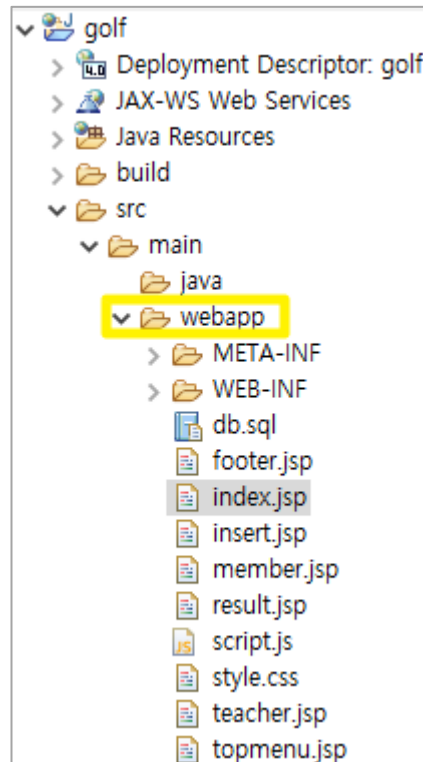


23-11-28

- 이클립스
- Jsp

톰캣 실행했을 때 작동하지 않았던 것이
webapp아래에 넣어야 작동이 된다.



톰캣파일안에 넣는것과 eclipse webapp안에 넣는 것 둘다 해줘야 작동이 되는건가?

톰캣과 Eclipse는 따로 봐야한다.

톰캣은 서버이고 , eclipse는 파일을 한번에 보면서 톰캣이라는 서버를 사용할 수 있는 것이다
그래서 파일을 tomcat ROOT안에 넣어서 작동하는 것은 톰캣 서버만을 가지고 사용하는 것이다.
(예전에는 이렇게 사용했었다. 클래식 방법)

Webapp에 넣어주는 것은 eclipse에서 톰캣을 연결해서 사용하도록 만들기 위해서이다. 서버를 통
해서 연결해주려면 webapp > WEB_INF파일 안이 아니라 webapp에 있어야 함
(지금까지 작업을 WEB_INF에서 했었다)

이클립스로 여러서버를 사용가능(톰캣, SQL 등)

```
<%@ page import="java.util.*" %>
<%@ page import="java.sql.*" %>
```

import 개념 : 내가 필요한 라이브러리를 가지고 와서 사용하는 것
필요없는 라이브러리도 있으니까 전체 가지고 오면 효율이 떨어짐

Sql.* -> sql의 관련 200개가 있다고 치면 200개 다 가지고 오겠다는 것(전체를 가지고 오기 때문에 1,2초의 딜레이 발생할 수도 있다, 개발자 부분에서는 전체 가지고 오는 것이 편하지만)
sql.select 이렇게 사용할 수도 있다. (전체부분이 아니라)

스프링은 jsp를 사용 8이나 11을 더 많이 사용

All Modules	Java SE	JDK	Other Modules
Module	Description		
java.base	Defines the foundational APIs of the Java SE Platform.		
java.compiler	Defines the Language Model, Annotation Processing, and Java Compiler APIs.		
java.datatransfer	Defines the API for transferring data between and within applications.		

<https://docs.oracle.com/en/java/javase/11/docs/api/index.html>

자바 오라클 문서에 모듈이다(import 설명에 필요한)

예전에는 java EE로 갔어야 했다

Java 활성화된 툴이 이클립스

자바8-> 11이 되면서 굳이 java EE를 사용하지 않는다.

그러면서 웹용 개발이 아니라 응용프로그램으로도 사용하게 되었다.

웹 단 보여주기 위해 사용했는데 앱 프로그램 형태를 사용할 때도 se?

<사이트>

<https://docs.oracle.com/en/java/javase/index.html>

정보들이 버전별로 이 문서에 다 담겨있다. 아래 사이트들을 통틀어서 보게 되는 것이 이 페이지

<https://dev.java/learn/>

<https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>

teacher.jsp

```
try{
    Class.forName("oracle.jdbc.OracleDriver");
    Connection con = DriverManager.getConnection
        ("jdbc:oracle:thin:@//localhost:1521/xe","system","1234");
```

기존문서 가지고 온 것, 뒤에 system과 1234만 바꿔주면 된다.

오라클 DB 연동해서 가지고 오라는 내용!

```
while(rs.next()){%
```

리턴하는 것

```
<tr>
    <td><%=rs.getString("teacher_code") %></td>
    <td><%=rs.getString("teacher_name") %></td>
    <td><%=rs.getString("class_name") %></td>
    <td><%=rs.getString("class_price") %></td>
    <td><%=rs.getString("teacher_regist_date") %></td>
</tr>
```

?

Action.jsp

여기에 사용하는 내용을 db에 데이터 집어넣을 때 사용한다(수강신청 화면에서)

이 파일은 전체 함수로 되어있다.

```
String regist_month = request.getParameter("regist_month");
String c_no = request.getParameter("c_no");
String class_area = request.getParameter("class_area");
String tuition = request.getParameter("tuition");
String teacher_code = request.getParameter("teacher_code");
```

http요청의 파라미터 값을 얻기 위해 사용하는 것이 request.getParameter() 메소드이다.

클라이언트가 설정한 파라미터를 가지고와 지정해주는 것으로 해석했다.

```
String sql = "INSERT INTO TBL_CLASS_202201 ";
sql+= "VALUES (?, ?, ?, ?, ?)";
```

values 값 5개 집어넣어라

```
PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, regist_month);
pstmt.setString(2, c_no);
pstmt.setString(3, class_area);
pstmt.setInt(4, Integer.parseInt(tuition));
pstmt.setString(5, teacher_code);
```

DB의 필드명이다.

#문제1

join문제이다 standard(sql시험문제)

다음 SQL 중 결과가 다른 SQL은 무엇인가?

- ① SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 >= B.COL1
AND A.COL1 <= B.COL2;
- ② SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 BETWEEN B.COL1 AND B.COL2;
- ③ SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE B.COL1 <= A.COL1
AND B.COL2 >= A.COL1;
- ④ SELECT B.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 >= B.COL1
AND A.COL1 <= B.COL2;

1,3번은 부등호가 다르지만 결과는 같게 나온다
select값이 다른 4번이 정답

```
1 SELECT A.PRODUCT_NAME,  
2     B.MEMBER_ID,  
3     B.CONTENT,  
4     C.EVENT_NAME  
5 FROM PRODUCT A,  
6     PRODUCT_REVIEW B,  
7     EVENT C  
8 WHERE A.PRODUCT_CODE = B.PRODUCT_CODE  
9     AND B.REG_DATE BETWEEN C.START_DATE AND C.END_DATE;
```

#	PRODUCT_NAME	MEMBER_ID	CONTENT	EVENT_NAME
1	무소음 무선 마우스	sqlchild02	무선이라 정말 편하네요!	20% 할인쿠폰 증정
2	무소음 무선 마우스	sqlbaby01	무소음인데 소음이 조금 있는 듯?	20% 할인쿠폰 증정
3	기계식 게이밍 키보드	sqladult03	게임할 맛 납니다	마우스패드 증정

where 의 첫번째 줄은 inner join문이다 a의 코드값과 b의 코드 값 같을 때 출력

AND c의 시작데이터와 c의 끝데이터 사이에 존재하는 값들 중에 b.REG_date의 값이 있으면 출력
(나는 위에 처럼 이해했다)

between은 범위를 지정하는 것

#문제2

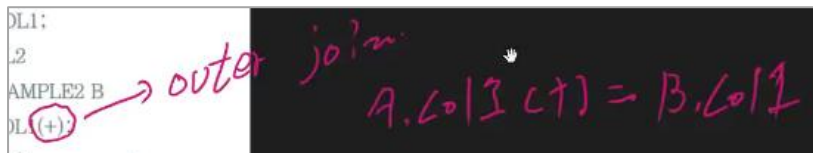
다음 SQL 중 결과가 다른 SQL은 무엇인가?

- ① SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 = B.COL1;
- ② SELECT A.COL1, B.COL2
FROM SAMPLE1 A INNER JOIN SAMPLE2 B
ON A.COL1 = B.COL1;
- ③ SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 = B.COL1(+);
- ④ SELECT B.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 = B.COL1;

4번은 select 다르지만 조건이 a.col과 b.col이 같아서 a.col1의 값과 동일하게 나올 것

3번은 (+)가 붙어 있는데, +붙으면 OUTER JOIN이다.

1,2,4번은 INNER JOIN이다.



코드는 줄여서 사용하지만 SQL문은 줄여서 사용하지 말기->약어 처리하면 나중에 판독할 때 어려움을 느낀다

#문제3

As 뒤에 오는 것은 신규 필드명이다.

sum은 합계출력

count는 null이 아닌 레코드의 개수(개수출력)

distinct는 중복없이 개수출력

[SQL]
SELECT SUM(A.COL2) AS RESULT1,
COUNT(A.COL3) AS RESULT2,
COUNT(DISTINCT A.COL3) AS RESULT3
FROM SAMPLE A, SAMPLE2 B
WHERE A.COL1 = B.COL1;

[SAMPLE1 테이블]

COL1	COL2	COL3
가	100	A
나	110	A
다	200	NULL
라	150	B
마	50	NULL

[SAMPLE2 테이블]

COL1	COL2
가	10
나	0
다	5

가장 먼저 확인

[SQL]
SELECT SUM(A.COL2) AS RESULT1,
COUNT(A.COL3) AS RESULT2,
COUNT(DISTINCT A.COL3) AS RESULT3
FROM SAMPLE A, SAMPLE2 B
WHERE A.COL1 = B.COL1;

[SAMPLE1 테이블]

COL1	COL2	COL3
가	100	A
나	110	A
다	200	NULL
라	150	B
마	50	NULL

[SAMPLE2 테이블]

COL1	COL2
가	10
나	0
다	5

Result1	Result2	Result3
410	2	1

→ 3개에

#문제4

다음 두 테이블을 COL1을 JOIN 컬럼으로 하여 LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN했을 때 각각 출력되는 데이터 건수로 가장 적절한 것은 무엇인가? (단, SAMPLE1 테이블이 LEFT TABLE, SAMPLE2 테이블이 RIGHT TABLE이라고 가정)

[SAMPLE1 테이블]

COL1	COL2	COL3
1	A	D
3	B	E
5	C	F

[SAMPLE2 테이블]

COL1	COL4
2	G
3	H
3	I

다음 두 테이블을 COL1을 JOIN 컬럼으로 하여 LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN했을 때 각각 출력되는 데이터 건수로 가장 적절한 것은 무엇인가? (단, SAMPLE1 테이블이 LEFT TABLE, SAMPLE2 테이블이 RIGHT TABLE이라고 가정)

COL1	COL2	COL3
1	A	D
3	B	E
5	C	F

COL1	COL4
2	G
3	H
3	I

L =
R =
F =

①) 1회

C1	C2	C3	C1	C4
1	A	D	NULL	NULL
3	B	E	3	H
3	B	E	3	I
5	C	F	NULL	NULL
NULL	NULL	NULL	2	G

full join했을 때 순서가 중요하다 (1->3->3->5->2)

C1	C2	C3	C1	C4
1	A	D	NULL	NULL
3	B	E	3	H
3	B	E	3	I
5	C	F	NULL	NULL
NULL	NULL	NULL	2	G

<sql로 실행>

```
select * from sample1 s1 FULL OUTER JOIN sample2 s2 on s1.col1=s2.col1;
```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 5(0.011초)

	COL1	COL2	COL3	COL1_1	COL4
1	(null)	(null)	(null)	2	G
2	3	B	E	3	H
3	3	B	E	3	I
4	1	A	D	(null)	(null)
5	5	C	F	(null)	(null)

<pre>select * from sample1 s1 left OUTER JOIN sample2 s2 on s1.col1=s2.col1;</pre>				
<div> <div>스크립트 출력 x</div> <div> <div>▶</div> <div>질의 결과 x</div> </div> </div>				
<div> <div> <div></div> <div></div> <div></div> </div> <div>SQL 인출된 모든 행: 4(0,003초)</div> </div>				
COL1	COL2	COL3	COL1_1	COL4
1	3 B	E	3 H	
2	3 B	E	3 I	
3	1 A	D	(null)	(null)
4	5 C	F	(null)	(null)

<pre>select * from sample1 s1 right OUTER JOIN sample2 s2 on s1.col1=s2.col1;</pre>				
<div> <div> <div>스크립트 출력 x</div> <div> <div>▶</div> <div>질의 결과 x</div> </div> </div> </div>				
<div> <div> <div></div> <div></div> <div></div> </div> <div>SQL 인출된 모든 행: 3(0,004초)</div> </div>				
COL1	COL2	COL3	COL1_1	COL4
1	3 B	E	3 I	
2	3 B	E	3 H	
3	(null)	(null)	2 G	

#문제5

A.col1=B.col2 니까 on기준이 먼저라서

A의 col1이 먼저 나온다

그런데 right outer join이니까 오른쪽기준의

테이블이 모두 나올 것이다

[SQL]
SELECT *
FROM SAMPLE1 A RIGHT OUTER JOIN SAMPLE2 B
ON (A.COL1 = B.COL1 AND B.COL2 IS NOT NULL);

[SAMPLE1 테이블]

COL1	COL2
2	G
3	H
3	I

[SAMPLE2 테이블]

COL1	COL2
1	A
2	B
3	NULL
4	D
5	E

Handwritten notes and calculations:

COL1	COL2	COL1-1	COL2-1
2	G	2	B

out

COL1	COL2	COL1	COL2
NULL	NULL	1	A
2	G	2	B
NULL	NULL	3	NULL
NULL	NULL	4	D
NULL	NULL	5	E

[SQL]
SELECT *
FROM SAMPLE1 A RIGHT OUTER JOIN SAMPLE2 B
ON (A.COL1 = B.COL1 AND B.COL2 IS NOT NULL);

[SAMPLE1 테이블]

COL1	COL2
2	G
3	H
3	I

[SAMPLE2 테이블]

COL1	COL2
1	A
2	B
3	NULL
4	D
5	E

ON

1	2	1	2
N	N	1	A
2	G	2	B
3	H	3	N
3	I	3	N
N	N	4	D
N	N	5	E

->

ON

1	2	1	2
N	N	1	A
2	G	2	B
3	H	3	N
3	I	3	N
N	N	4	D
N	N	5	E

select
from outer join
where
group by
having
order by

조건 순위

1. where (where가 없으면 on이 기준이된다)

#문제 6

inner join

outer join 이라는 말이 없으니까

Full 이다.

A.col1 의 값과

B.col2 의 값이 전체 나온다

Inner join 과 outer join 만 null 값이 나옴!

c1	c2	c1	c2
2 g	1 55		
2 g	2 12		
2 g	3 30		
2 g	4 33		
2 g	5 20		
4 i	1 55		
4 i	2 12		
4 i	3 30		
4 i	4 33		
4 i	5 20		

->

c1	c2	c1	c2
2 g	1 55		
2 g	3 30		
2 g	5 20		
4 i	1 55		
4 i	3 30		
4 i	5 20		

다음 SQL의 결과로 가장 적절한 것은 무엇인가?

[SQL]
SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE B.COL3 % 5 = 0;

[SAMPLE1 테이블]

COL1	COL2
2	G
4	I

[SAMPLE2 테이블]

COL1	COL3
1	55
2	12
3	30
4	33
4	20

Where조건 하지 않았을 때 전체 데이터는
왼쪽 표와 같을 것.

where조건을 하게 되면 오른쪽 표와 같이
해당되지 않는 데이터는 지워진다

이후 select a.col1과 b.col3만 출력하면
6개의 값이 출력됨

위의 것처럼 사용하기 위한 예시가 다른 테이블의 데이터를 합칠 때
 중복되는 값들을 입력하기 위하여 합칠 때 사용할 수 있다.
 아래와 같이 회원번호가 상세테이블에 중복으로 들어간다.

N		O	P	Q	R	S	T
회원테이블					상세테이블		
회원번호	성명	주민번호			회원번호	전화번호	주소
1001	박원일	710901-			1001	010-6463-8218	인천광역시
데이터 채워줄때					1001	010-2222-3333	인천광역시
					1001	010-2222-3333	경기도 수원시

DB 사용하기 위한 것은 저장공간 줄이기 위해서

테이블 쪼개는 것 제 1 정규화?

데이터 중복 데이터 제거 제 1 정규화

필드 데이터 값

제 1 정규형은

집주소, 직장테이블 ,부모연락처로 테이블나뉘서 쪼개는 것이 제 1 정규형 (테이블 4 개)

➔ 전체 데이터에서 직장테이블 부모연락처 테이블로 나눠 생성했다

학생번호만 primary key 이다.

학생					수강내역				
학번	이름	메일	대학	전공	학번	수강학기	과목명	성적	제한인원
101	Tom	tom@naver.com	이과대학	생물학과	101	2016.1	생물1	A	20
102	Dan	Dan@naver.com	IT대학	컴퓨터과	101	2016.1	생물2	B	30
103	Jin	jin@google.com	정경대학	경제학과	102	2016.2	C++	A	40
					102	2017.1	C언어	C	40
					103	2017.2	경제	A	30

제 2 정규형은

부모연락처에서 부와 모를 따로 생성했다

학생번호	부모	성명	연락처
1001	부	우리빠	010-1111-2222
1002	부	111	010-1111-3333
1003	부	김땡땡	010-2222-1111
1004	부		010-2222-2222
1012	부		
1013	부	마이콜	010-1234-1234
1014	부	김철수	010-7891-6515
1015	부	부	010-4443-3333
1001	모	우리마	010-3333-4444
1002	모	222	010-3333-5555
1003	모	임땡땡	010-4444-6666
1004	모		010-4444-7777
1012	모		010-2233-5684
1013	모	오마니	010-111-1234
1014	모	윤영희	010-1515-7878
1015	모	모	010-2222-4444

제 3 정규형은

필드형에 대한 데이터 줄여서 집어넣는 것(부에 해당되지 않는 데이터 삭제)

#문제 7

2) 다음 SQL 중 에러가 발생하지 않는 SQL은 무엇인가? (Oracle이라고 가정)

- ① SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE B.COL2 IS NULL;
- ② SELECT A.COL1, B.COL2
FROM SAMPLE1 A NATURAL JOIN SAMPLE2 B
ON A.COL1 = B.COL1;
- ③ SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1(+) = B.COL1(+);
- ④ SELECT B.COL1, B.COL2
FROM SAMPLE1 A JOIN SAMPLE2 B
USING (COL1, COL2);

정답 ①

② Natural join 은 on 을 사용할 수 없다 -> where 사용

③ +가 나오면 무조건 outer join 이다. 동시에 사용할 수 없다(left 와 right)

왼쪽에 +가 붙여지면 right outer join 이고 오른쪽에 +붙인거는 left outer join 이다.

L1 = B.COL1;
L1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B;
L1(+) = B.COL1(+);
L1, B.COL2
FROM SAMPLE1 A JOIN SAMPLE2 B

outer join
LEFT
Right

④ 테이블 지정 시에 사용하는 것이 using, join 시에는 사용할 수 없다.

<MVC 패턴>

~~Mvc 는 원래 백단 마지막에 배우는 것이다~~

1. **Model** (완벽한 백단, 백단은 db 서버의 관련된 것이다)

- DAO (data Access Object) : 자료를 읽어오기 위한 객체, DB 에서 읽어서 JDBC 를 통해서 bms 에 자료를 요청하는 구조

2. **View** (웹서버 쪽에 관련된 것들)

목록 필요할 때 List.do 추가

수정 필요할 때 modify.do 추가 (사람마다 쓰는 이름이 다름, 정해져 있는 것이 아님)

- Do (data object) : 회원가입할 때 입력할 수 있는

3. **Controller**

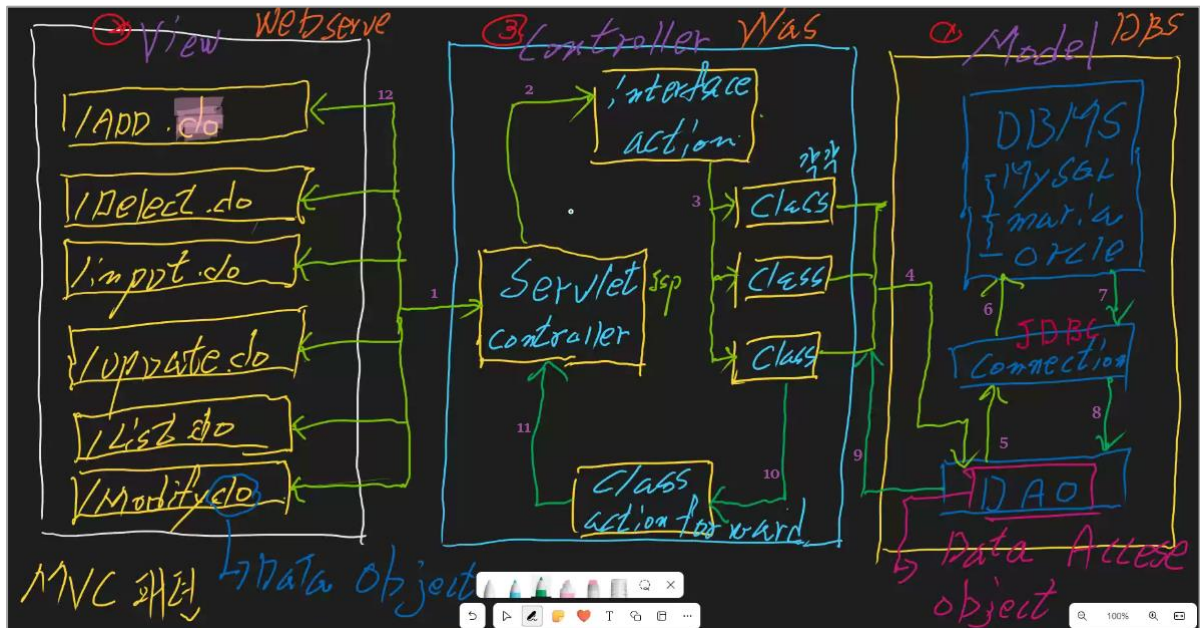
이 안에서 구성하는 모든 것이 기본적으로 class 의 개념이다.

여기서 class 는 작업하는 하나의 단위로 자바의 class 를 말함

- DTO : 데이터 처리하는 컨트롤러와 컨트롤러간의 의사소통
서로 통신할 때 규격이 필요하다 그 규격이 DTO 이다.
- ➔ Spring 과 boot 로 가면 전체 구조에서 controller 부분이 달라짐

Jsp 는 spring 때 다시 할 것.

상세모델



뷰는 웹브라우저랑은 일단은 상관없음, 웹 서버랑 관련있다

추가해야 하는 내용 필요한 .do 를 호출(회원가입해서 신규다 그러면 add.do 추가)

각각하는 일들이 달라서 do 들 쪼개놓은 것

DAO 로 바뀌서 model 의 DAO 로 보낸다 그 보내는 과정이다

=> DAO 에서 JDBC 로 보내고 JDBC 에서는 DBMS 로 보낸다(호출) 그리고 거기서 원하는 자료를 받아온다. 받아온 자료들을 다시 뷰로 보낸다

보내는 과정안에서 그 값들을 controller 안의 controller 로 보낸다

controller 에서 interface 로 보내고 그 인터페이스에서 각각의 class 에 뿌려준다

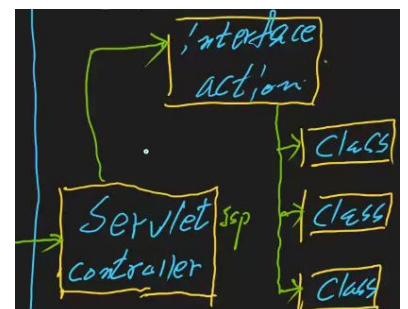
이 클래스에서 각각 작업한 내용을 DAO 에 질문을 던진다

그리고 나서 class 에서 받은 내용이 DAO 로 가고 JDBC 로 간다

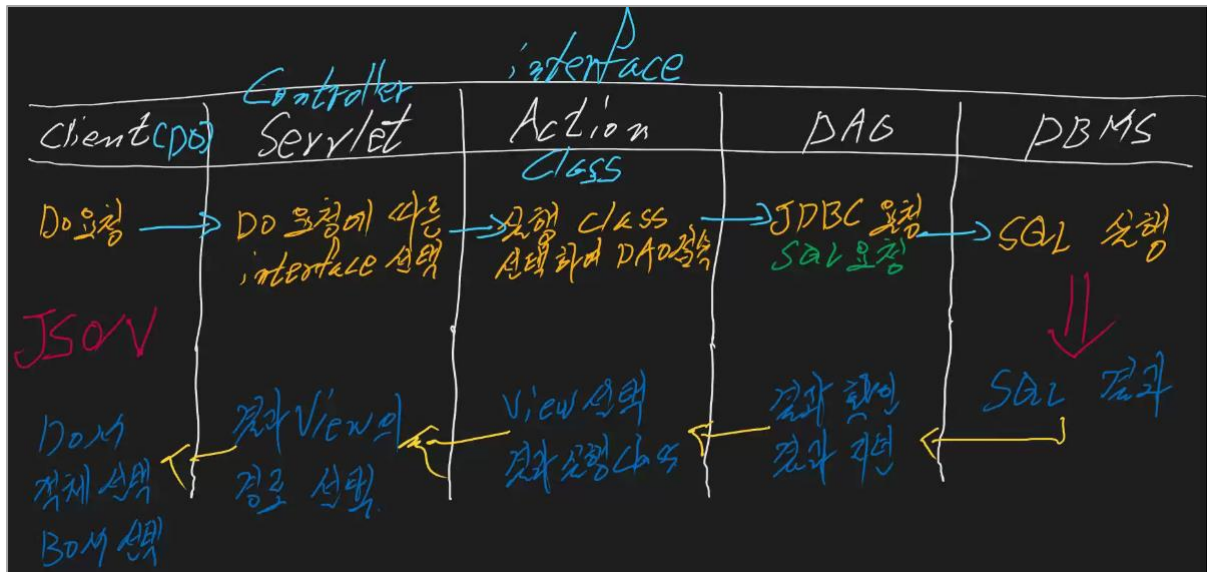
받아온 값들은 DAO 에서 각각의 class 에 쏜다

그리고 해당하는 값이 class 밑으로 간다 (classaction forward 이다.)

받은 값들을 do 에 저장



더 세부사항으로 나누게 되면 아래와 같다.



Do : 요청

Servlet : do 요청에 따른 interface 선택

Action(interface): 실행 class 선택하여 DAO 접속하는 것

DAO : 인터페이스에서 요청이 들어오면 JDBC에 요청하는 것 (SQL로 요청)

DBMS : SQL 실행

⇒ SQL 실행하면 결과도 있다. 그 결과는 DAO로 보냄

DAO

⇒ 결과 확인, 결과 return -> 각각 필요한 class에 보냄

Action(interface)

⇒ View를 선택 (즉, 결과 실행 class를 선택한다는 것)

Servlet

⇒ 인터페이스 선택하는 것도 있지만, 결과 view에 대한 경로 선택

Do

⇒ 경로선택한 라인을 DOM 객체 선택 -> DOM 선택 : 백단으로 신호를 보내기 시작

신호의 시작이 JSON 형태로 할 것 ! JSON 형태로 자료 받을 것이다