181206

//원래 ibatis 였는데 나중에 팀에서 나와서 mybatis로 바꿈

//처음 깔고 개발 환경 구축하는 과정이 복잡

// Java (Myabatis 개발 환경 순서)

1.드라이버, 라이브러리 준비. 오라클 연결, 마이바티스

+드라이버) C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib\ojdbc6.jar

+라이브러리) C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext

여기에다가 ojdbc6.jar 랑 mybatis의 jar 파일을 넣어 두면 자동 인식.

jdk 안에 외부 라이브러리 추가하는 곳.

버전을 잘 맞춰야 함. 자바, 마이바티스, 이클립스, 등등 서로 앞서거니 뒷서거니

하면서 계속 새 버전이 나옴. 호환 맞춰야 함.

jdk 이클립스 /오라클-> 자바랑은 다른 애라서 괜춚

마이바티스-> 자바랑 오라클 둘 다 신경써야 함

자바에서 만든 코딩을 DB를 통해 저장해야 하는데, 그 사이에 있는 게 드라이버.

2.프로젝트 생성

3.환경 설정

4.mybatis 관련 객체 생성

5.db에 테이블 생성

//DB와 Java를 항상 통화 중, 실시간으로 만들어 놓기는 어려움. 필요할 때만 전화 걸어서 자료 넘기고, 얼른 끊어야.

openSession(); 은 전화 걸기. 근데 전화 걸어서 못 받을 수도 있지.

전화 못 받으면 exception. 전화 받고 자료 다 넘기면 close();

dao에 자료 다 넣어 놓음. 다른 데에서는 요청만 주고 받아야.

//다른 사람들이 오라클 쓸 줄 아는 사람이 들어가서 수정하고 쓰라고

properties 클래스 따로 만들어서 정보를 넣어 둠.

주석은 달지 말게

1521은 포트 번호. 기본 번호임. 이거 바로 들킴. 기본으로 쓰지 않음.

//comfig.xml - 밑에 design 화면 말고 source 화면

<configuration>과 </configuration> 사이에 모든 환경 설정이 다들어가 있어야

property 에서 driver랑 url, username 등등을 여기서 정확히 바꿔 줘야 오라클이랑 연결이 가능해 짐.

//typeAliases (에일리어스)

에서 줄임말을 등록 해놓은 거임. 그냥 이름만으로 불러지는 게 아님.

Mapper 에서 부를 때 줄여서 부름.

//오라클 드라이버 파일/바이바티스 라이브러리 파일 두 개 준비

+)파일만 있으면 몰라.

+)마이바티스의 db.properties 에 드라이버, DB주소, 아이디 비번 저장해 놔야.

+)마이바티스 설정 파일

mybatis-config.xml 에 설정 파일

그걸 config에 넣어 둠.

+)mapper 가 보통 한 프로젝트에 수십 개씩 됨.

한 줄씩 추가하면 됨

+) math는 new로 객체를 못 만들어. 생성자를 private 지정해버렸어

calendar는 getInstance 명령어를 따로 만들어서 필요할 때 하나씩 만들어서 쓰라고.

mybatis는 생성자를 실행하면 static으로 객체를 생성하게 해놓고,

static { ~~~} 해서 static 초기화? 라는 걸 해서, 객체를 만들고 시작.

이거는 한 번만 실행 되는 것.

+)getMapper 구문 -> 인터페이스 이름을 받아 와서, 그 이름이랑 같은 클래스를 만든

다음에, 그 클래스에서 객체를 만들고, 우리가 그 객체를 통해 실행할 수 있게끔 해줌.

+) ~Mapper.xml 은 ~Mapper 인터페이스의 이름을 위에

mapper namespace = "ffas.adsf.ads" 경로를 지정해줘야 함.

+) inserPerson 이나 updatePerson 이런 거에서 메소드 리턴 값이 사실은 int임.

그 값의 의미는 수정된 객체 값.

//원래는

번호 이름 나이

1 김 20

2 최 30

3 박 40

이거 첫 행에서 번호 받아서 넣어주고, 이름 받아서 넣어주고, 나이 받아서 넣어주고,

다음 행 가서 번호 받아서 넣어주고, 이름 받아서 넣어주고, 나이 받아서 넣어주고,

해야 해. person 객체에서 일일이 꺼내가지고 db에 넣어 줘야 하는데,

이 번거로운 과정을 mybatis는 한 마디로 해결해 줌. resulttype = “person”이라는 이름으로.

// for(Person p : list) {

System.out.println(p);

}

이거는 새로 생긴 기능인데, 처음부터 끝까지 프린트 해주는 기능.

for(int i = -; i < list.size(); i = i + 1) {

Person p = list.get(i);

System.out.println(p);

}

//DAO = Data Acess Object

//Manager mng1 = new Manager();

Manager mng2 = new Manager();

이렇게 만들어서

mng1.input(list);

mng2.output(list);

이런다고 input과 output의 실행 속도 자체가 빨라지는 건 아니잖아.

이것과 마찬가지로, 다음의 경우도 해당 됨.

SqlSessionFactory sf1 = MybatisConfig.getSqlSessionFactory();

SqlSessionFactory sf2 = MybatisConfig.getSqlSessionFactory();

SqlSession ss = null;

ss = sf1.openSession();

ss = sf2.openSession();

이렇게 두개 열고 쓴다고 해도 처리 속도 자체가 빨라지는 건 아님.

--각 부서별 최고 연봉자

select \* from employees as e1 where salary in (select max(salary) from employees group by department\_id) and department in

//join 테이블에서 where 구문 안의 and의 의미

select \* from employees e1 where salary in (select max(salary) from employees group by department\_id) and department\_id

//distinct에 다른 칼럼도 가져오면 어쩌냐

// 그룹함수 쓸 때 ‘짝이 안 맞기 때문에 에러 난다’의 뜻은?

서브쿼리 from select 이런 명령어는 언제 포함해 줘야 하고, 언제는 생략하고?

그룹함수 사용법. 어디에 넣을 것인가.

연산 순서 - 여러 가지 명령어들이 공존하고 있을 때, 연산 순서가 어떻게 되는지?

//dao 역할은?

//xml과 interface 관계

//왜 mapper까지 클래스 변수로 정의를 안 하고 각 메소드마다 계속 따로 만들어주지?

close땜에 그런가?

181210

//프로젝트 우클릭 – properties – 위에 탭 중에 library – add library 하든가

Edit에서 use default workspace

//SqlSession ss= null;

ss = ssf.openSession();

이거는 SqlSessionFactory의 멤버 메소드인 openSession을 실행해서, 열려 있는 세션

하나를 생성하여 넣어주지. 그럼 ss에는 이미 open된 session의 객체가 들어 있는 거라고.

그럼 그 객체를 통해서 close 해야지, close할 때도 factory 통해서 하면 이상하지.

ss = ssf.close();

이렇게 하면 SqlSessionFactory의 멤버 메소드인 closeSession을

다시 생성해서 다시 넣는 거잖아. 그러면 그냥 그동안의 작업하던 걸 그대로 버려둔 채로

다른 객체를 넣어두는 거지. 그 변수가 가리키는 주소만이 바뀔 뿐이지.

//xml에서 type 쓸 때는 같은 프로젝트 내에 있다고 하고, 패키지명부터 들어감.

181211

//여러 번 쓸 거면 VO클래스를 따로 만들겠다만, 한두 번 쓰고 말 거면, 그냥 map 에

담아서 return 해 줌.

//insert update delete 이런 tag들을 이용해서 mapper를 짜면, resultType을 따로

정의하지 않음. 이것들은 자동으로 int를 되돌려 주는데, 그 값의 의미는 몇 개의 자료가

등록/수정/삭제 되었는지. 값임.

//xml에서 sql문이 잘 짜여져 있는지? 자바와의 연동을 떠나서, sql의 구문 자체 똑바로

말하고 있는지?

+) return 이나 parameter의 type이 잘 쓰여져 있는지? 우리가 정의한 거라면,

오타는 없는지? “ 패키지.클래스 ” 해서 잘 쓰였는지? (프로젝트는 쓰지 않음.)

+) 인터페이스에서의 메소드 이름과, xml의 메소드 이름이 정확히 일치하는지?

+) 변수 이름을 맞게 해 놨는지? 오타는 없는지?

//받아오는 값이 하나뿐이라면 그냥 아무거나 적어도 얘가 parameter로 알아들어.

여러 개가 있으면 #{name} 하면 getName 으로 변환해 줌.

// ‘ 동적 SQL ’ 기능을 mybatis가 지원해 줌. 예를 들어 똑같은 ‘검색’의 기능이더라도,

‘제목에서 검색’, ‘본문에서 검색’, ‘작성자에서 검색’ 등등이 있잖아. 그렇다고

총 세 개의 구문과 쿼리를 짜서, if문으로 각각 실행하는 법이 가장 먼저 떠오르겠지.

하지만, mybatis에서는 어떤 값이 넘어오느냐에 따라 각각 다른 쿼리를 실행하도록

만들어 주는 기능을 제공해. 그것을 동적 SQL이라 함.

+) <if test "조건문"> ~~~ </if> 이런 식으로 조건 나열. if test까지가 한 문법.

+) 이것도 태그를 이용함.(mybatis에서 제공하는 기능들을 다들 태그로 감싸는 것들인 듯)

+) 거의 뭐, 자바의 문법과, SQL의 구문을 혼용한, 새로운 프로그래밍 언어를 만드는 듯.

//hashmap - HashMap< k , v > => key / value

아무거나 다 저장하려면, object 타입 하나면 됨.

+) hashmap.put("A", "B"); 이러면 "A"라는 이름을 붙여 놓은 "B"라는 애를 저장하겠다.

+) hashmap.get(Object key) 이러면 key라는 이름으로 저장해놓은 값을 불러온다.

+) key를 똑같이 저장해 놓으면, 저장은 되겠다만 그냥 앞에 녀석을 갖다 씀.

+) 이름은 항상 String으로 붙여도, 저장할 대상이 그때 그때 String이였다가,

int였다가, 하면, 담을 내용을 Object로 지정해놓으면 됨. 그럼 아무거나 다 담을

수 있음. 넣어 놨다가, 꺼낼 때 다시 그 자료형으로 뽑아 줌.

HashMape<String, Object>; 해놓고 put으로 하나씩 넣어 줌.

select \* from board where

<if test = " type == '1' ">

title like '%' || #{keyword} || '%'

</if>

<if test = " type == '2' ">

content like '%' || #{keyword} || '%'

</if>

<if test = " type == '3' ">

id like '%' || #{keyword} || '%'

</if>

이러다가

type 에 4가 들어가 있으면, 모든 if문이 다 그냥 통과하고

" select \* from board where " 이거 하나만 남음. sql 구문 에러임.

따라서, 이러한 default 상태를 '에러가 안 나는 상태'로 만들기 위한 여러 가지 궁리가 있음.

select \* from board where 1 = 1

<if test = " type == '1' ">

and title like '%' || #{keyword} || '%'

</if>

<if test = " type == '2' ">

and content like '%' || #{keyword} || '%'

</if>

이런 식으로, 앞에 항상 true인 조건을 넣어 놓은 후, 뒤에 따라오는 모든 if문의 시작에

and를 추가해 줌.

또는

select \* from board

<where> <if test = " type == '1' ">

title like '%' || #{keyword} || '%'

</if>

이런 식으로 where를 태그로 감싸기. 그러면 조건이 하나도 없을 경우,

where를 자동으로 삭제해 줌. 조건이 있으면 where 넣어 줌.

넣어주고 싶은 구문을 <where> ~~~ </where> where 태그로 감쌈.

mybatis에서 자동으로 감지해서 넣어줄지, 빼줄지, 판단함.

태그에 의한 다른 구문이 where 구문 앞에 추가될 경우, where를 넣어 줌.

게다가, 구문 시작할 때 들어 있는 and 나 or를 자동으로 삭제해 줌.

mybatis 한국어 가이드 - p.45

//VO클래스의 변수명이랑 같으면 뭐가 되는 거지?

//게스트 모드이면 생성되는 객체가 달라버리기

제공되는 기능이 다름.

// HashMap<String, Object> hm; 선언해 놓고,

hm.put("num", num); / hm.put("title", title); / hm.put("content", content);

이렇게 해놓고, int num = 5; / content = "cccc2222" 넣어 놓은 상태에서,

mapper 넣어 놓은 xml에다가

update board set #{column} = #{revise} where num = #{num}

이러면 오류 난다. 왜냐하면 #{revise} 자체가 'cccc2222'가 되는 것이므로

칼럼 이름을 인식하지 못하고 문자열로 인식. 저게 mybatis에 의해 번역되면 다음과 같음.

update board set 'content' = 'cccc2222' where num = '5'

원래 오라클 sql 쿼리에서 num = 5 해도 되고, num = '5' 해도 되고, 둘 다 작동함.

181214

//마이바티스에서 if구문 쓸 때 test = "" 큰따옴표 안에 있으면, 자바에서 변수명과 같다..

//if 구문 안에서 test = " ~~" 의 큰따옴표 안에 있는 이름은, 자동으로 변수명으로 생각.

따라서, parameterType으로 받아온 객체로부터, 해당 변수명을 가지는 멤버변수를

getter 이용해서 뽑아 옴. 예를 들어, HashMap hm을 받아 왔으면,

<if test = "hello == 1"> </if> 이렇게 하면, hm.getHello(); 를 실행해서

큰 따옴표 안의 'hello' 대신에 넣어 줌. 결국 실행되는 구문은,

<if test = " hm.getHello(); == 1"> </if> 이렇게 실행되지.

+)그러니까 만약에 HashMap이나, 아니면 우리가 직접 만든 자료형이나, 등등처럼

get변수명() 클래스가 있는 애들이면, 바로 쓰면 되고. 하나만 보낼 경우에는,

\_parameter로 써줘야, getter가 실행되지 않고, 넘어온 패러미터 변수를 그대로 사용함.

//내가 1번을 선택해서 1번만 실행하고 싶어도,

1번을 선택하면 그 변수 choice가 1인지, 2인지, 3인지, 4인지, 일일이 다

if구문 통해서 확인해야 하니까, if구문의 조건문 <if test = " ~~~ "> </if> 이 구문은

모든 구문을 지나가야만하지. 근데 중간에 뭔가 오류 뜨는 게 있으니까 더 이상

못지나가는 거지. 오류 내버리는 거지.

//내가 실수로 테이블 만들 때 hi 칼럼을 char로 선언해 놨어

근데 where로 hi 가져와서 hi값이 3인 로우를 가져오고 싶어. where hi = to\_char('3')

이러면 못 찾아 와. 왜냐면 hi가 char로 선언할 때 4byte로 크기를 선언해 놨으면,

3만들어 놓고, 4byte에서 남는 용량은 다 공백으로 처리해 놔.

3 \_ \_ \_ 이렇게 해놔. 따라서 hi like '%' || to\_char('3') || '%'