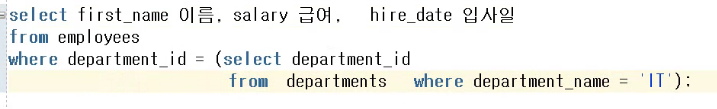
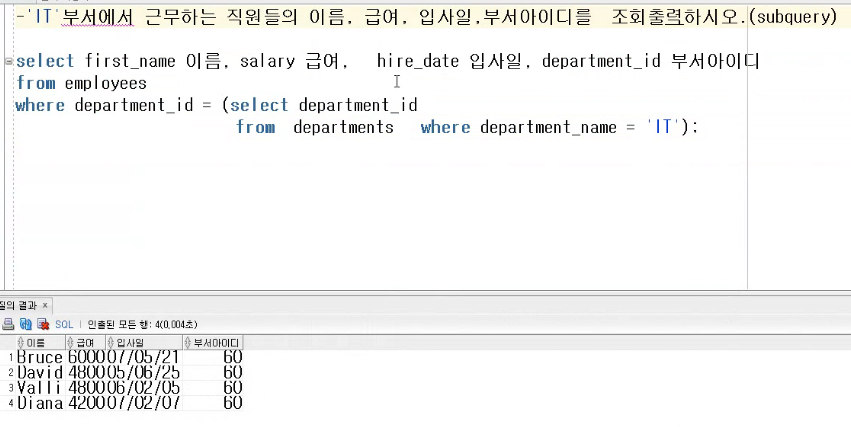
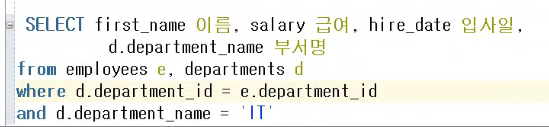
**0614**





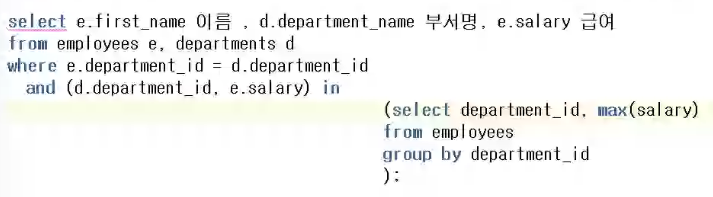


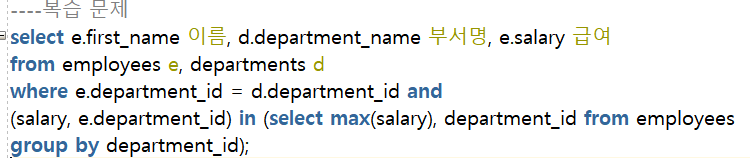




Join과 subquery는 거의 유사하다.





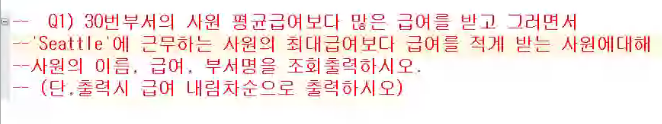


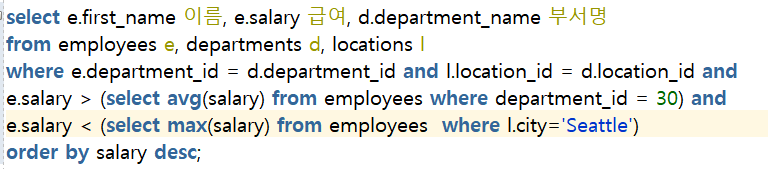
부서명은 employees가 아닌 departments에 있어서 department\_id로 조인을 해줘야 하기 때문에 e, d의 alias를 쓰고 where 조건문에서 조인식을 먼저 작성해 준다.

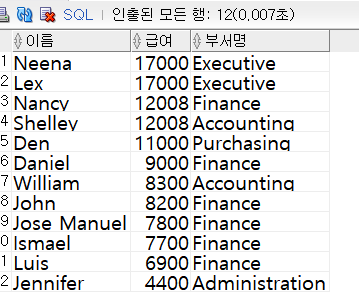
그룹별로 최대 급여를 출력하는 것이다. 최대 급여와 부서명을 엮고 싶었으나, employees 한 테이블에 있는 두 컬럼 값을 묶어서 계산하는 것이 용이하다고 판단하였다.

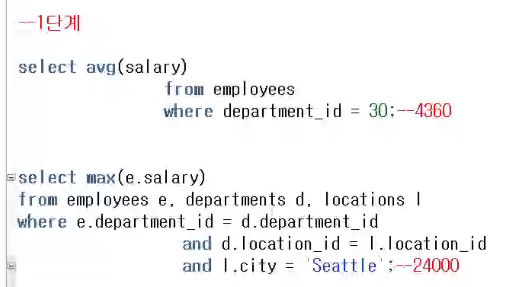
즉 급여와 부서 아이디를 묶고 그것이 부서 아이디별 최대 급여와 부서 아이디와 매칭되어 계산된 결과 값들에 있는지를 in으로 묻는다. In을 사용한 이유는 부서별 최대 급여를 받는 직원이 여러 명일 수 있기 때문에 사용한 것이다.

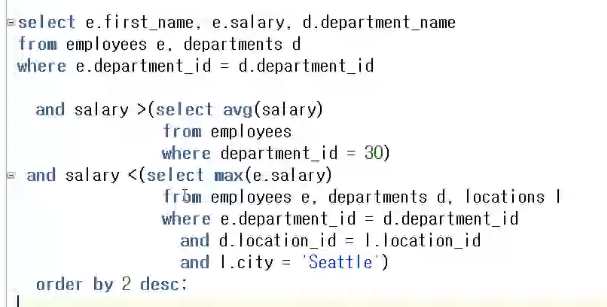
괄호로 묶은 두 칼럼, 부서별 그리고 최대 급여에 매칭 되도록 in 뒤의 select 문에서 두 컬럼 값을 순서에 맞춰서 써야 한다. 그리고 앞의 employees e 와 where 문 뒤의 employees는 다르다고 보면 된다. 따라서 in 뒤의 칼럼에 e를 쓰지 않도록 하자. 그리고 group by는 where 문 안에서 최대 값을 계산하는 select에 적용되는 것이 때문에 in 뒤의 괄호 select 안에 써야 한다.

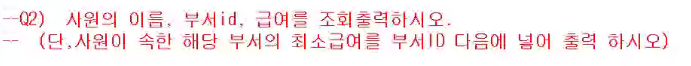


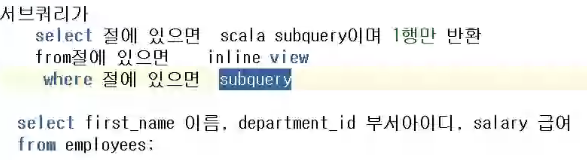


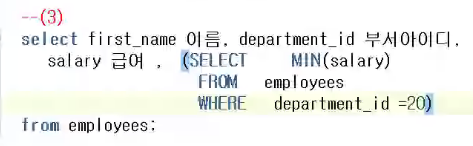


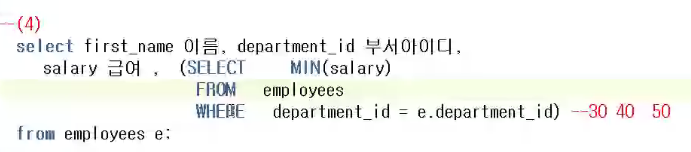


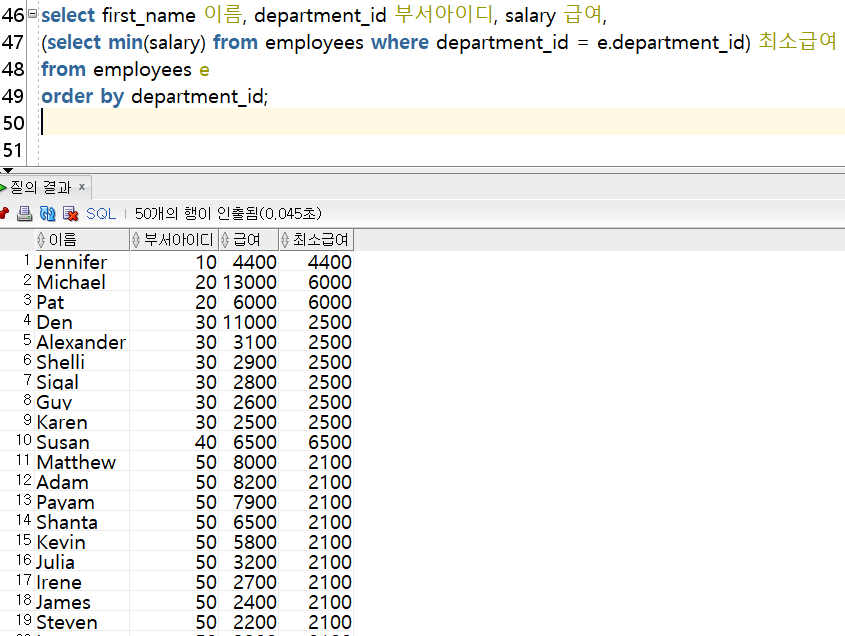


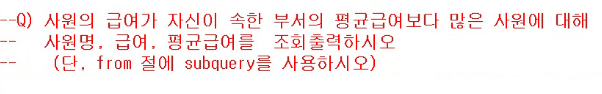


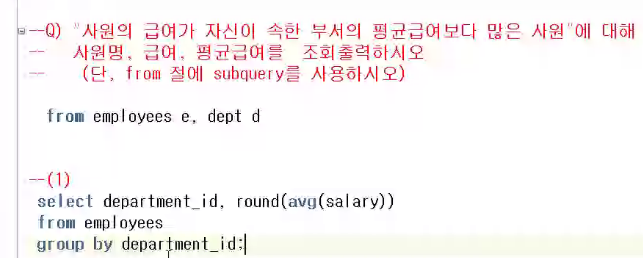






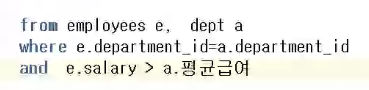




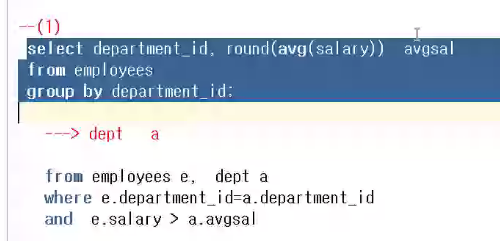


1. 전체를 테이블로 본다.

Dept a 라고 별명을 지었다.

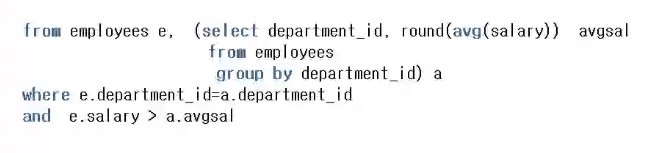


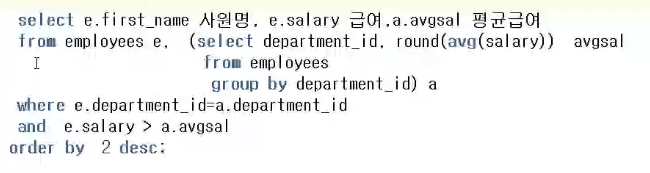
같은 컬럼끼리 join 해준다. 그리고 조건을 넣어준다.

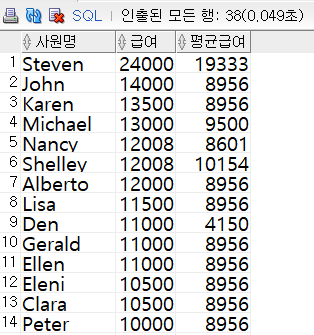


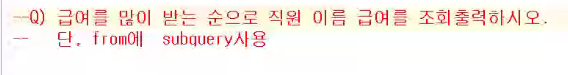
평균급여의 별명으로 바꿔준다.

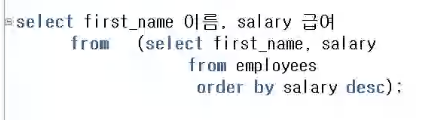
먼저 계산되는 subquery는 괄호로 묶어준다.



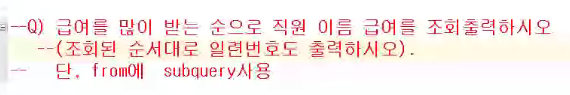




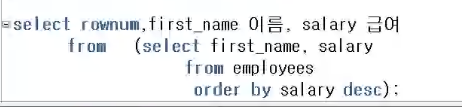




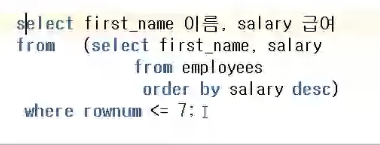




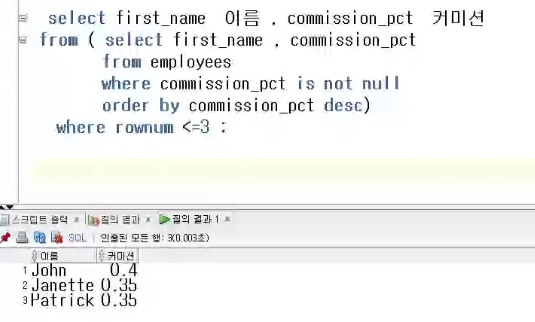




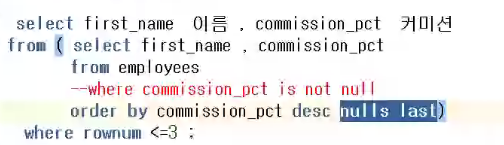




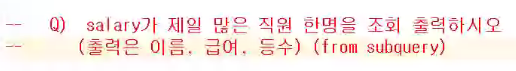


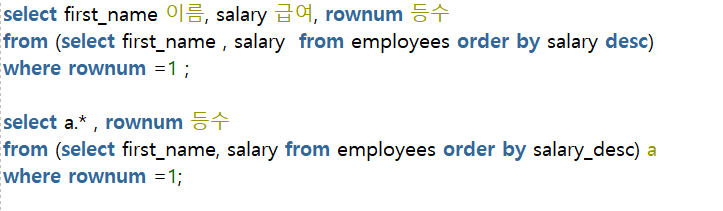


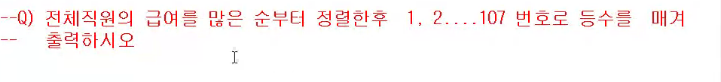
Null을 가장 큰 값으로 인지하기 때문에, null을 제외하는 조건을 넣어주어야 한다.

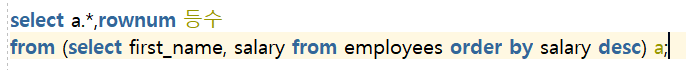


Null을 제일 밑으로 보낸다.

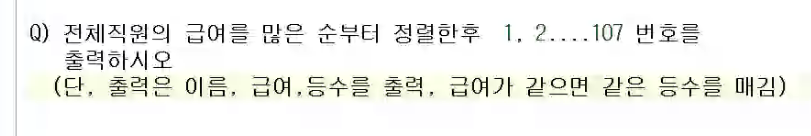




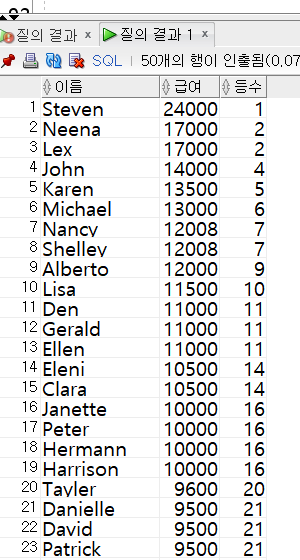


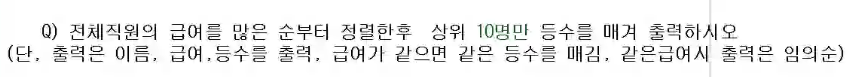






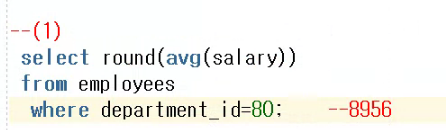


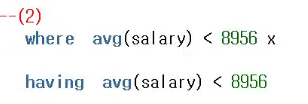


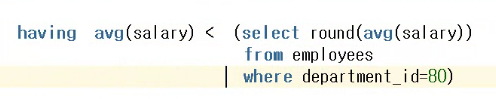




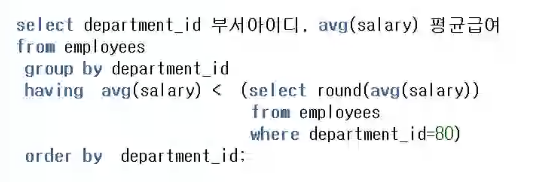


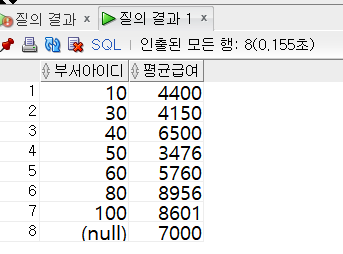






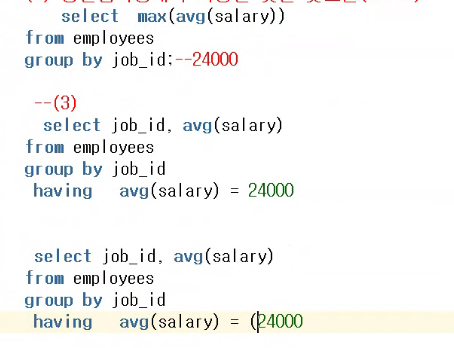


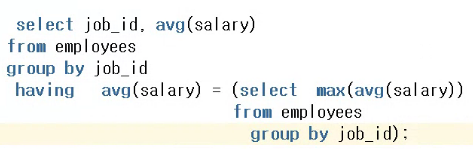


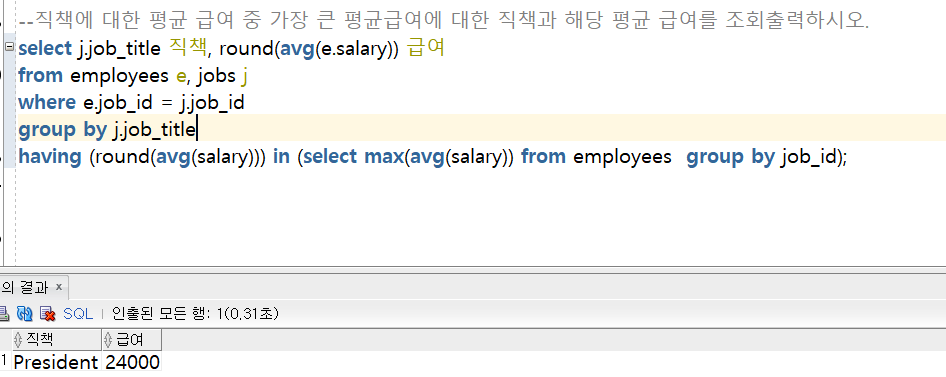




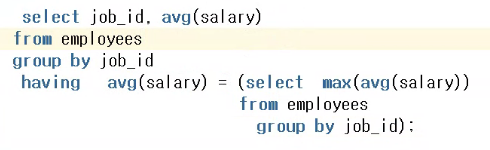






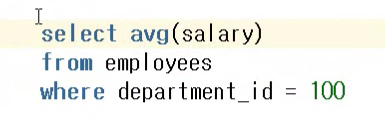


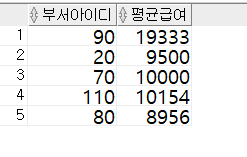
Group by 하기 전에 where로 조인을 시켜준다.

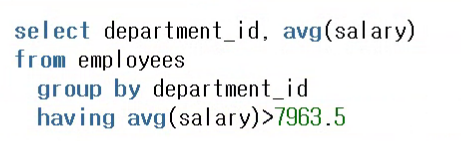


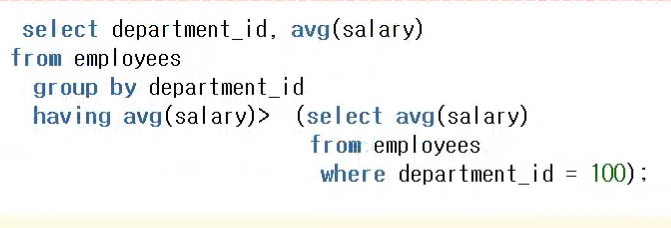


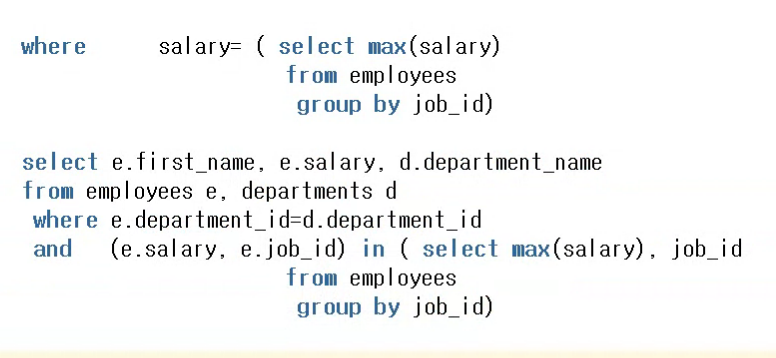






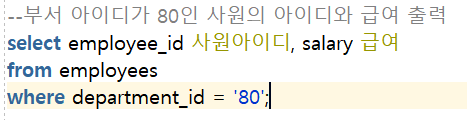


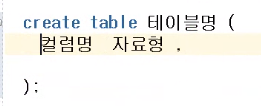




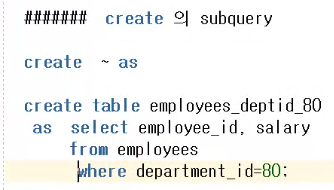
**0615**

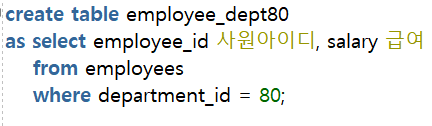
****

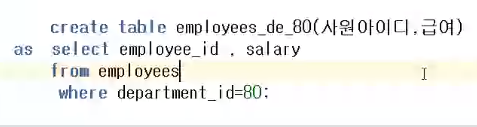
****

****

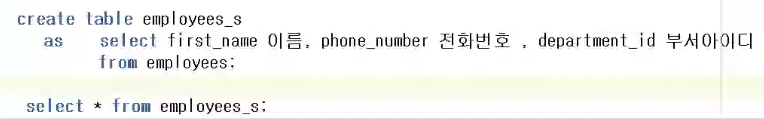
**Create의** subquery. 테이블을 만들기 전에 34명의 사원을 조회하는 식을 작성한 후 넣기 때문에.

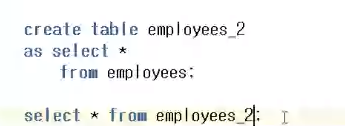




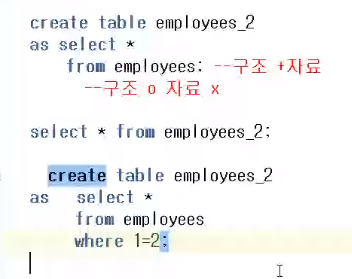


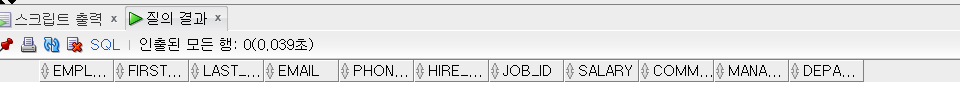
비상연락망



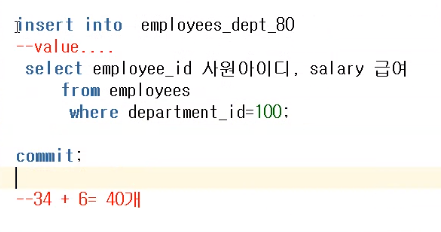
****

**구조만 주고** 자료는 없앤다.

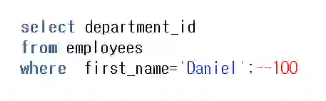
일부러 틀린 조건을 주어서 데이터가 들어가지 않게 한다. 조건이 맞지 않으면서 테이블이 생성되면 only 구조만 들어가게 된다.

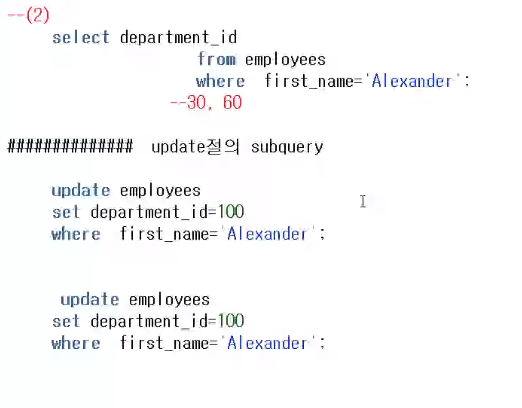


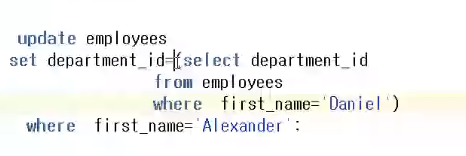




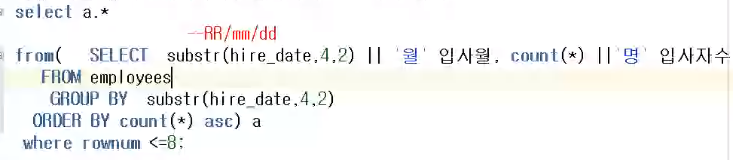


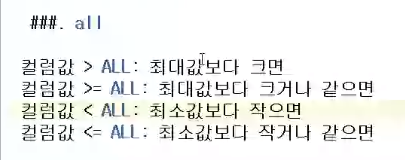


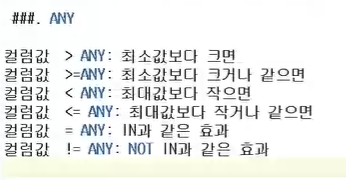




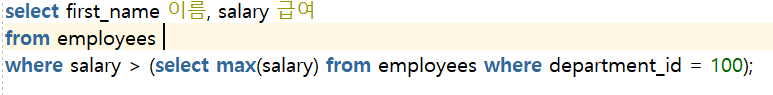




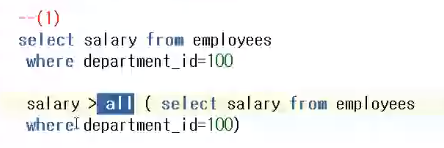


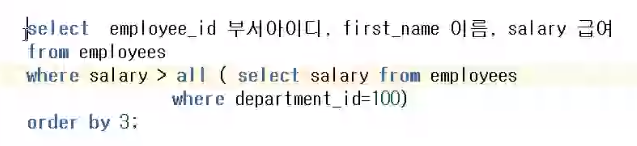


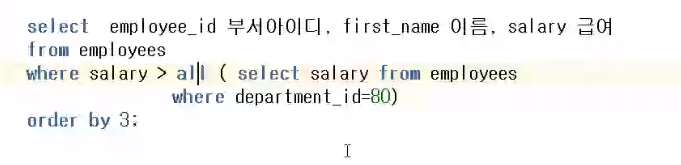


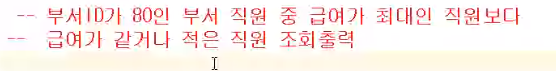


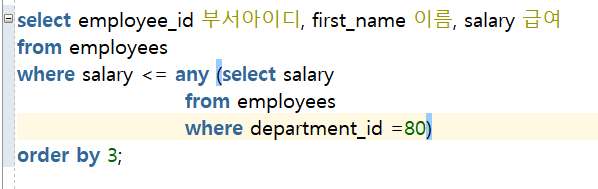
ALL을 사용해보자.

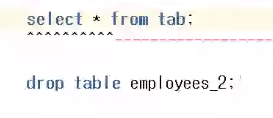












 휴지통 비우기

테스트 대비

Is null, nvl, to\_char. 커미션이 있는가 없는가, 출력할 때 부서가 없으면 어떤 문장을 출력하고 싶을 때. Select 문장 6가지 절

Order by 마지막.

간단한 문제인데, 복합적으로 연결되어 있는 문제 풀어보기.

Salary between and .

Is null, is not null. 있나 없나.

Group by.

결과와 문제.

입사년도 별 사원 수 . group by, count.

국가별

Employees, department, lations, countries . join. Count- 직원수

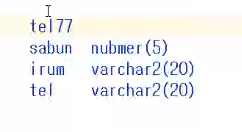
매니저, ger , 포함된 글자 like

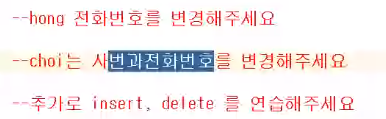
킴벌리 -> null -> outer join.

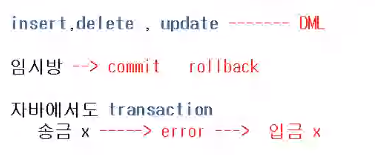
3~4문제.

6단계. Select from where group by, having , order by

테이블 만들기



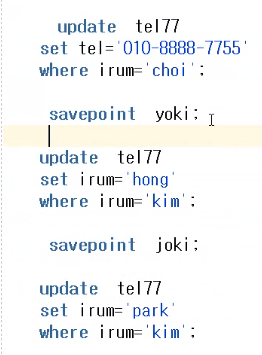




Update, insert, delete DML. Rollback을 하면 모두 하지 않았던 것이 된다.

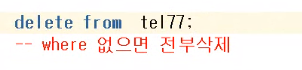
Commit을 하면 앞의 명령문이 싹 다 유에 저장된다.

Rollback의 범위는? Rollback도 앞에 있는 것 모두 적용된다.



Savepoint를 사용하자.

 yoki까지만 rollback 처리 그 이후에는 아직 임시방에 있음.



단 rollback;을 하면 지운 것이 돌아온다.



Desc tel77; 하면 구조는 남아있을 것을 볼 수 있다.



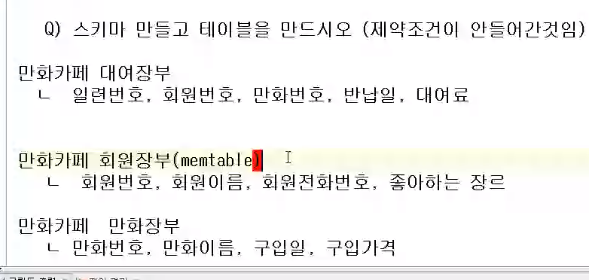
Drop 한 것을 되살린다.

purge는 완전 삭제. 되돌릴 수 가 없다.



참조 원본을 삭제할 경우 cascade constraints를 쓴다.

From -> where 조건 -> 그 외 문제에서 주어진 조건 -> select 마지막에.



스키마

**대여장부**

이름 번호 회원번호 만화번호 반납일 대여료

컬럼명 nums m\_num man\_num turn\_in\_date rental\_ fee

타입 number varchar2 varchar2 date number

길이 5 20 20 10

옵션 default sysdate

**회원장부**

이름 회원번호 회원이름 회원전화번호 좋아하는 장르

컬럼명 m\_num m\_name phone genre

타입 varchar2 varchar2 varchar2 varchar2

길이 20 15 15 15

**만화장부**

이름 만화번호 만화이름 구입일 구입가격

컬럼명 man\_num man\_name in\_date price

타입 varchar2 varchar2 date number

길이 20 15 20 10

옵션 default sysdate