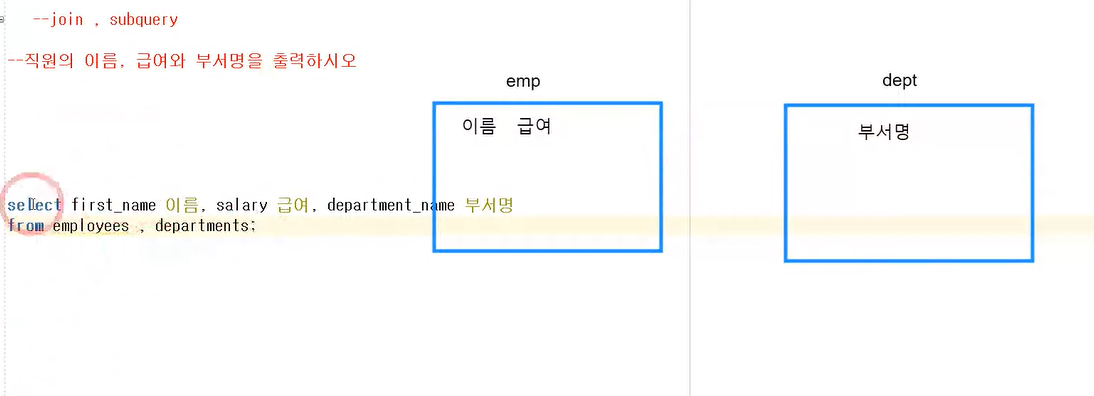
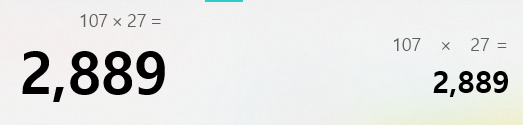


해당 파일은 desktop -컴퓨터 -자바웹개발- test.sql로 만들었음.



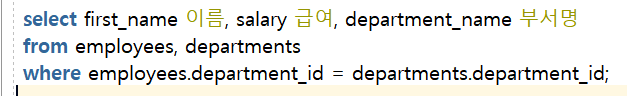
한 직원에 모든 부서가 연결되었다. 따라서 107\*27개의 행이 출력된다.

두 개의 테이블이 cartesian product로 연결되어 있다.

**Join**

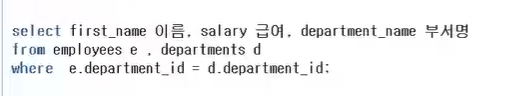
테이블이 2개 이상이고,

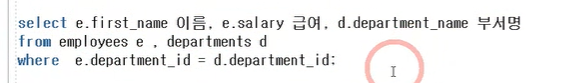
Pk, Fk로 연결되어 있다.



PK와 FK가 같은지 조건을 넣어서 join해주고, 앞에 **테이블.키**명 으로 입력한다.

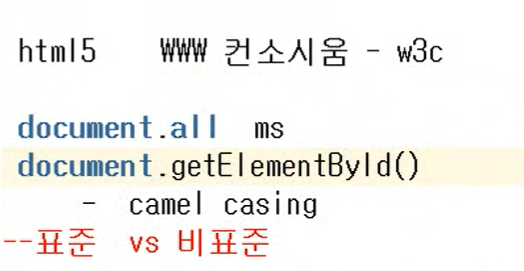
테이블의 별명

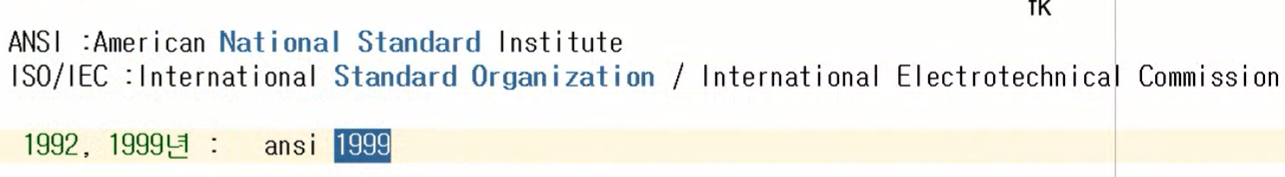




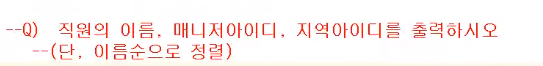


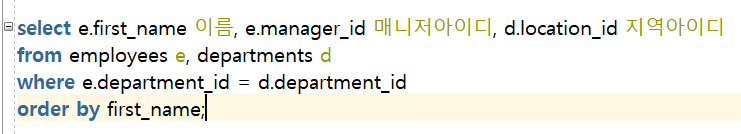
같은 것을 기준으로 묶는다.



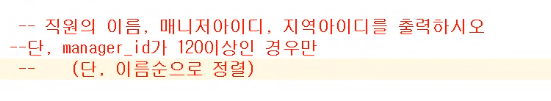
SQL의 표준 이름

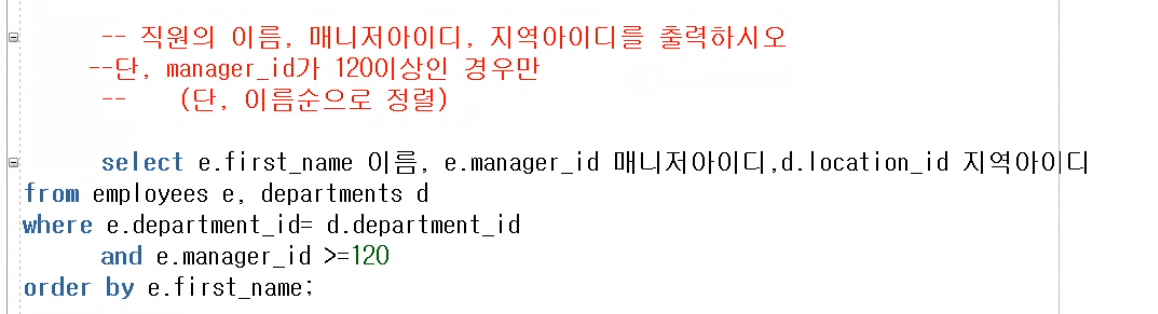
Join : oracle only -4개













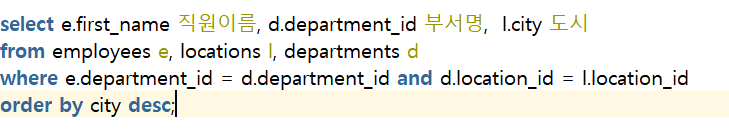


연결고리…!

**0610**

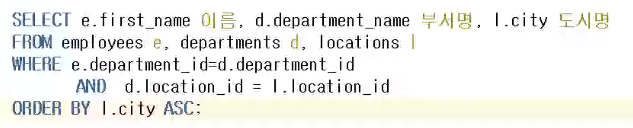


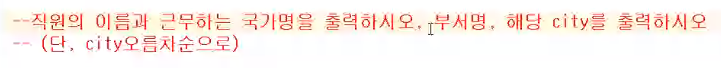


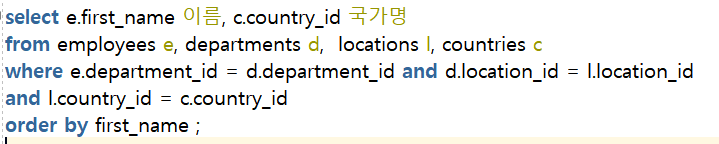


테이블 3개. 서로 연결되어야 한다.

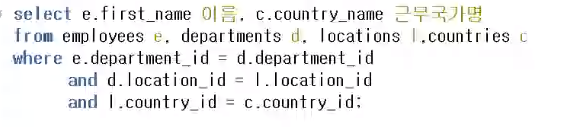




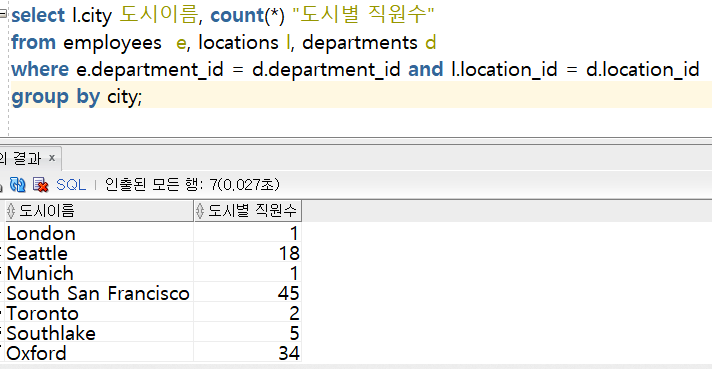


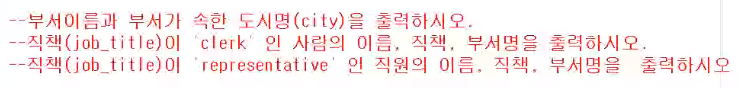


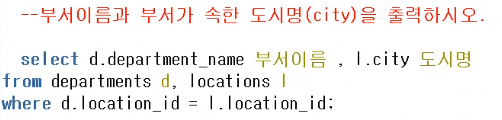


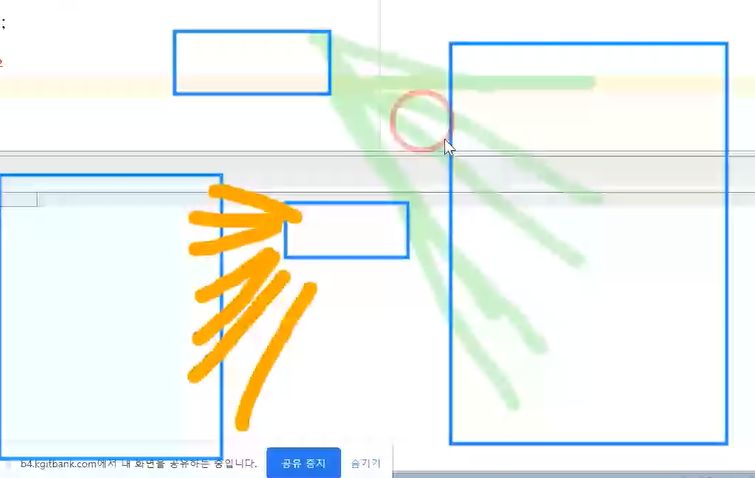




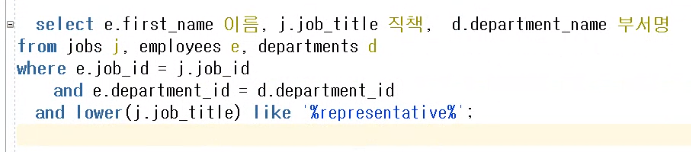


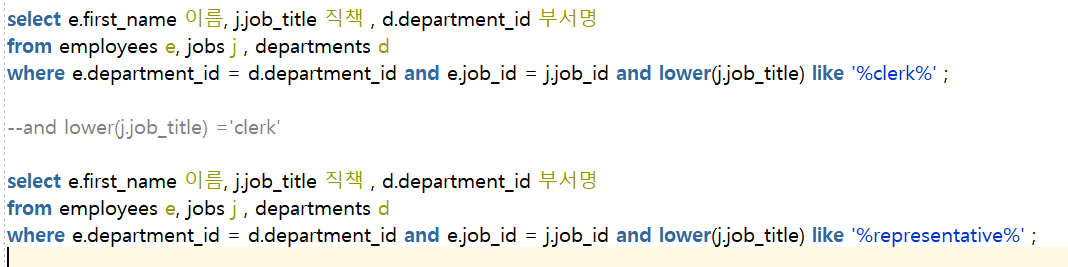


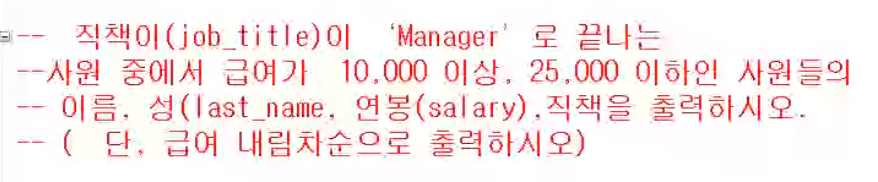


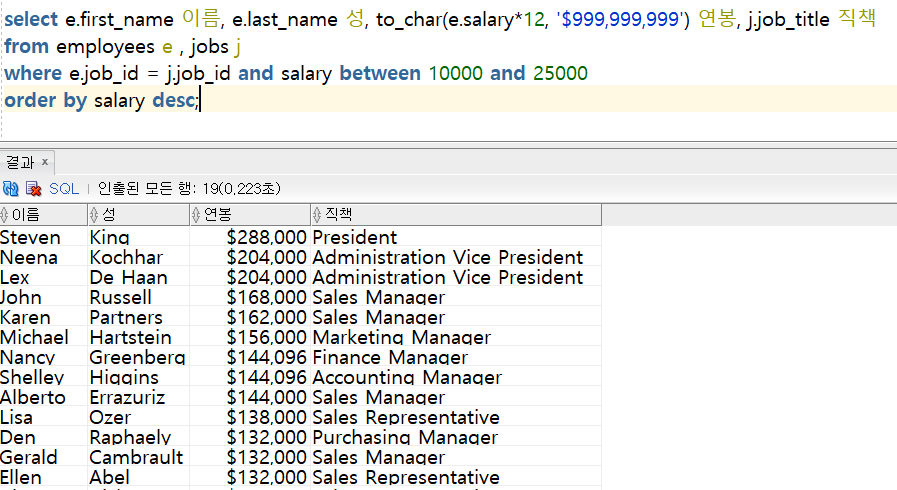
크기가 작은 테이블이 앞에 있는 것이 뒤에 있는 것보다 빠르다.

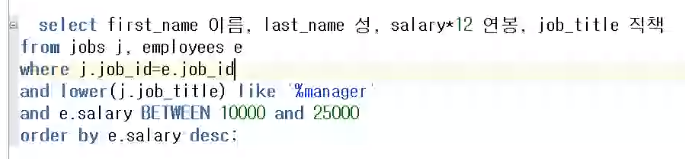
앞의 작은 테이블을 driving table이라고 한다.

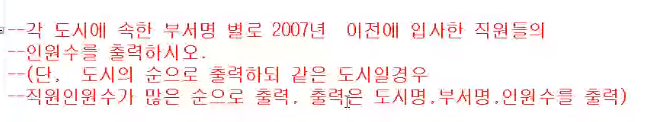


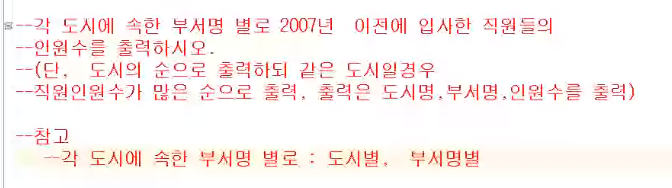




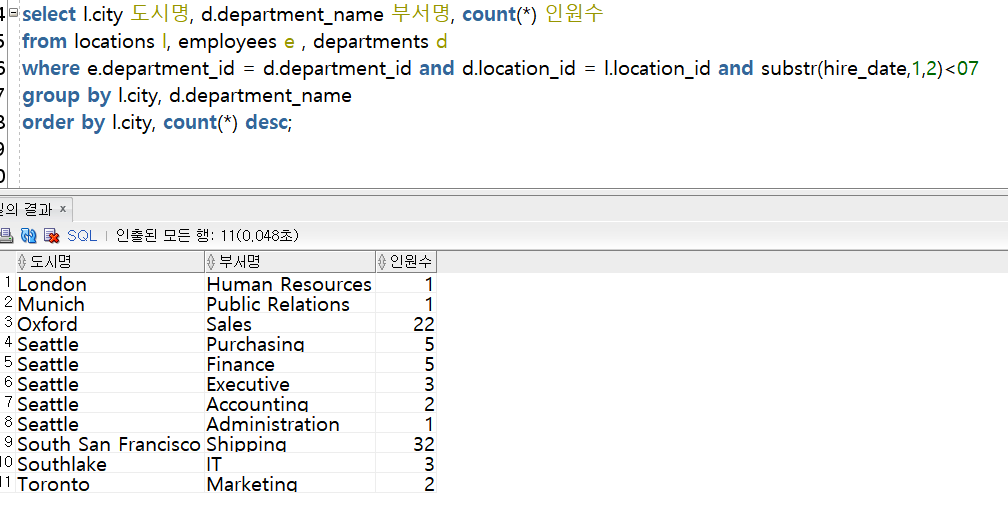


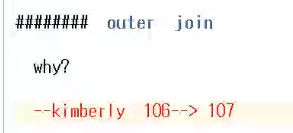


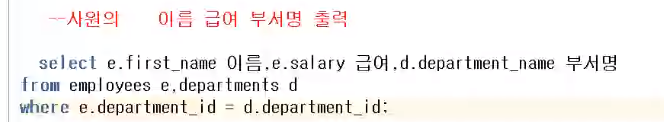


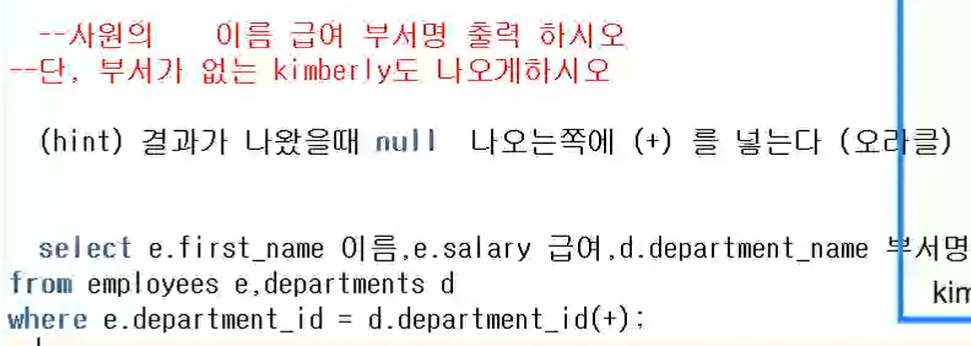








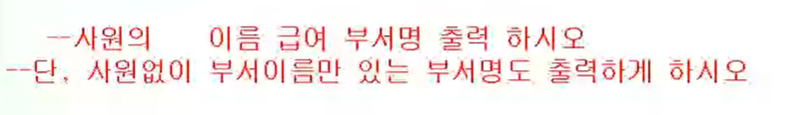




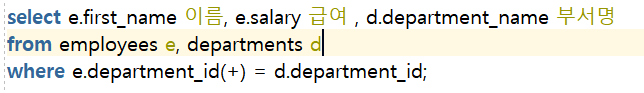




값이 나왔을 때 null이 나오는 쪽에 +를 붙인다.

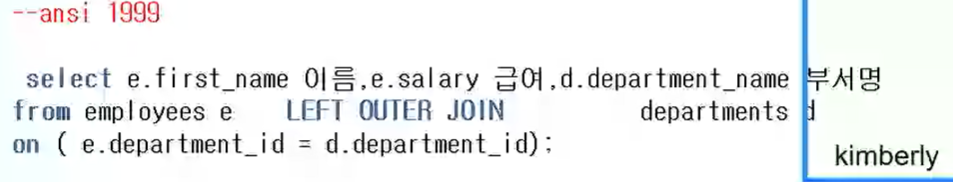


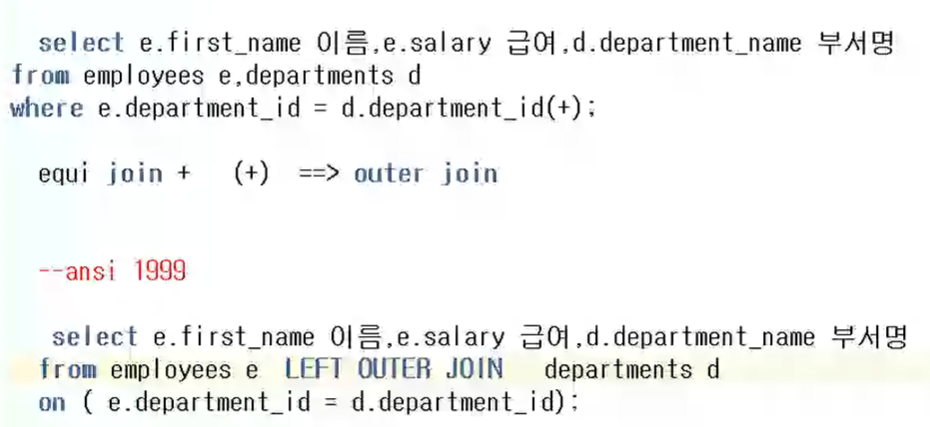
Null인 것도 나와야 한다 -> outer join

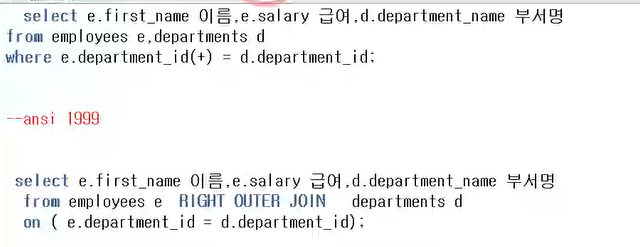


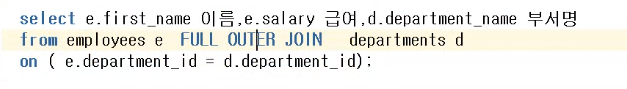


데이터가 있는 쪽에 outer join을 쓴다.

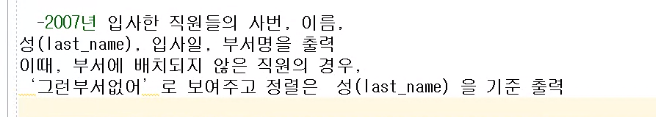




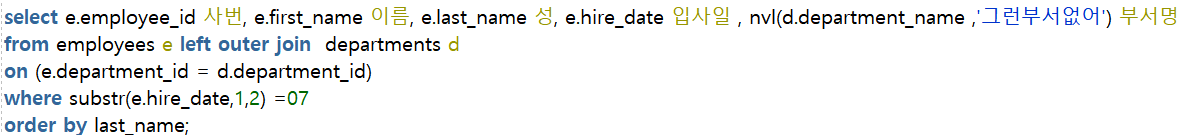


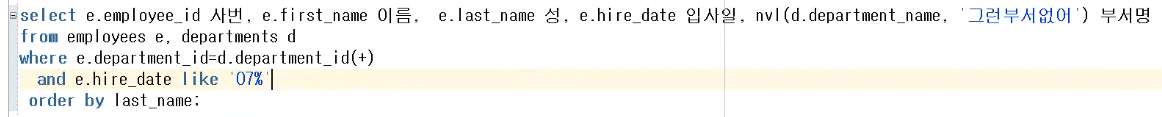


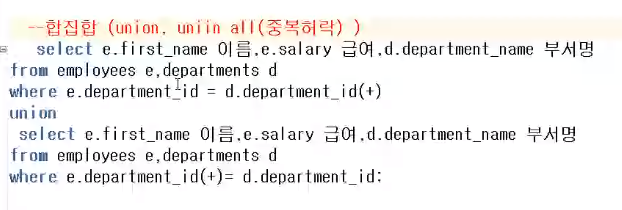
양쪽 다 outer join.

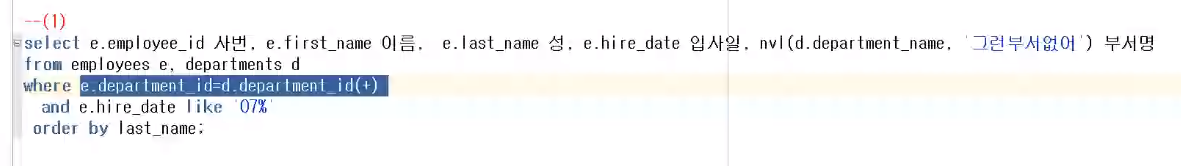


Ansi 방식





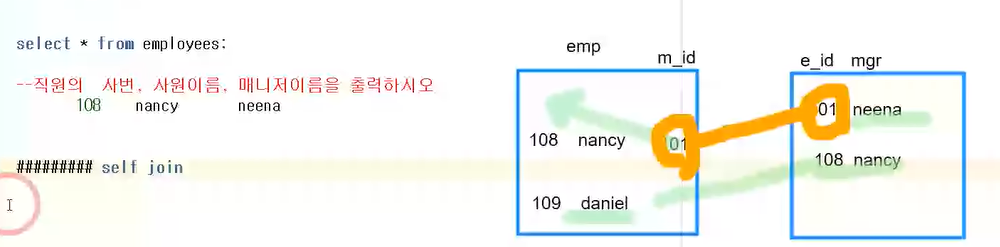




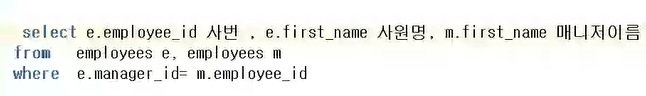


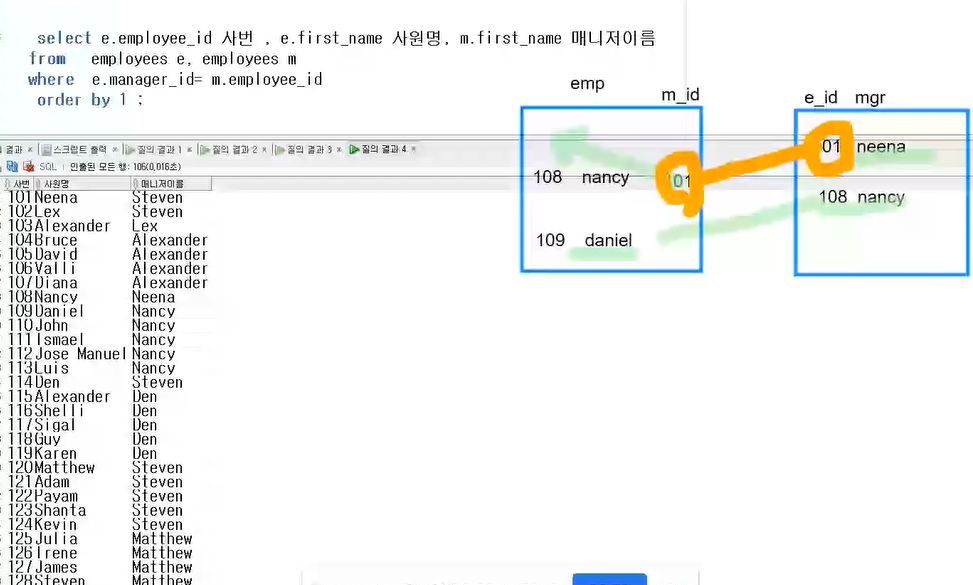
On 다음에 나오는 것은 조인 조건이다. Where에 쓸 것을 on에 쓰지 않도록 주의

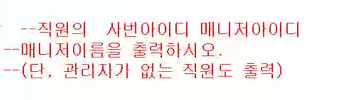


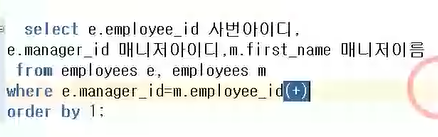


Selft join도 equi join에서 시작한다.

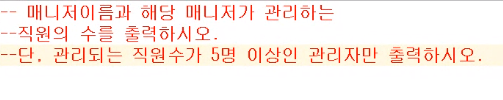




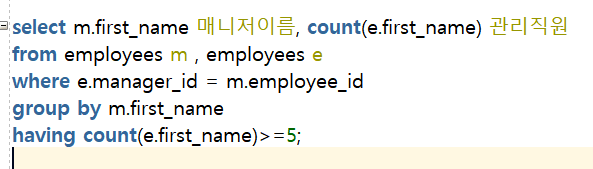




**0611**

****

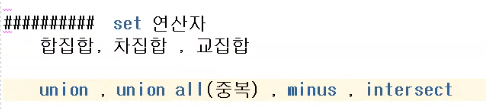
****

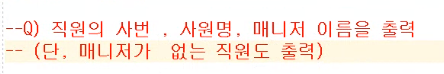
****

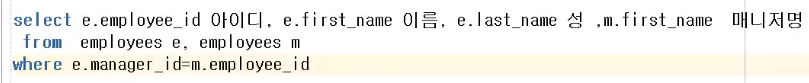
Self join을 사용해서 테이블 내의 관계를 설정해 준다.

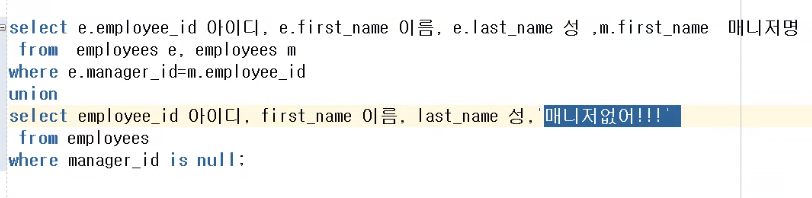
매니저 이름, 관리직원 이름. 일반 직원 테이블의 매니저 아이디가 관리 직원 테이블의 직원 아이디와 일치한다. 그리고 매니저별 관리 직원의 수를 구하는 것이기 때문에 group을 사용해야 한다.

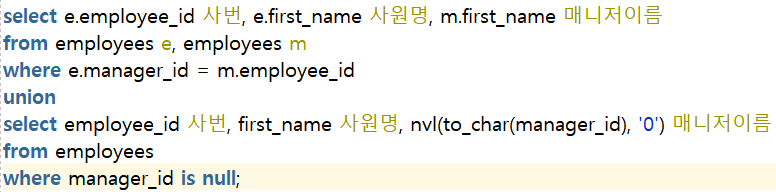
그리고 group의 조건이기 때문에 having을 사용한다.



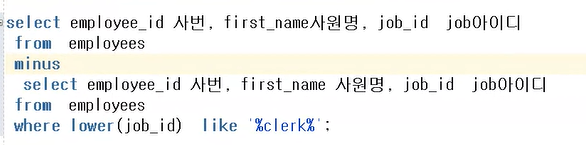




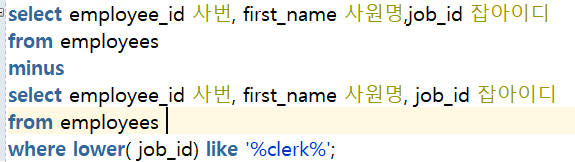




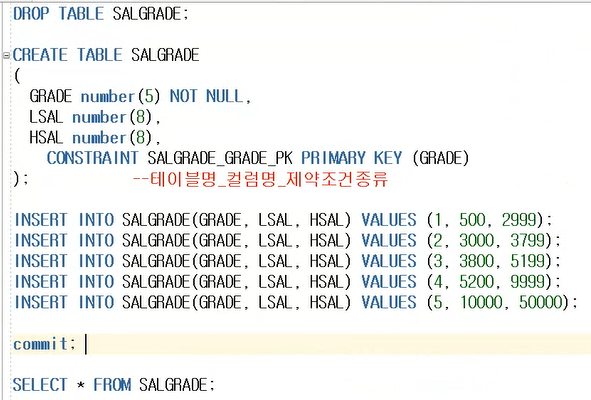


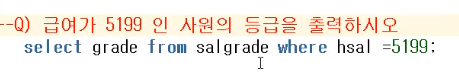


PK, FK 가 같은 EQUI Join. PK, FK로 안 나오는 Outer Join , 한 테이블을 마치 두 개의 테이블로 쓰는 것 Self Join.



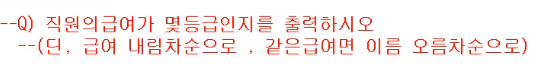
Grade.sql

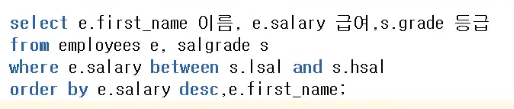










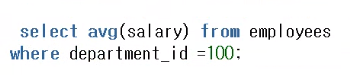


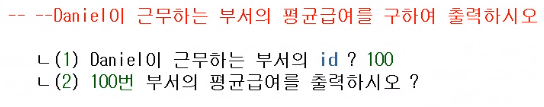


**4가지의 join**

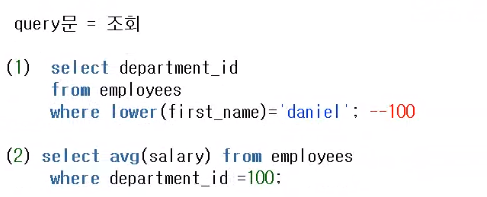
**Main query, Sub Query**

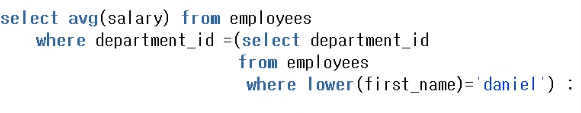


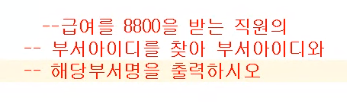


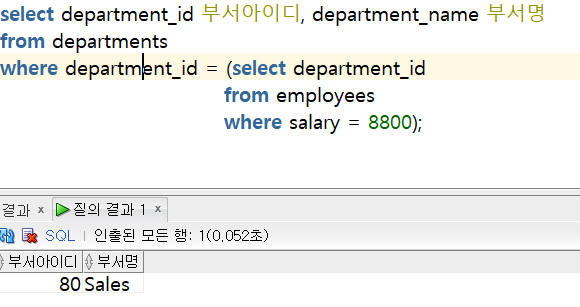


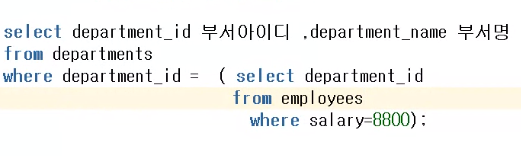
(2)는 구하고자 하는 메인 문제인 main query이고, main query를 위해 도움을 주는 query를 (1) sub Query라고 한다.



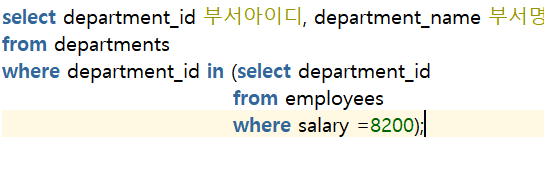


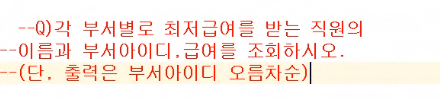


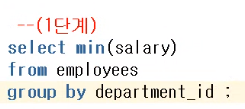


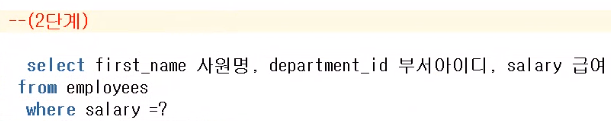


8200으로 바꿔서 돌릴 경우 에러가 나는데, = 은 하나의 값 single row를 산출할 때만 쓰는 것이다. 따라서 여러 개의 값 중 하나를 넣는다는 의미로 in을 사용해야 한다.

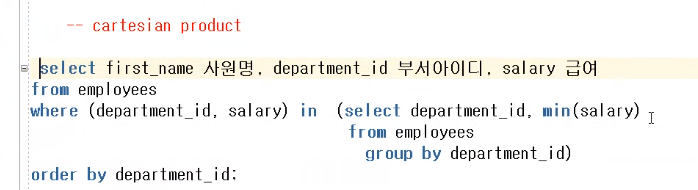




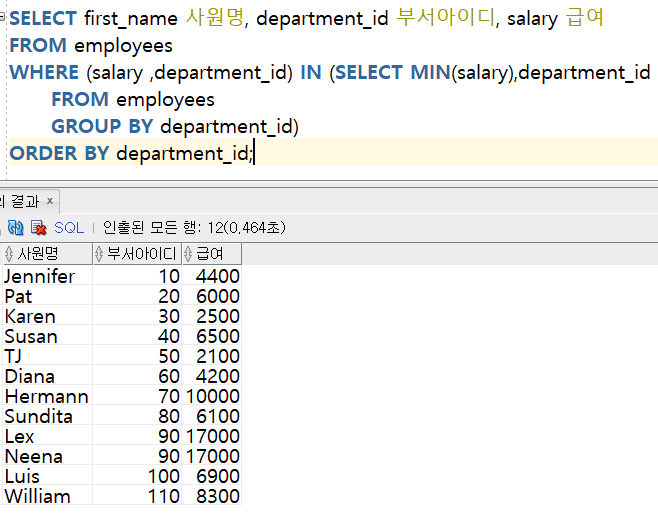


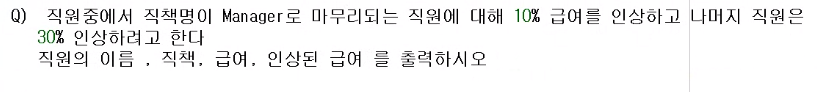


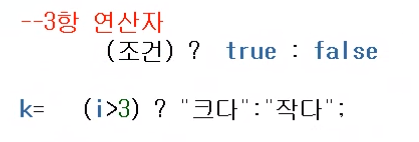
* Pair wise 되었다. 모든 경우의 수가 나오는 것은 cartesian

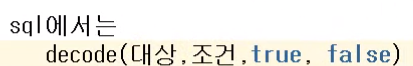


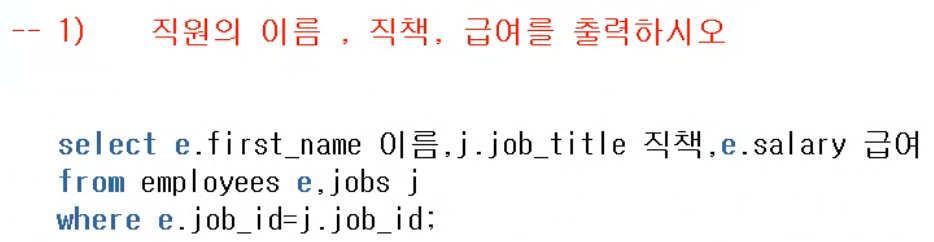
짝을 만들어서 넘기면, 부서에 따라 최소 급여가 넘어가게 된다.







s



2. 여기에 인상된 급여를 추가

