <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
135     <dependency>
136         <groupId>org.projectlombok</groupId>
137         <artifactId>lombok</artifactId>
138         <version>1.18.20</version>
139         <scope>provided</scope>
140     </dependency>
141
142 5)pom.xml > right-click > Run As > Maven install
143     [INFO] BUILD SUCCESS 확인
144
145
146 11. MybatisDemo/resources folder 생성
147 1)MybatisDemo project > right-click > Build Path > Configure Build Path
148 2)Source Tab > Add Folder

```

```
149 3)MybatisDemo click
150 4)Create New Folder > Folder name : resources > Finish > OK
151 5)MybatisDemo/resources(new) 확인
152 6)Apply and Close
153
154
155 12. resources/dbinfo.properties file 생성
156 1)resources > right-click > New > File
157 2)File name : dbinfo.properties > Finish
158 3)Oracle Database 일 경우
159 db.driverClass=oracle.jdbc.driver.OracleDriver
160 db.url=jdbc:oracle:thin:@localhost:1521:XE
161 db.username=hr
162 db.password=hr
163
164 4)MySQL Database 인 경우
165 db.driverClass=com.mysql.jdbc.Driver
166 db.url=jdbc:mysql://localhost:3306/world
167 db.username=root
168 db.password=javamysql
169
170 5)MariaDB인 경우
171 db.driverClass=org.mariadb.jdbc.Driver
172 db.url=jdbc:mariadb://localhost:3306/world
173 db.username=root
174 db.password=javamariadb
175
176
177 13. world database downloads
178 1)https://dev.mysql.com/doc/index-other.html
179 2)Example Databases > world database > Zip
180 3)Unzip
181 4)MariaDB login 후 world database 실행
182
183
184 14. 여러 Package 생성
185 1)/src/com.example.vo
186 2)/src/com.example.service
187 3)/src/com.example.dao
188
189
190 15. VO class 작성
191 1)/src/com.example.vo.CityVO.java 생성
192
193 package com.example.vo;
194
195 import lombok.Getter;
196 import lombok.Setter;
197 import lombok.NoArgsConstructor;
198 import lombok.ToString;
199
200 @Getter
201 @Setter
202 @ToString
203 @NoArgsConstructor
204 public class CityVO {
205     private int id;
206     private String name;
207     private String countryCode;
208     private String district;
209     private int population;
210 }
211
212
213 16. Mapping file 작성 및 MyBatis 설정
214 1)/resources/SqlMapConfig.xml
215
216 <?xml version="1.0" encoding="UTF-8" ?>
217 <!DOCTYPE configuration
218 PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
219 "http://mybatis.org/dtd/mybatis-3-config.dtd">
220 <configuration>
221 <typeAliases>
222 <typeAlias type="com.example.vo.CityVO" alias="cityVO" />
```

```

223         </typeAliases>
224
225     </configuration>
226
227 2)/resources/mybatis-mapper.xml
228
229     <?xml version="1.0" encoding="UTF-8"?>
230     <!DOCTYPE mapper
231         PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
232         "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
233     <mapper namespace="City">
234
235     </mapper>
236
237
238 17. Dao 객체 생성
239 1)mybatis-mapper.xml
240
241     <resultMap id="cityResult" type="cityVO">
242         <result property="id" column="ID" />
243         <result property="name" column="Name" />
244         <result property="district" column="District" />
245         <result property="countryCode" column="CountryCode" />
246         <result property="population" column="Population" />
247     </resultMap>
248
249     <select id="selectCityByName" resultType="cityVO" resultMap="cityResult">
250         SELECT * FROM world.city WHERE name = #{name}
251     </select>
252
253 2)CityDao interface 생성
254 -/src/com.example.dao.CityDao.java
255
256     package com.example.dao;
257
258     import java.util.List;
259     import com.example.vo.CityVO;
260
261     public interface CityDao {
262         List<CityVO> readAll();
263         CityVO read(String name);
264     }
265
266 3)CityDaoImpl class 생성
267 -/src/com/src/com.example.dao.CityDaoImpl.java
268
269     package com.example.dao;
270
271     import java.util.List;
272
273     import org.apache.ibatis.session.SqlSession;
274     import org.springframework.beans.factory.annotation.Autowired;
275     import org.springframework.stereotype.Repository;
276
277     import com.example.vo.CityVO;
278     @Repository("cityDao")
279     public class CityDaoImpl implements CityDao {
280
281         @Autowired
282         private SqlSession session;
283
284         @Override
285         public List<CityVO> readAll() {
286             return null;
287         }
288
289         @Override
290         public CityVO read(String name) {
291             CityVO city = session.selectOne("City.selectCityByName", name);
292             return city;
293         }
294     }
295
296

```

```

297 18. Service 객체 생성
298 1)CityService interface 생성
299    -/src/com.example.service.CityService.java
300
301    package com.example.service;
302
303    import java.util.List;
304    import com.example.vo.CityVO;
305
306    public interface CityService {
307        List<CityVO> getCityList();
308        CityVO getCity(String name);
309    }
310
311 2)CityServiceImpl class 생성
312    -/src/com.example.service.CityServiceImpl.java
313    package com.example.service;
314
315    import java.util.List;
316
317    import org.springframework.beans.factory.annotation.Autowired;
318    import org.springframework.stereotype.Service;
319
320    import com.example.dao.CityDao;
321    import com.example.vo.CityVO;
322
323    @Service("cityService")
324    public class CityServiceImpl implements CityService {
325
326        @Autowired
327        private CityDao cityDao;
328
329        @Override
330        public List<CityVO> getCityList() {
331            return null;
332        }
333
334        @Override
335        public CityVO getCity(String name) {
336            return this.cityDao.read(name);
337        }
338    }
339
340

```

```

341 19. Bean Configuration XML 작성
342 1)/resources > right-click > New > Spring Bean Configuration File
343 2)File name : beans.xml > Finish
344 3)Namespace Tab
345 4)Check context - http://www.springframework.org/schema/context
346

```

```

347 <?xml version="1.0" encoding="UTF-8"?>
348 <beans xmlns="http://www.springframework.org/schema/beans"
349     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
350     xmlns:context="http://www.springframework.org/schema/context"
351     xsi:schemaLocation="http://www.springframework.org/schema/beans
352         http://www.springframework.org/schema/beans/spring-beans.xsd
353         http://www.springframework.org/schema/context
354         http://www.springframework.org/schema/context/spring-context-3.2.xsd">
355
356     <!-- mybatis-spring 설정 -->
357     <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
358         <property name="dataSource" ref="dataSource" />
359         <property name="configLocation" value="classpath:SqlMapConfig.xml" />
360         <property name="mapperLocations">
361             <list>
362                 <value>classpath:mybatis-mapper.xml</value>
363             </list>
364         </property>
365     </bean>
366
367     <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate">
368         <constructor-arg ref="sqlSessionFactory" />
369     </bean>

```

```

369     <context:property-placeholder location="classpath:dbinfo.properties" />
370     <bean id="dataSource" class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
371         <property name="driverClass" value="${db.driverClass}" />
372         <property name="url" value="${db.url}" />
373         <property name="username" value="${db.username}" />
374         <property name="password" value="${db.password}" />
375     </bean>
376 </beans>
377
378

```

20. Project의 Bean 등록 및 의존 관계 설정

- 1) <context:component-scan> tag 사용
- 2) @Service, @Repository Annotation을 선언한 class들과 @Autowired Annotation을 선언하여 의존관계를 설정한 Class들이 위치한 Package를 Scan하기 위한 설정을 XML에 해주어야 한다.
- 3) beans.xml에 다음 code 추가한다.

```

384     <context:component-scan base-package="com.example" />
385
386

```

21. Spring TestContext Framework 사용하기

- 1) MvnRepository에서 'spring test'로 검색
- 2) Spring TestContext Framework에서 5.3.100 click
- 3) 다음 코드를 pom.xml에 붙여 넣기

```

392     <!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->
393     <dependency>
394         <groupId>org.springframework</groupId>
395         <artifactId>spring-test</artifactId>
396         <version>5.3.100</version>
397         <scope>test</scope>
398     </dependency>
399

```

- 4) pom.xml > right-click > Run As > Maven Install

```

401     [INFO] BUILD SUCCESS 확인
402

```

- 5) /src/com.example.test package 생성
- 6) /src/com.example.test > right-click > New > JUnit Test Case
- 7) Name : MybatisDemoTest > Finish
- 8) [New JUnit Jupiter test] 선택
- 9) Finish

```

409     import org.junit.jupiter.api.Test;
410     import org.junit.jupiter.api.extension.ExtendWith;
411     import org.springframework.beans.factory.annotation.Autowired;
412     import org.springframework.test.context.ContextConfiguration;
413     import org.springframework.test.context.junit.jupiter.SpringExtension;
414
415     import com.example.service.CityService;
416     import com.example.vo.CityVO;
417
418     @ExtendWith(SpringExtension.class)
419     @ContextConfiguration(locations="classpath:beans.xml")
420     public class MybatisDemoTest {
421         @Autowired
422         private CityService cityService;
423
424         @Test
425         public void test() {
426             CityVO city = cityService.getCity("Seoul");
427             System.out.println(city);
428         }
429     }
430

```

- 10) 만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면

- 해당 Project > right-click > Build Path > Libraries tab
- spring-test-5.3.100.jar 선택 후 [Remove] 로 삭제
- [Add External JARs...] Click
- Local M2 Repository(e.g C:\Users\bluee\.m2\repository\org\springframework\spring-test\5.3.100)에서 직접 jar를 선택할 것
- [Order and Export] tab에서 spring-test-5.3.100.jar 선택 후 [Up] button을 클릭
- 해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click

22. Test

```
441 1)right-click > Run As > Junit Test
442 2)결과 -> Junit View에 초록색 bar
443 CityVO(id=2331, name=Seoul, countryCode=KOR, district=Seoul, Population=9981619)
444
445
```

23. All City 읽어오기

```
447 1)com.example.service.CityServiceImpl.java
```

```
449 @Override
450 public List<CityVO> getCityList() {
451     return cityDao.readAll();
452 }
453
```

```
454 2)com.example.dao.CityDaoImpl.java
```

```
456 @Override
457 public List<CityVO> readAll() {
458     List<CityVO> cityList = session.selectList("City.selectList");
459     return cityList;
460 }
461
```

```
462 3)mybatis-mapper.xml
```

```
464 <select id="selectList" resultType="cityVO" resultMap="cityResult">
465     SELECT * FROM world.city ORDER BY id DESC
466 </select>
467
```

```
468 4)MybatisDemoTest.java
```

```
469 @Autowired
470 CityService cityService;
471
472 @Disabled @Test
473 public void test() {
474     CityVO city = this.cityService.getCity("Seoul");
475     System.out.println(city);
476 }
477
478 @Test
479 public void test1() {
480     List<CityVO> list = this.cityService.getCityList();
481
482     for(CityVO vo : list){
483         System.out.println(vo.getId());
484         System.out.println(vo.getName());
485         System.out.println(vo.getDistrict());
486         System.out.println(vo.getCountryCode());
487         System.out.println(vo.getPopulation());
488         System.out.println("-----");
489     }
490 }
491
492
493
```

24. Test

```
495 1)right-click > Run As > Junit Test
496 2)결과 -> Junit View에 초록색 bar
497
498
499
```

Task2. SungjukManagement Project with MyBatis

1. Stuent table creation

```
503 CREATE TABLE scott.Student (
504     hakbun CHAR(4), /* 학번 */
505     name VARCHAR2(20) NOT NULL, /* 이름 */
506     kor NUMBER(3) DEFAULT 0 NOT NULL, /* 국어 */
507     eng NUMBER(3) DEFAULT 0 NOT NULL, /* 영어 */
508     mat NUMBER(3) DEFAULT 0 NOT NULL, /* 수학 */
509     edp NUMBER(3) DEFAULT 0 NOT NULL, /* 전산 */
510     tot NUMBER(3), /* 총점 */
511     avg NUMBER(5,2), /* 평균 */
512     grade CHAR(1), /* 평점 */
513     CONSTRAINT student_hakbun_pk PRIMARY KEY(hakbun)
514 );
```

515
516
517 2. SungjukMgmt project 생성
518 1)In Package Explorer > right-click > New > Java Project
519 2)Project name : SungjukMgmt
520 3)JRE : Use default JRE 'jdk-11.0.12' and workspace compiler preferences
521 4)Uncheck [Create module-info.java file]
522 5)Next
523 6)Finish
524
525

526 3. vo package 생성
527 1)src > right-click > New > Package
528 2)Name : com.example.vo
529 3)Finish
530 4)com.example.vo.StudentVO class 생성
531

```
532     package com.example.vo;  
533  
534     public class StudentVO {  
535         private String hakbun;  
536         private String name;  
537         private int kor;  
538         private int eng;  
539         private int mat;  
540         private int tot;  
541         private double avg;  
542         private char grade;  
543     }  
544  
545
```

546 4. service package 생성
547 1)src > right-click > New > Package
548 2)Name : com.example.service
549
550 3)StudentService interface 생성
551 -com.example.service > right-click > New > Interface
552 -Name : StudentService
553 -Finish
554

```
555     package com.example.service;  
556  
557     import java.util.List;  
558  
559     import com.example.vo.StudentVO;  
560  
561     public interface StudentService {  
562         void create(StudentVO student);  
563         StudentVO read(String hakbun);  
564         List<StudentVO> readAll();  
565         void update(StudentVO student);  
566         void delete(String hakbun);  
567     }  
568
```

569 4)StudentServiceImpl class 생성
570 -com.example.service > right-click > New > Class
571 -Name: StudentServiceImpl
572 -Interfaces : com.example.service.StudentService
573

```
574     package com.example.service;  
575  
576     import java.util.List;  
577  
578     import com.example.vo.StudentVO;  
579  
580     public class StudentServiceImpl implements StudentService {  
581  
582         @Override  
583         public void create(StudentVO student) {  
584             // TODO Auto-generated method stub  
585  
586         }  
587  
588         @Override
```



```

589     public StudentVO read(String hakbun) {
590         // TODO Auto-generated method stub
591         return null;
592     }
593
594     @Override
595     public List<StudentVO> readAll() {
596         // TODO Auto-generated method stub
597         return null;
598     }
599
600     @Override
601     public void update(StudentVO student) {
602         // TODO Auto-generated method stub
603     }
604 }
605
606     @Override
607     public void delete(String hakbun) {
608         // TODO Auto-generated method stub
609     }
610 }
611 }
612 }
613
614
615 5. dao package 생성
616 1)src > right-click > New > Package
617 2)Name : com.example.dao
618 3)Finish
619 4)StudentDao interface 생성
620 -com.example.dao > right-click > New > Interface
621 -Name : StudentDao
622 -Finish
623
624     package com.example.dao;
625
626     import java.util.List;
627
628     import com.example.vo.StudentVO;
629
630     public interface StudentDao {
631         void insertStudent(StudentVO student);
632         StudentVO selectStudent(String hakbun);
633         List<StudentVO> selectAllStudent();
634         void updateStudent(StudentVO student);
635         void deleteStudent(String hakbun);
636     }
637
638 5)StudentDaoImpl class 생성
639 -com.example.dao > right-click > New > Class
640 -Name : StudentDaoImpl
641 -Interfaces : com.example.dao.StudentDao
642 -Finish
643
644     package com.example.dao;
645
646     import java.util.List;
647
648     import com.example.vo.StudentVO;
649
650     public class StudentDaoImpl implements StudentDao {
651
652         @Override
653         public void insertStudent(StudentVO student) {
654             // TODO Auto-generated method stub
655         }
656     }
657
658     @Override
659     public StudentVO selectStudent(String hakbun) {
660         // TODO Auto-generated method stub
661         return null;
662     }

```

```

663
664     @Override
665     public List<StudentVO> selectAllStudent() {
666         // TODO Auto-generated method stub
667         return null;
668     }
669
670     @Override
671     public void updateStudent(StudentVO student) {
672         // TODO Auto-generated method stub
673
674     }
675
676     @Override
677     public void deleteStudent(String hakbun) {
678         // TODO Auto-generated method stub
679
680     }
681
682 }
683
684

```

6. Java Project를 Spring Project로 변환

1) SungjukMgmt Project > right-click > Configure > Convert to Maven Project

```

-Project : /SungjukMgmt
-Group Id : SungjukMgmt
-Artifact Id : SungjukMgmt
-version : 0.0.1-SNAPSHOT
-Packaging : jar
-Finish

```

2) SungjukMgmt Project > right-click > Spring > Add Spring Project Nature

7. pom.xml 파일에 Dependency 추가하기

```

<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.3.10</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>5.3.10</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.3.10</version>
  </dependency>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-api</artifactId>
    <version>5.6.2</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.20</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.26</version>
  </dependency>
  <dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.7</version>

```

```
737     </dependency>
738     <dependency>
739         <groupId>org.mybatis</groupId>
740         <artifactId>mybatis-spring</artifactId>
741         <version>2.0.6</version>
742     </dependency>
743 </dependencies>
744
745 -pom.xml > right-click > Run As > Maven install
746 [INFO] BUILD SUCCESS 확인
747
748
```

8. resource folder 생성

- 1) SungjukMgmt project > right-click > New > Source Folder
- 2) Folder name : resources
- 3) Finish

9. dbinfo.properties 파일 생성

- 1) /resources > right-click > New > File
- 2) File name : dbinfo.properties
- 3) Finish
- 4) Oracle Database 일 경우
db.driverClass=oracle.jdbc.driver.OracleDriver
db.url=jdbc:oracle:thin:@localhost:1521:XE
db.username=hr
db.password=hr

5) MySQL Database 인 경우

```
db.driverClass=com.mysql.jdbc.Driver
db.url=jdbc:mysql://localhost:3306/world
db.username=root
db.password=javamysql
```

10. Mybatis 관련 설정 파일 생성하기

- 1) mybatis-config.xml 생성
-resources > right-click > New > Other > XML > XML File > Next
-File name : mybatis-config.xml
-Finish

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
    PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <typeAliases>
        <typeAlias type="com.example.vo.StudentVO" alias="studentVO" />
    </typeAliases>
</configuration>
```

5) mybatis-mapper.xml 생성

- resources > right-click > New > Other > XML > XML File > Next
- File name : mybatis-mapper.xml
- Finish

```
<?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="Student">

</mapper>
```

10. Bean Configuration XML 작성

- 1) /resources > right-click > New > Other... > Spring > Spring Bean Configuration File > Next
- 2) File name : beans.xml
- 3) Finish
- 4) Namespace Tab click
- 5) Check context

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
```

```

811     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
812     xmlns:context="http://www.springframework.org/schema/context"
813     xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.3.xsd">
815
816     <context:property-placeholder
817         location="classpath:dbinfo.properties" />
818     <context:component-scan
819         base-package="com.example" />
820
821 </beans>
822
823
824 11. HikariCP pom.xml에 추가하기
825 1) https://mvnrepository.com/ 에서 'HikariCP' 검색
826 2) 다음 코드를 pom.xml에 추가
827
828     <dependency>
829         <groupId>com.zaxxer</groupId>
830         <artifactId>HikariCP</artifactId>
831         <version>5.0.0</version>
832     </dependency>
833
834 3) pom.xml > right-click > Run As > Maven Install
835 4) beans.xml 추가하기
836
837     <bean id="hikariConfig" class="com.zaxxer.hikari.HikariConfig">
838         <property name="driverClassName" value="${db.driverClass}" />
839         <property name="jdbcUrl" value="${db.url}" />
840         <property name="username" value="${db.username}" />
841         <property name="password" value="${db.password}" />
842     </bean>
843
844     <bean id="dataSource" class="com.zaxxer.hikari.HikariDataSource" destroy-method="close">
845         <constructor-arg ref="hikariConfig" />
846     </bean>
847
848 5) 추가적으로 다음을 테스트하였다.
849
850     <!-- c3P0 : 안됨
851     <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"
852         destroy-method="close">
853         <property name="driverClass" value="${db.driverClass}" />
854         <property name="jdbcUrl" value="${db.url}" />
855         <property name="user" value="${db.username}" />
856         <property name="password" value="${db.password}" />
857     </bean>
858     -->
859
860     <!-- Spring Jdbc : 잘됨
861     <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
862         <property name="driverClassName" value="${db.driverClass}" />
863         <property name="url" value="${db.url}" />
864         <property name="username" value="${db.username}" />
865         <property name="password" value="${db.password}" />
866     </bean>
867     -->
868
869     <!-- Apache DBCP2 : 잘됨
870     <bean id="dataSource"
871         class="org.apache.commons.dbcp2.BasicDataSource" destroy-method="close">
872         <property name="driverClassName" value="${db.driverClass}" />
873         <property name="url" value="${db.url}" />
874         <property name="username" value="${db.username}" />
875         <property name="password" value="${db.password}" />
876     </bean>
877     -->
878
879     <!-- Apache Tomcat CP : 잘됨 -->
880     <bean id="dataSource"
881         class="org.apache.tomcat.jdbc.pool.DataSource"
882         destroy-method="close">

```

```

883     <property name="driverClassName" value="${db.driverClass}" />
884     <property name="url" value="${db.url}" />
885     <property name="username" value="${db.username}" />
886     <property name="password" value="${db.password}" />
887     <property name="initialSize" value="20" /><!-- 초기 커넥션 갯수 -->
888     <property name="maxActive" value="50" /><!-- 최대 커넥션 갯수 -->
889     <property name="maxIdle" value="10" /><!-- 최대 유지 커넥션 갯수 -->
890 </bean>

```

12. Spring TestContext Framework 사용하기

- 1)src > right-click > New > Package
- 2)Name : com.example.test
- 3)Finish
- 4)com.example.test > right-click > New > JUnit Test Case
- 5>Select [New JUnit Jupiter test]
- 6)Name : StudentTest
- 7)Finish
- 8)Build Path에 junit-jupiter-api.5.8.1.jar와 spring-test-5.3.10.jar 추가하기

```

903     package com.example.test;
904
905     import static org.junit.jupiter.api.Assertions.fail;
906
907     import java.sql.Connection;
908     import java.sql.SQLException;
909
910     import javax.sql.DataSource;
911
912     import org.junit.jupiter.api.Test;
913     import org.junit.jupiter.api.extension.ExtendWith;
914     import org.springframework.beans.factory.annotation.Autowired;
915     import org.springframework.test.context.ContextConfiguration;
916     import org.springframework.test.context.junit.jupiter.SpringExtension;
917
918     import lombok.Setter;
919     import lombok.extern.java.Log;
920
921     @ExtendWith(SpringExtension.class)
922     @ContextConfiguration(locations = {"classpath:beans.xml"})
923     @Log
924     class StudentTest {
925         @Setter(onMethod_ = {@Autowired })
926         private DataSource dataSource;
927
928         @Test
929         public void test() {
930             try (Connection conn = this.dataSource.getConnection()){
931                 log.info(conn.toString());
932             } catch (SQLException e) {
933                 fail(e.getMessage());
934             }
935         }
936     }
937 }

```

- 9)right-click > Run As > Junit Test

- 10)결과 -> Junit View에 초록색 bar

```

942     INFO: HikariProxyConnection@1097069005 wrapping com.mysql.cj.jdbc.ConnectionImpl@2228db21
943

```

- 11)만일 다음과 같은 오류가 발생했을 경우

```

946     nested exception is java.lang.NoClassDefFoundError:
947     org.springframework.jdbc.datasource.TransactionAwareDataSourceProxy
948     -spring-jdbc dependency가 추가되어 있지 않아 발생했을 가능성이 높다.
949

```

13. MyBatis와 Spring 연동

- 1)beans.xml 다음 코드 추가

```

953     <bean id="sqlSessionFactory"
954         class="org.mybatis.spring.SqlSessionFactoryBean">
955         <property name="dataSource" ref="dataSource" />

```

```

956     <property name="configLocation"
957         value="classpath:mybatis-config.xml" />
958     <property name="mapperLocations">
959         <list>
960             <value>classpath:mybatis-mapper.xml</value>
961         </list>
962     </property>
963 </bean>
964 <bean id="sqlSession"
965     class="org.mybatis.spring.SqlSessionTemplate">
966     <constructor-arg ref="sqlSessionFactory" />
967 </bean>

```

969 2)StudentTest.java에 다음 코드 추가

```

970
971 @Setter(onMethod_ = {@Autowired })
972 private SqlSessionFactory sqlSessionFactory;
973
974 @Test
975 public void test() {
976     try(SqlSession session = sqlSessionFactory.openSession();
977         Connection conn = session.getConnection()){
978         log.info(session.toString());
979         log.info(conn.toString());
980     }catch(Exception e) {
981         fail(e.getMessage());
982     }
983 }

```

985 3)right-click > Run As > Junit Test

986 4)결과 -> Junit View에 초록색 bar

```

987
988 INFO: org.apache.ibatis.session.defaults.DefaultSqlSession@68ace111
989 INFO: HikariProxyConnection@1778877569 wrapping com.mysql.cj.jdbc.ConnectionImpl@5f13be1

```

992 14. StudentVO 수정

```

993
994 package com.example.vo;
995
996 import lombok.Getter;
997 import lombok.NoArgsConstructor;
998 import lombok.NonNull;
999 import lombok.RequiredArgsConstructor;
1000 import lombok.Setter;
1001 import lombok.ToString;
1002
1003 @NoArgsConstructor
1004 @RequiredArgsConstructor
1005 @Getter
1006 @ToString
1007 public class StudentVO {
1008     private @NonNull String hakbun, name;
1009     private @NonNull @Setter int kor, eng, mat, edp;
1010     private @Setter int tot;
1011     private @Setter double avg;
1012     private @Setter char grade;
1013 }

```

1016 15. Input.java & Calc.java 생성하기

1017 1)src/com.example/test/Input Class 생성

```

1018
1019 package com.example.test;
1020
1021 import java.io.File;
1022 import java.io.IOException;
1023 import java.util.List;
1024 import java.util.Scanner;
1025 import com.example.vo.StudentVO;
1026
1027 public class Input {
1028     private Scanner scanner;
1029     private List<StudentVO> list;

```

```

1030     public Input(List<StudentVO> list, String path) throws IOException {
1031         this.scanner = new Scanner(new File(path));
1032         this.list = list;
1033     }
1034     public void fileInput() {
1035         while(this.scanner.hasNextLine()) {
1036             String line = this.scanner.nextLine();
1037             Scanner lineScan = new Scanner(line).useDelimiter(",");
1038             StudentVO student = new StudentVO(lineScan.next().trim(),
1039                 lineScan.next().trim(), Integer.parseInt(lineScan.next().trim()),
1040                 Integer.parseInt(lineScan.next().trim()),
1041                 Integer.parseInt(lineScan.next().trim()), Integer.parseInt(lineScan.next().trim()));
1042             this.list.add(student);
1043         }
1044     }
1045 }
1046 }

```

2)src/com.example/test/Calc class 생성

```

1050     package com.example.test;
1051
1052     import java.util.List;
1053     import com.example.vo.StudentVO;
1054     public class Calc {
1055
1056         private List<StudentVO> list;
1057         public Calc(StudentVO student) {
1058             this.calc(student);
1059         }
1060         public Calc(List<StudentVO> list) {
1061             this.list = list;
1062             this.list.forEach(student -> this.calc(student));
1063         }
1064         private void calc(StudentVO student) {
1065             int tot = student.getKor() + student.getEng() + student.getMat() + student.getEdp();
1066             double avg = tot / 4.;
1067             char grade = avg >= 90 ? 'A' :
1068                 avg >= 80 ? 'B' :
1069                 avg >= 70 ? 'C' :
1070                 avg >= 60 ? 'D' : 'F';
1071             student.setTot(tot);
1072             student.setAvg(avg);
1073             student.setGrade(grade);
1074         }
1075     }

```

16. 성적 데이터 다운로드

- 1)<https://github.com/swacademy/Spring5>
- 2)resources/sungjuk_utf8.csv

17. 성적 데이터 Insertion

- 1)mybatis-mapper.xml

```

1086     <insert id="insertStudent" parameterType="studentVO" statementType="PREPARED">
1087         insert into Student(hakbun, name, kor, eng, mat, edp, tot, avg, grade)
1088         values(#{hakbun}, #{name}, #{kor}, #{eng}, #{mat}, #{edp}, #{tot}, #{avg}, #{grade})
1089     </insert>

```

- 2)com.example.dao.StudentDaoImpl.java

```

1092     package com.example.dao;
1093
1094     import java.util.List;
1095
1096     import org.apache.ibatis.session.SqlSession;
1097     import org.springframework.beans.factory.annotation.Autowired;
1098     import org.springframework.stereotype.Repository;
1099
1100     import com.example.vo.StudentVO;
1101
1102     import lombok.extern.java.Log;

```

```

1104     @Repository("studentDao")
1105     @Log
1106     public class StudentDaoImpl implements StudentDao {
1107         @Autowired
1108         private SqlSession session;
1109         ...
1110         @Override
1111         public void insertStudent(StudentVO student) {
1112             //log.info(student.getName());
1113             this.session.insert("Student.insertStudent", student);
1114         }
1115
1116 3)com.example.service.StudentServiceImpl.java
1117     package com.example.service;
1118
1119     import java.util.List;
1120
1121     import org.springframework.beans.factory.annotation.Autowired;
1122     import org.springframework.stereotype.Service;
1123
1124     import com.example.dao.StudentDao;
1125     import com.example.vo.StudentVO;
1126
1127     @Service("studentService")
1128     public class StudentServiceImpl implements StudentService {
1129         @Autowired
1130         private StudentDao studentDao;
1131
1132         @Override
1133         public void create(StudentVO student) {
1134             this.studentDao.insertStudent(student);
1135
1136 4)com.example.test.StudentJUnitTest.java
1137
1138     package com.example.test;
1139
1140     import java.util.ArrayList;
1141     import java.util.List;
1142     import java.util.Scanner;
1143
1144     import org.junit.jupiter.api.BeforeEach;
1145     import org.junit.jupiter.api.Disabled;
1146     import org.junit.jupiter.api.Test;
1147     import org.junit.jupiter.api.extension.ExtendWith;
1148     import org.springframework.beans.factory.annotation.Autowired;
1149     import org.springframework.test.context.ContextConfiguration;
1150     import org.springframework.test.context.junit.jupiter.SpringExtension;
1151
1152     import com.example.service.StudentService;
1153     import com.example.vo.StudentVO;
1154
1155     import lombok.extern.java.Log;
1156
1157     @ExtendWith(SpringExtension.class)
1158     @ContextConfiguration(locations = "classpath:beans.xml")
1159     @Log
1160     public class StudentJUnitTest {
1161         @Autowired
1162         private StudentService studentService;
1163         private List<StudentVO> list;
1164         private Scanner scan;
1165
1166         @BeforeEach
1167         public void init() {
1168             this.list = new ArrayList<StudentVO>(12);
1169             try{
1170                 Input input = new Input(list, "resources/sungjuk_utf8.csv");
1171                 input.fileInput();
1172                 Calc calc = new Calc(list);
1173             }catch(Exception ex) {
1174                 ex.printStackTrace();
1175             }
1176             this.scan = new Scanner(System.in);
1177         }

```



```

1178     @Test
1179     public void test() {
1180         //log.info("" + this.list.size());
1181         this.list.forEach(student -> this.studentService.create(student));
1182     }
1183
1184

```

18. 입력한 모든 학생 데이터 조회하기

1)mybatis-mapper.xml

```

1187
1188     <resultMap type="studentVO" id="studentResultMap">
1189         <result property="hakbun" column="hakbun"/>
1190         <result property="name" column="name"/>
1191         <result property="kor" column="kor"/>
1192         <result property="eng" column="eng"/>
1193         <result property="mat" column="mat"/>
1194         <result property="edp" column="edp"/>
1195         <result property="tot" column="tot"/>
1196         <result property="avg" column="avg"/>
1197         <result property="grade" column="grade"/>
1198     </resultMap>
1199     <select id="selectAllStudent" resultType="studentVO" resultMap="studentResultMap">
1200         select * from student order by hakbun desc
1201     </select>
1202

```

2)com.example.dao.StudentDaoImpl.java

```

1203
1204
1205     @Override
1206     public List<StudentVO> selectAllStudent() {
1207         return this.session.selectList("Student.selectAllStudent");
1208     }
1209

```

3)com.example.service.StudentServiceImpl.java

```

1210
1211
1212     @Override
1213     public List<StudentVO> readAll() {
1214         return this.studentDao.selectAllStudent();
1215     }
1216

```

4)com.example.test.StudentJUnitTest.java

```

1217
1218
1219     @Test
1220     public void test1() {
1221         this.studentService.readAll().forEach(student -> log.info(student.toString()));
1222     }
1223
1224

```

19. 특정 학생 조회하기

1)mybatis-mapper.xml

```

1225
1226
1227
1228     <select id="selectStudent" resultType="studentVO" resultMap="studentResultMap">
1229         select * from student where hakbun = #{hakbun}
1230     </select>
1231

```

2)com.example.dao.StudentDaoImpl.java

```

1232
1233
1234     @Override
1235     public StudentVO selectStudent(String hakbun) {
1236         return this.session.selectOne("Student.selectStudent", hakbun);
1237     }
1238

```

3)com.example.service.StudentServiceImpl.java

```

1239
1240
1241     @Override
1242     public StudentVO read(String hakbun) {
1243         return this.studentDao.selectStudent(hakbun);
1244     }
1245

```

4)com.example.test.StudentJUnitTest.java

```

1246
1247
1248     @Test
1249     public void test2() {
1250         System.out.print("학번 : ");
1251         String hakbun = scan.nextLine();

```

```
1252         log.info(this.studentService.read(hakbun).toString());
1253     }
1254
1255
```

1256 20. 특정 학생 삭제하기

1257 1)mybatis-mapper.xml

```
1258     <delete id="deleteStudent">
1259         delete from student where hakbun = #{hakbun}
1260     </delete>
1261
1262
```

1263 2)com.example.dao.StudentDaoImpl.java

```
1264
1265     @Override
1266     public void deleteStudent(String hakbun) {
1267         this.session.delete("Student.deleteStudent", hakbun);
1268     }
1269
```

1270 3)com.example.service.StudentServiceImpl.java

```
1271
1272     @Override
1273     public void delete(String hakbun) {
1274         this.studentDao.deleteStudent(hakbun);
1275     }
1276
```

1277 4)com.example.test.StudentJUnitTest.java

```
1278
1279     @Test
1280     public void test3() {
1281         System.out.print("학번 : ");
1282         String hakbun = scan.nextLine();
1283         this.studentService.delete(hakbun);
1284     }
1285
1286
```

1287 21. 특정 학생 데이터 수정하기

1288 1)mybatis-mapper.xml

```
1289     <update id="updateStudent" parameterType="studentVO">
1290         update student set kor = #{kor}, eng = #{eng}, mat = #{mat}, edp = #{edp},
1291         tot = #{tot}, avg = #{avg}, grade = #{grade}
1292         where hakbun = #{hakbun}
1293     </update>
1294
1295
```

1296 2)com.example.dao.StudentDaoImpl.java

```
1297
1298     @Override
1299     public void updateStudent(StudentVO student) {
1300         this.session.update("Student.updateStudent", student);
1301     }
1302
```

1303 3)com.example.service.StudentServiceImpl.java

```
1304
1305     @Override
1306     public void update(StudentVO student) {
1307         this.studentDao.updateStudent(student);
1308     }
1309
```

1310 4)com.example.test.StudentJUnitTest.java

```
1311
1312     @Test
1313     public void test4() {
1314         System.out.print("학번 : ");
1315         String hakbun = scan.nextLine();
1316         StudentVO student = this.studentService.read(hakbun);
1317         student.setKor(100);
1318         student.setEng(100);
1319         student.setMat(100);
1320         student.setEdp(100);
1321         Calc calc = new Calc(student);
1322         this.studentService.update(student);
1323     }

```