181203

//아이디, 글 번호, 조회수, 등등은 그에 맞는 자료형 지정해서 칼럼에 넣으면 되는데,

리플의 경우는 어쩔래. 칼럼을 몇 개를 준비해 놔야, 글 하나에 리플을 충분히 대비할 수

있을까. 리플이 없을 수도 있고, 몇 백개, 몇 천개가 될 수 있는데.

이럴 때는 다른 테이블을 따로 만들어서 리플을 넣어 놓고, 그 테이블이 다른 어떤 테이블의

어떤 칼럼의 어떤 글에 담겨 있는 테이블인지를 명확히 지정해 주면 됨. 이걸 외래키라고 함.

//한글 기본 3바이트

//회원 탈퇴할 때, 글 내용 다 그냥 저절로 삭제 돼? 작성했던 글과 리플과 함께 회원 정보 모두 삭제? 그럼 DB가 깔끔하긴 하곘지만, 기록이 안 남지. 문제가 생길 수도 있지. 왜냐면

만약 사기치고 탈퇴하면? 깔끔하게 기록이 다 지워지잖아? 캐쉬 충전 기록 같은 거는 어쩔?

그래서 탈퇴하는 회원 정보를 어쩔지, 뭐 다른 테이블로 기록을 옮겨 놓든지, 깔끔히 삭제하든지, 그대로 놓든지, 코딩 상으로는 그냥 그게 그거. 다만 회사 정책일 뿐.

//게시판 글 목록을 [번호, id, 제목, 날짜] 형식으로 출력

select num, id, [select name from userinfo where userinfo.id = board.id) name, title from

//각 행을 실행할 때마다 서브 쿼리를 실행. 100개 행을 만들 거면, 서브 쿼리 100번 실행. 각 로우마다.

select num, title, ( select count(num) from reply where reply.num = board.num) from board;

이러면 board 에서 num 칼럼 값이 1인 행을 만들 때, 괄호 안의 서브쿼리가 한 번 실행 되고,

board에서 num 칼럼 값이 2인 로우를 만들 때, 괄호 안의 서브 쿼리가 또 한 번 실행 되고.... 반복.

//앞에서 내가 주겠다고 했으면, 디폴트 값이 있어도 무시.

insert into reply(num, bnum, id, text, inputdate) values (replyseq.nextval, 2, 'ididid', '리플리플플리플리', '');

이러면, table reply를 create 하면서 inputdate 칼럼의 default를 sysdate로 줬어도,

앞에서 내가 insert into reply(~~~ ,nputdate)라고, 값 줄 거라고 미리 말해 놨으니까,

뭐 받으려고 준비한다고. 그럼 ‘ ’ 이렇게 공백 따옴표 넣어도, default 안 들어가고

공백, null 들어 간다고.

//앞에서 준다고 해놓고 뒤에서 안 주면, not enough values

앞에서 준다는 말 없었는데 뒤에서 갑자기 주면, too many values

//자바랑 똑같아. 특정 테이블의 특정 칼럼 값 가져오려면. 테이블명.칼럼명.

자바도 클래스이름.메소드이름 하듯이. 큰 거 먼저, 작은 거.

//select 칼럼들 해서

--'aaa'라는 사람이 쓴 글 중에 조회수가 가장 높은 글보다, 조회수가 높은 글들.

select id,title, hits from board where hits <= (select hits from (select hits from board where id = 'aaa' order by hits) where rownum = 1 );

/'aaa'라는 아이디를 가진 사람이 쓴 글을 hits 순으로 나열한 표를 만들고,

그 표에서 rownum이 1인 행의 hits 값을 가져온 후에,

그 값보다 hits 값이 큰 행의 id, title, hits 를 board에서 가져오고 싶다.

//rownum 사용법

그냥 max;;;

--전체 평균보다 조회수가 10 이상 작은 글 목록

select id, title, hits from board where hits < (select avg(hits) from board) + 10;

//avg 쓰는 법

--위의 글을 쓴 사람들 명단

select distinct id from board where hits > ( avg(hits) );

--회원별 게시글 작성 개수와 리플 개수

select id, count(id), ( select count(id) from reply where board.id = reply.id ) from board group by id;

select userid, ( select count(userid) from board group by userid) from board;

--처음

select id, ( select count(id) from board where ) from board;

--중간

select id, (select name from userinfo where board.id = userinfo.id), count(id) as count\_board, ( select count(id) from reply where board.id = reply.id ) as count\_repply from board group by id;

--끝

select id,

(select name from userinfo where board.id = userinfo.userid) as name,

count(id) as count\_board,

( select count(id) from reply where board.id = reply.id ) as count\_reply

from board group by id;

--이름도 같이 뽑기

select id

, (select name from userinfo where board.id = userinfo.userid) as name

, count(id) as count\_board

, ( select count(id) from reply where board.id = reply.id ) as count\_reply

from board group by id;

select name from userinfo where userinfo.userid in (select id from board);

select name from userinfo where board.id = userinfo.userid;

위는 되고 아래는 안 되고

반대로 위는 옮기면 안 되고, 아래는 옮기면 되고

//select \* from board where hits > (select avg(hits) from board);

//select id from ( select \* from board where hits > (select avg(hits) from board) );

//select name from userinfo where userinfo.id in

(select id from board where hits > (select avg(hits) from board);

//select id , (select name from userinfo where board.id = userinfo.id) , count(id)

from reply where (reply.id = board.id)

이해 안 되는 거 모음

//select count(bnum) from reply where reply.bnum in (select num from board)

group by id;

--id 와 bnum 관계

항목 수입 지출

//select id, num, ( select count(bnum) from reply where board.num = bnum )

from board;

--이거는 서브 쿼리 안에 맘대로 암거나 넣어도 되나

다른 서브 쿼리는, group by에 해당하는 녀석들이 서브 쿼리 안에 등장해야 하잖아.

행 개수가 맞춰 지려면, 바깥쪽 select 칼럼이 서브쿼리 안에 있어야지. 서브쿼리 자체가 반환하는 행의 개수는,서브 쿼리 안의 where에 있는 변수의 개수 만큼 이니까.

//select id, ( select count(bnum) from reply where board.id = reply.id )

from board group by id;

// select count(hits) from board group by count(hits) desc;

--노수연 창조경제

181204

// order는 항상 맨 뒤에 붙임. 우선 순위에 따라 여러개 하고 싶으면

쉼표로 구분.

우선 id로 구분하고, id 중에서도 글 번호로 구분하고 싶으면

select \* from board order by id, num ;

만약 id는 오름차순, num은 내림차순이면 그 각 칼럼 옆에 마다 정렬 방식 추가해 줌

select \* from board order by id asc, num desc ;

select num, id, to\_char(inputdate, ‘yyyy’ || ‘년’ || 'mm' || ‘월’ || dd || ‘일’) from board; --이건 안 되나?

행에 변환 주는 것들 - > order 명령어, where 명령어, group by 명령어,

그룹함수 - group by 랑 같이 씀.

그룹함수 ( count, sum, acg, max, mix )

count(칼럼) sum(칼럼) 보통 행 단위 갯수 세려면 \* 쓰기도 함. null도 있을 수 있으니

group by는 단독으로 쓰이지 않고 항상 group 함수랑 같이 씀.

group by 에다가 where 까지 넣어 주면, sum(합계를)을 구하되, count(갯수를 세)를 구하되, group으로 나눠서 알려 줘라.

select count(\*) from board where id like 'aaa';

select count(\*) from board where id like 'bbb';

select count(\*) from board where id like 'bbb';  
--이거를 축약한 게 group by.  
select count(\*) from board group by id;

--distinct 효과도 있잖아. 근데 같이 쓰면 어째?

id, title, num, sum(hits) group by id 이러면?

안 돼. group by 의 대상만을 앞에 select할 수 있음.

group by id 했으면 앞에도 select id 또는 select id, sum(hits)

group by num 했으면 앞에도 select num 또는 select num, sum(hits)

등등, group 함수 or group 묶어주는 칼럼만 select 할 수 있음.

우연히 맞으면 어쩔래 -> 같은 행에 있는 애들이 서로 연관이 없을 수 있는가

예를 들어 select id, name, count(num) group by id

or select id, distinct name, count(num) group by id

select count(\*) from board where to\_char(inputdate, 'yyyy') = '2018';

2018년에 쓴 글이지.

select count(\*) from board where to\_char(inputdate, 'yyyy') = '2018';

select count(\*) from board where to\_char(inputdate, 'yyyy') = '2017';

select count(\*) from board where to\_char(inputdate, 'yyyy') = '2016';

select count(\*) from board where to\_char(inputdate, 'yyyy') = '2015';

select count(\*) from board group by to\_char(inputdate, 'yyyy');

--연도별 글 개수 통계

group by 대상에도 내가 원하는 조건 직접 만들어 낼 수도. 기존의 칼럼만을 사용할 수밖에 없는 것이 아니라.

select avg(hits) from board where to\_char(inputdate, 'yyyy') = 2018 group by to\_char(inputdate, 'mm');

2018년도의 월별 조회수 평균

//연산 순서

select to\_char(inputdate, 'yyyy'), count(\*) from board

group by to\_char(inputdate, 'yyyy')

order by to\_char(inputdate, 'yyyy') ;

우선 where 다음 group by 다음 그룹 함수 다음 정렬

//라벨링

select to\_char(inputdate, 'yyyy') as d, --이름 붙인다.

count(\*) from board

group by to\_char(inputdate, 'yyyy')

order by d ; --이름 쓴다. 연산 순서에 맞게, 그 앞에 만들어진 상태여야 가능.

대신 라벨링 했으면, 뒤에서 다시 부를 땐 꼭 써줘야 해. 그 문장에서는

해당 from ‘테이블명’ 의 ‘테이블명’이 없는 거나 마찬가지. 그대로 찾으면 못 찾아.

from '테이블명‘ as tn 이렇게 해놨으면, 그 문장에서는 테이블명.num 이러면

못 찾아. tn.num 해야 돼.

//검색할 때는 대소문자 구별을 안 하고 출력해줄 수 있도록.

그러려면 검색어와 검색 대상을 모두 대문자로 바꾸든가,

둘 모두를 소문자로 바꾸든가.

upper(last\_name) like upper(검색어); 이런 식으로.

아래처럼

select \* from employees where last\_name like 'A%' or last\_name like 'a%';

이렇게 하면 지금 검색어가 ‘ a ’ 하나니까 경우의 수가 두 가지인 거지,

hello big world 였어봐. 2^13 개 많아. upper 써야지.

// join 할 때는,

2개의 테이블 join 할 때는 최소 1개의 조건이 있어야 하고,

3개의 테이블 join 할 때는 최소 2개의 조건이 있어야,

join하는 테이블이 의미있는 테이블이 됨.

select \* from board, userinfo where userinfo.id = board.id;

join 하고자 하는 두 테이블에 각각 공통적으로 존재하는 요소를, 조건으로 놓고.

+ 테이블 이름에도 라벨링 해서 짧게 부를 수 있음

select num, m.id, n.title from userinfo as m, board as n, --등등..

+두 테이블 합치면 칼럼 이름이 그냥 붙어버려. 한 쪽 이름으로만 가져오기

//새 발 같은 거는, 가지 하나 있는 쪽에는 하나씩만 있지만, 그것이

가지 세 개 있는 쪽에서 여러 번 참조 돼 있다.

+서로 참조도 있어. 각 부서에는 팀장, 매니저, 직원 등이 있겠지. 이거는

전체 직원 목록에도 있는 거야. 근데, 전체 직원 목록에는, 자기 부서가 등록돼 있어.

‘부서’ 테이블일아 ‘직원 목록’이랑 서로 참조.

+복합키 - 두 칼럼을 조합해서 서로 다른 행을 만듦. 프라이머리 키가 두 개라면,

두 프라이머리 키가 동시에 같을 수는 없지.

똑같은 사람이 다른 곳으로 여러 번 옮길 수는 있지만,

똑같은 사람이 같은 곳으로 여러 번 옮길 수는 없지.

‘사람’과 ‘부서’를 두 프라이머리 키로 지정해서, 조합함.

똑같은 사람이 서로 다른 글에 여러 번 좋아요를 누를 수는 있지만

똑같은 사람이 같은 글에 여러 번 좋아요를 누를 수는 없음.

‘좋아요’ 테이블을 안에서,

‘사람’이라는 프라이머리 키만 놓고 보면 여러 번 등장할 수 있지.

하지만 ‘사람’과 ‘글 번호’라는 두 프라이머리 키가 동시에 여러 번 등장할 수는 없지.

//셀프 조인? 테이블은 하난데, 특정 값에 대해 참고할 대상이 자기 자신 테이블이야.

대신 라벨링 했으면, 뒤에서 다시 부를 땐 꼭 써줘야 해. 그 문장에서는

해당 from ‘테이블명’ 의 ‘테이블명’이 없는 거나 마찬가지. 그대로 찾으면 못 찾아.

from '테이블명‘ as tn 이렇게 해놨으면, 그 문장에서는 테이블명.num 이러면

못 찾아. tn.num 해야 돼.

181205

//문자열 비교할 때, ‘ = ’ 과 ‘ like ’ 명령문 차이.

‘ = ’ 은 원래 숫자 하는 애고, ‘ like ’ 는 원래 문자열 비교하는 애.

다만, 문자열이라도, 완전히 똑같은 경우는 ‘ = ’으로 비교하는 경우도 허용.

따라서 완전히 똑같은 문자열은 ‘ = ’ 와 ‘ like ’는 둘 다 사용할 수 있지만

와일드 카드 쓰는 경우, ‘ %work ’ 라든가 ‘ pro\_ ’ 이런 거는 ‘ like ’만 사용 가능.

//select 요소에는 as 를 붙이지만, 테이블 이름에는 as를 붙이면 에러.

//group by id 며는 앞에 select 에서 id와 그룹함수만 가능한데,

그 안에 서브 쿼리 넣을 때, where 에 메인 쿼리의 id가 들어가 있으면 되는 거 아녀?

응 아녀, 예를 들어 userinfo 랑, board 있으면 어쩔래.

//얼개를 파악하는 데에는, 새로운 문제 푸는 것 보다, 복습이 좀 더 도움이 되는 듯.

//select department\_name , count(\*) from employees e, departments d

where e.department\_id = d.department\_id group by department\_name, job\_id;

--이 구문은, 원래 employees 테이블만 가지고는, department\_name 으로 칼럼을 못

가져오지만, 가져오기 위해서는 서브 쿼리를 써야 하지만,

조인을 해 놓으면, 그냥 가져다 쓸 수 있어.

+ group도 하나만 엮는 게 아니라, 상하위 나눠서, 우선 A기준으로 그룹한 후에,

그렇게 나눠진 한 그룹 안에서 다시 B를 기준으로 그룹을 나누고, 하는 등으로

여러 층에 걸쳐서 그룹을 나눌 수 있음. group by A, B;

+order도 마찬가지. A기준으로 정렬한 후에, A가 같은 행들끼리는 B를 기준으로 정렬하고,

order by A, B;

//from 테이블1, 테이블2 where ~~~~~~~~ 할 때,

from 테이블1, 테이블2 만으로 생성되는 행들과,

그 중에서 where로 인해서 걸러지는 행들. 출력되는 행들.

//

--부서별 급여 평균 (급여가 3000 이하인 직원 제외하고 계산 후, 평균 급여가 5000 이상인 부서만 출력)

select department\_id, avg(salary) as i where salary > 3000 group by department\_id having i >= 5000;

--이게 내 꺼

select department\_id, avg(salary) from employees where salary > 3000 group by

department\_id having avg(salary) > 5000 order by department\_id;

--이게 쌤 꺼

//group by, having 등에서는, 테이블 라벨링은 되지만 셀렉트 라벨링는 안 된다

셀렉트에 라벨링은 그냥 출력할 때 칼럼 이름을 보기 좋게 하려고 이름 붙이는 거.

그 구문 안에서 호출하는 용도로 라벨링 하는 게 아님.

테이블에 라벨링은 그 구문 안의 다른 곳에서 호출할 때 간단하게 호출하려고 이름 붙이는 거.

따라서 order by 또는 group by에서는 select에서 붙여 놓은 라벨링을 호출할 수는 없다.

+) select department\_id, avg(salary) as i from employees e1 where salary > 3000 group by department\_id having i >= 5000;

--이런 경우는 ‘ i ’가, select 안의 칼럼에다가 붙여 놓은 이름, 라벨링이기 때문에,

뒤에서 group by에서는 호출할 수가 없지.

하지만 ‘ e1 ’은 테이블에 붙인 거니까, group by 또는 having 등에서는

having e1.salary > 5000 또는 group by e1.employee\_ID 등이 가능하지.

//decode (원본, 조건이 되는 경우의 값1, ‘출력할 말1’, 조건이 되는 값2, ‘출력할 말2’, .... ‘그 외 디폴트값’)

decode(commission, 10, ‘영업부’, 20, ‘사업부’, 30, ‘회계부’ , ‘미배정’)

// A in (b,c,d,e...) A가 ( )안에 있을 때. A가 (b,c,d,e...) 의 괄호 안에 있을 때. in.

//--[hr 계정 연습]

--전체 직원 목록

select \* from employees;

--60번 부서의 직원들

select \* from employees where department\_id = 60;

--성이 A로 시작하는 직원

select \* from employees where upper(last\_name) like upper('A%');

--사번이 100~200번인 직원

select \* from employees where employee\_id between 100 and 200;

--입사 년도가 2006년인 직원

select \* from employees where to\_char(hire\_date, 'yyyy') = '2006';

--2007년 입사 인원

select count(\*) from employees where to\_char(hire\_date, 'yyyy') = '2007';

--급여가 7000 넘는 직원

select \* from employees where salary > 7000;

--전체 평균 급여

select avg(salary) from employees;

--부서별 평균 급여

select department\_id, avg(salary) from employees group by department\_id;

--전체 평균 급여보다 많이 받는 직원

select \* from employees where salary > (select avg(salary) from employees);

--각 부서별 최고 연봉자

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

--위에꺼 다시

--급여가 10000 이상인 직원의 이름과 부서, 그 부서의 평균 급여

select first\_name , department\_id , round( (select avg(salary) from employees e1 where e1.department\_id = e2.department\_id) , 1 ) from employees e2 where salary > 10000;

--위에 꺼 다시

select first\_name

, last\_name

, (select department\_name from departments

where employees.dempartment\_id = departments.department\_id)

, (select avg(salary) from employees e2 where e2.department\_id = e1.department\_id)

from employees e1

where department\_id

in (select department\_id from employees where salary > 10000);

--사원 수가 3명 이상인 부서의 평균 급여를 소수점 아래 한 자리까지 출력 (부서 명도 함께)

select department\_id count(\*), avg(salary) from employees

where salary > 3000 group by department\_id having count(\*) > 3 order by

--부서별 급여 평균 (급여가 3000 이하인 직원 제외하고 계산 후, 평균 급여가 5000 이상인 부서만 출력)

select department\_id, avg(salary) as i where salary > 3000 group by department\_id having i >= 5000;

--이게 내 꺼

select department\_id, avg(salary) from employees where salary > 3000 group by

department\_id having avg(salary) > 5000 order by department\_id;

--이게 쌤 꺼

--급여가 10000 이상인 직원의 이름과 부서, 그 부서의 평균 급여

select first\_name , department\_id , round( (select avg(salary) from employees e1 where e1.department\_id = e2.department\_id) , 1 ) from employees e2 where salary > 10000;

--직원 수가 1 또는 3 또는 5명인 부서의 이름을 정렬하여 출력

select department\_name from departments d, employees e

where d.department\_id = e.department\_id and

(select count(department\_id) from employees e2

where e2.department\_id = e.department\_id) in (1, 3, 5);

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?