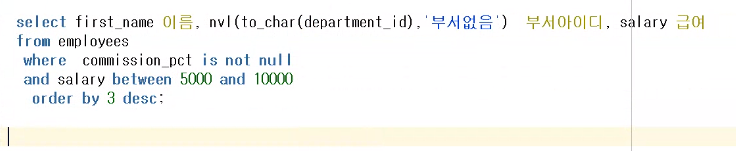
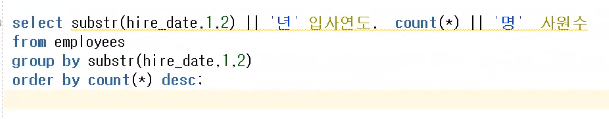
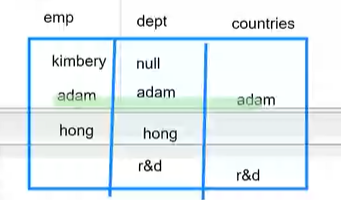
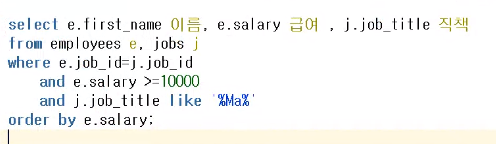
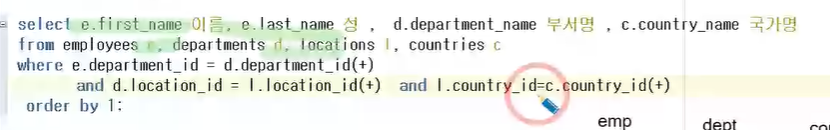
**0616**



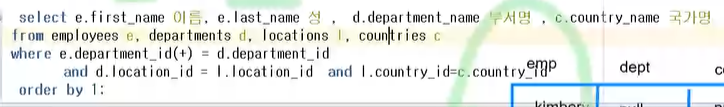




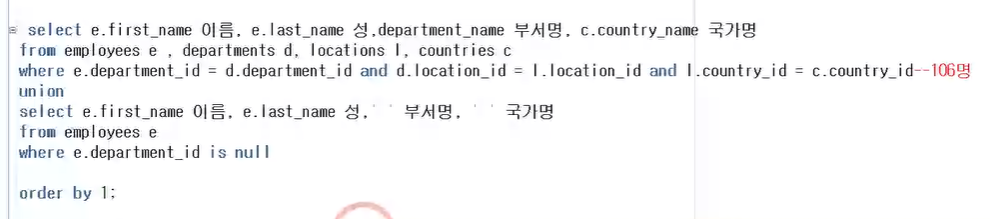


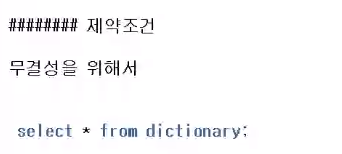
테이블의 기준이 왼쪽이라서 +의 위치를 처리해줘야 한다.

왼쪽은 값, 오른쪽은 null 이 나오게 된다.

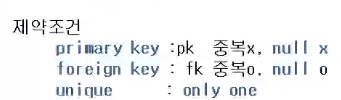


Employee 이름이 null인 경우 e.department\_id에만 붙이고 나머지는 연결되어 있기 때문에 +를 지운다.





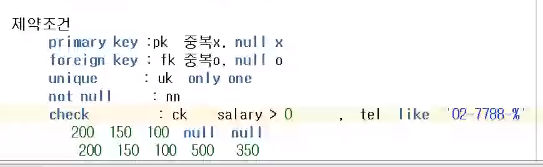


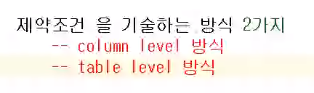




PK는 안되지만, 유일한 값, 중복되지 않는 값을 가지고 있기 때문에 unique하다.

Unique는 uk로 쓴다.

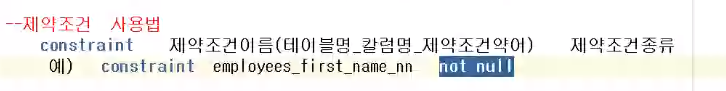




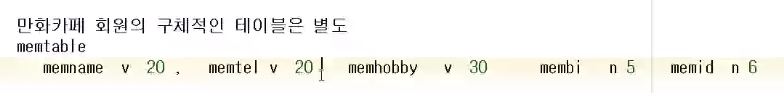
Column level 방식은 제약조건을 컬럼명 바로 뒤에서 이어서 기술

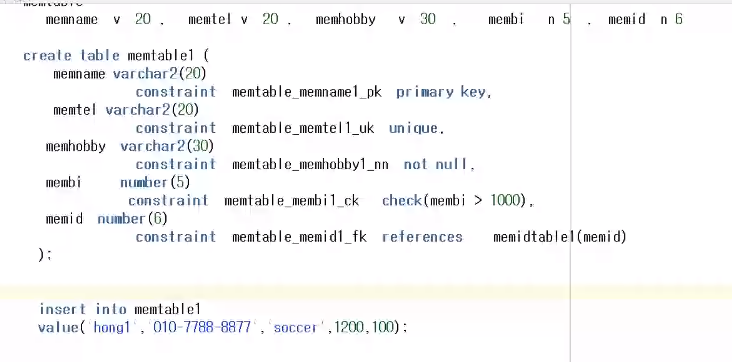
Table level 방식은 제약조건을 맨 뒤에 몰아서 기술.

예외 **not null은 무조건 column level 방식이다!!**



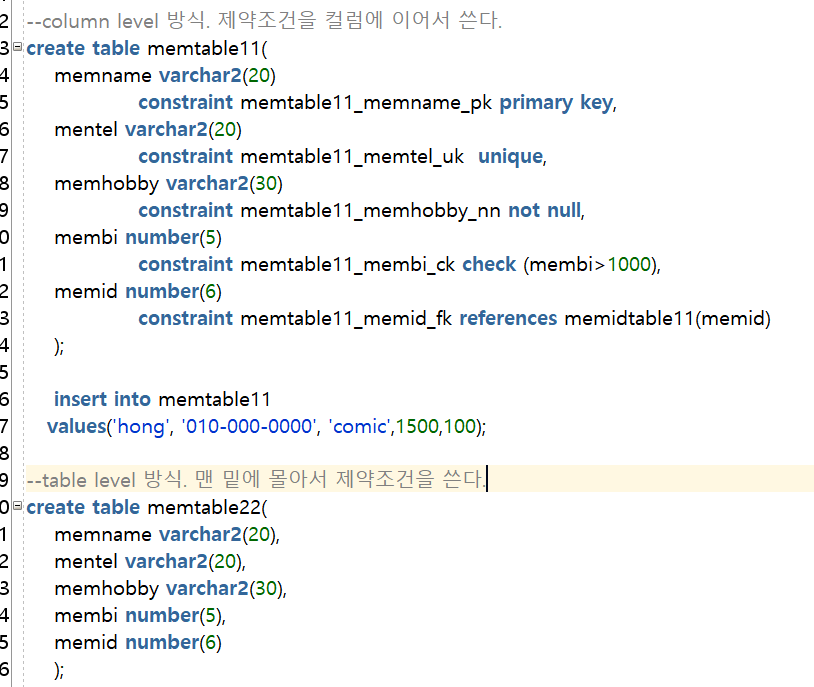






**0617**

Column level 방식



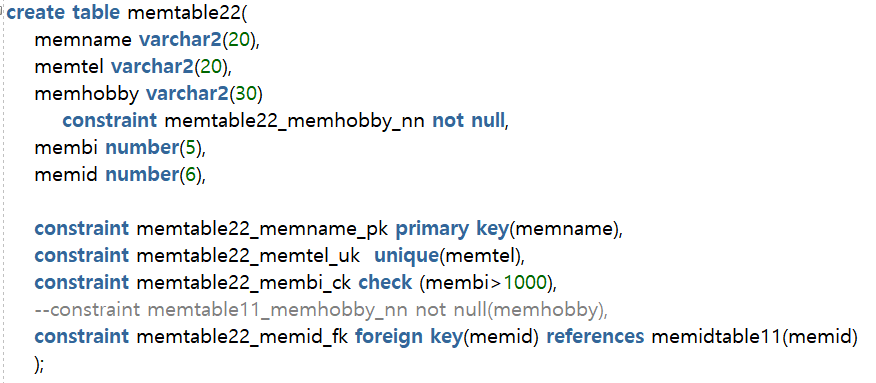


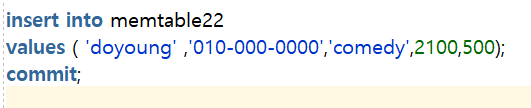
주어 동사 목적어의 형태. 주어가 memid. Memid가 memidtable의 memid를 참조한다.



memidtable11은 pk로 memid를 가진다.

Id 100에 hong, id 500에 doyoung을 넣어주었다.

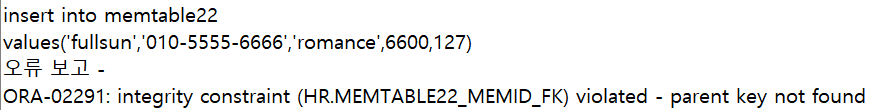




memidtable에서 넣어준 id 500의 값에 다른 정보들을 제약조건에 맞게 값을 추가하였다.

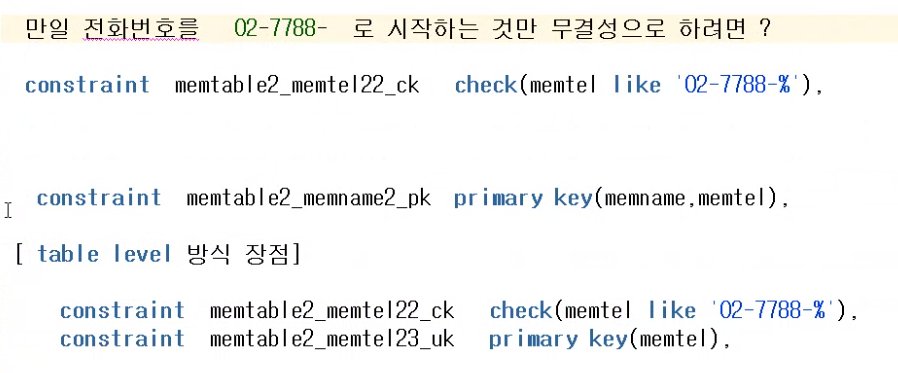


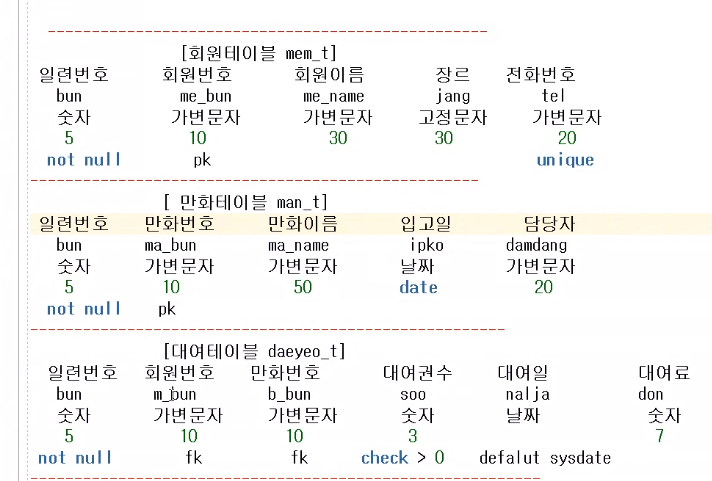
입력하지 않은 memid의 값이 있는 value인 경우 오류가 나며 들어가지 않는다.

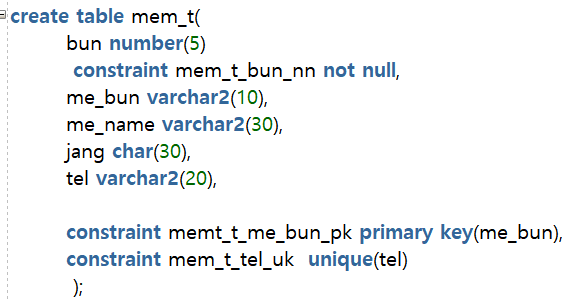


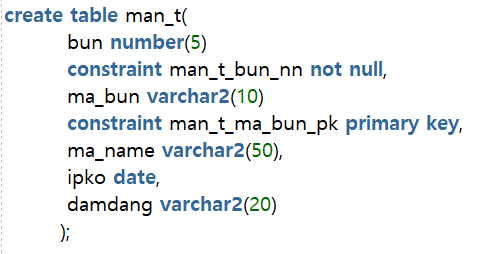




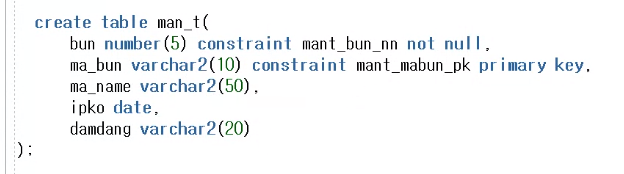


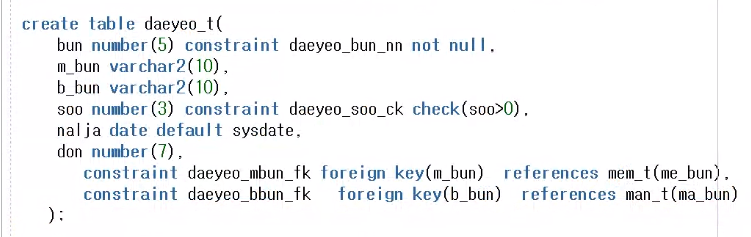


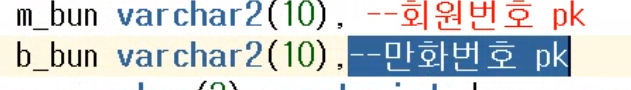




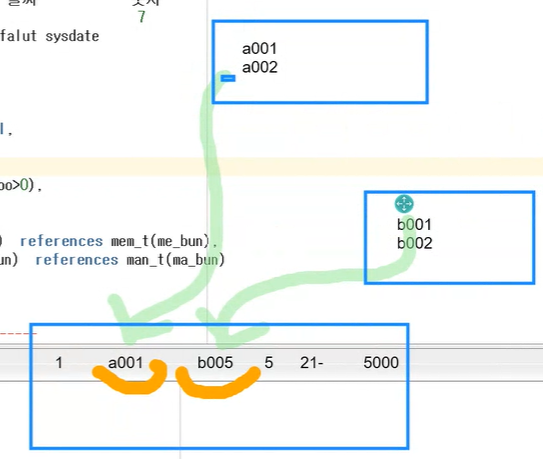


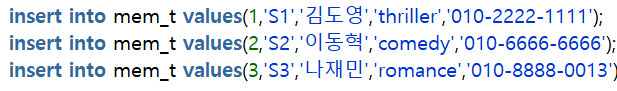


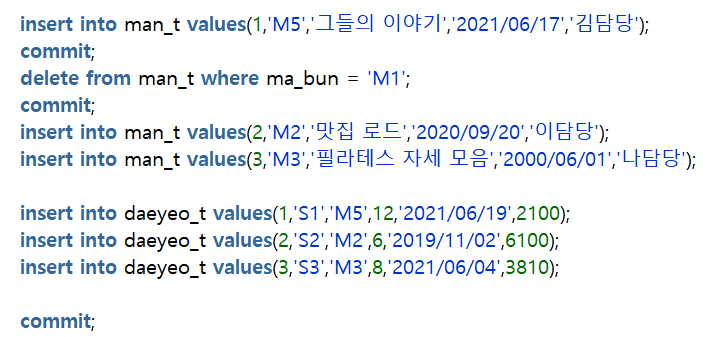


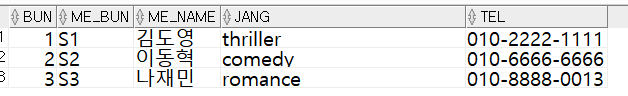


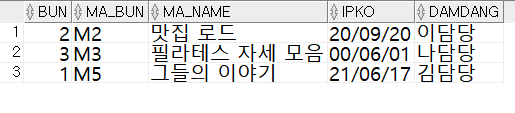
FK 작성 시 어떤 pk를 사용하는지 찾아서 연결시킨다. References 뒤에 나옴.

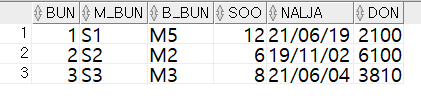




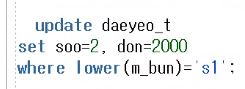






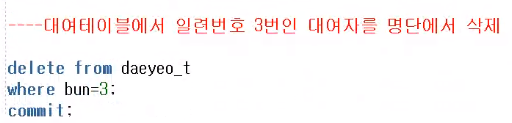


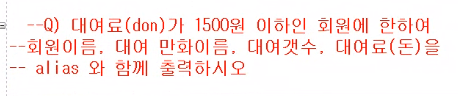


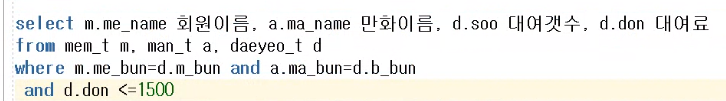


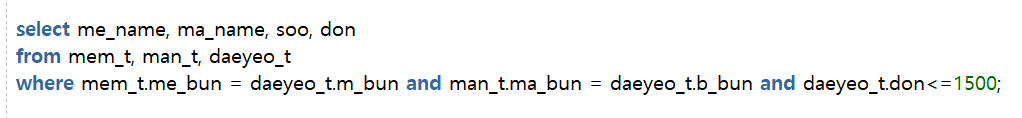


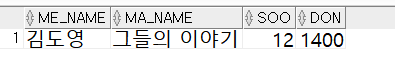






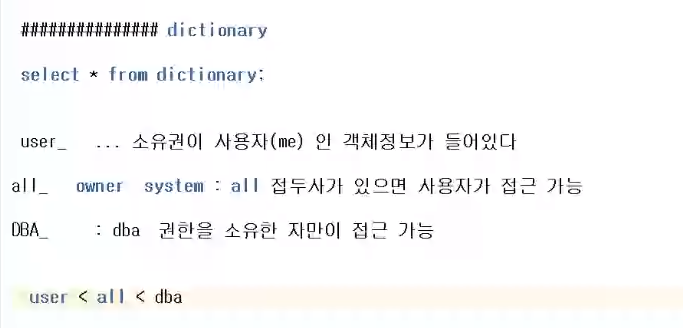


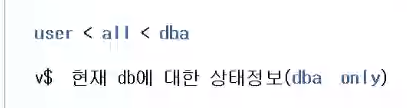




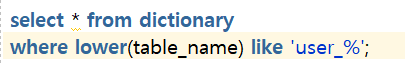
Dictionary

Access 범위

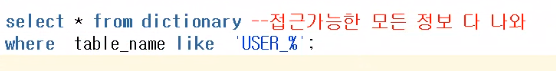




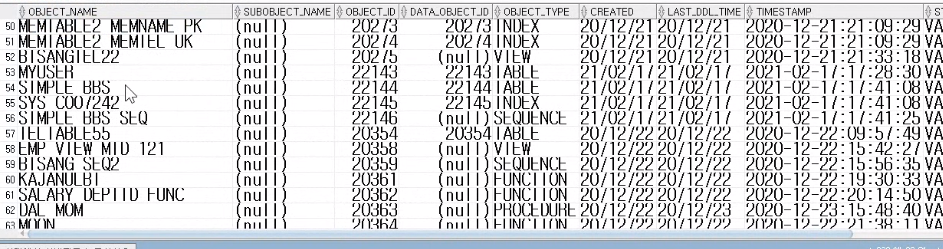
Select \* from dictionary; 사용자가 접근 가능한 모든 정보가 다 나온다.

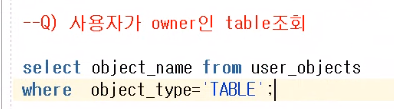


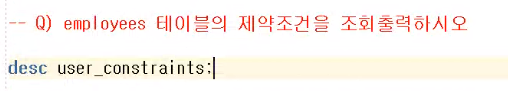
‘ ’ 안에 들어간 테이블명은 기본이 대문자로 들어간다.



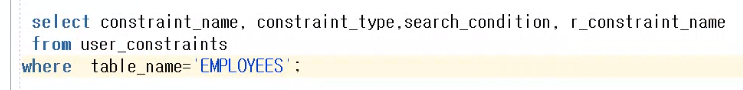


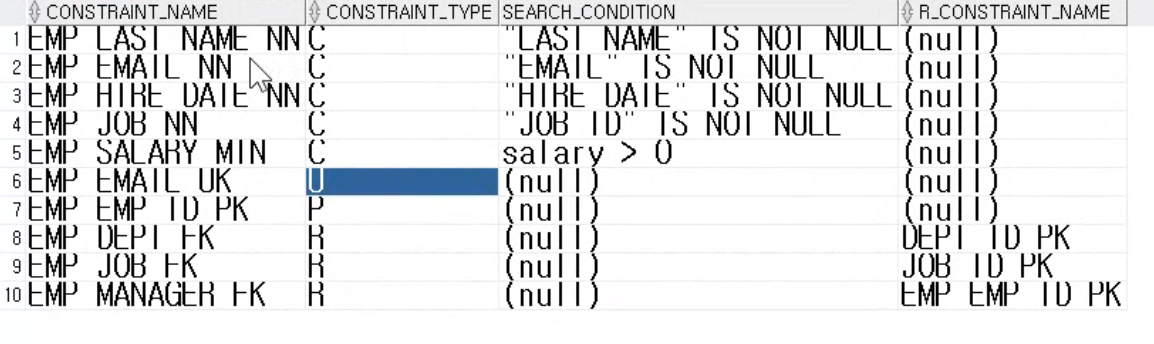








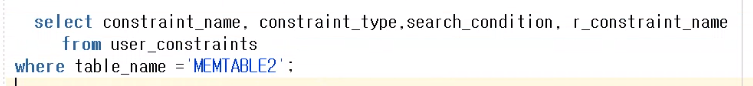


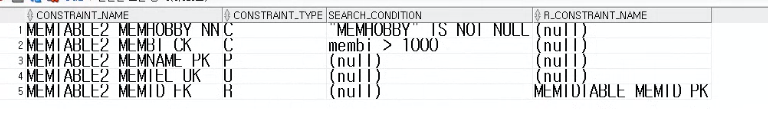


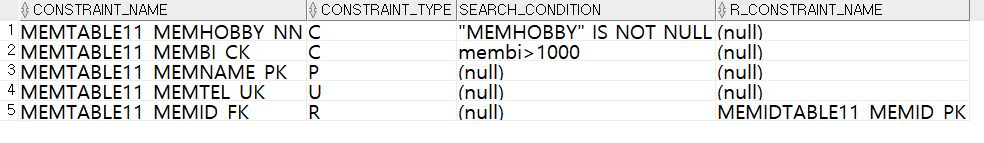
P -> primary, c -> check, R-> foreign key

R\_constraint\_name: 외래키가 참조하는 대상.









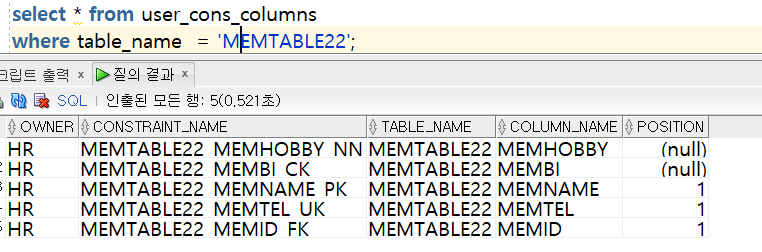


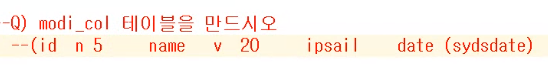
유저가 소유자인 테이블

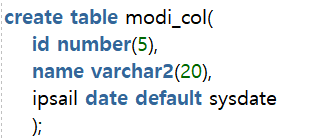


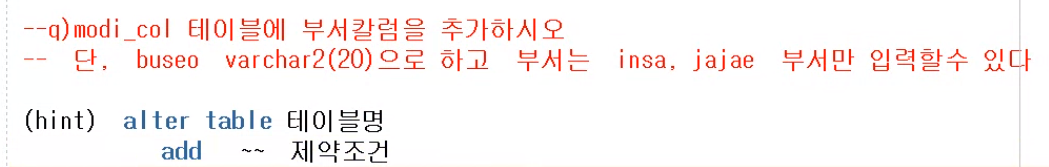
유저가 소유자인 테이블의 제약조건 나온다.

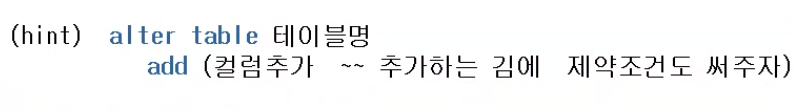
 각 테이블의 제약 조건 나오도록 조건 설정.

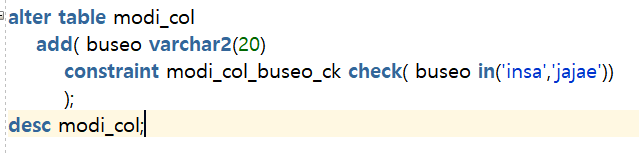


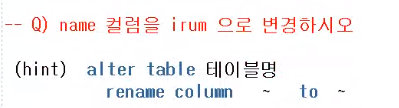


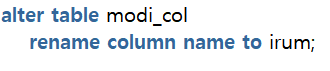








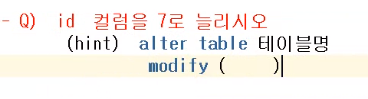


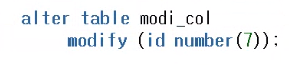


Alter table 테이블명 작용 명령어 + column

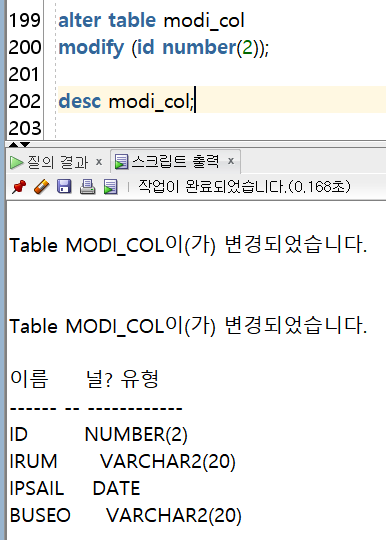
컬럼 추가는 add, 컬럼 이름 바꾸는 것은 rename, 컬럼 지우는 것은 drop





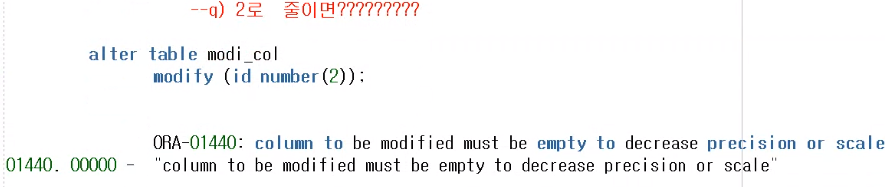


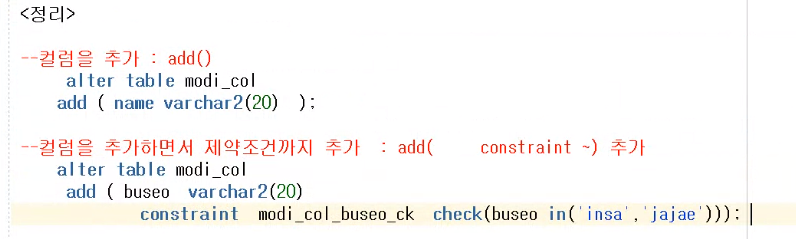
길이를 2로 바꾼다.





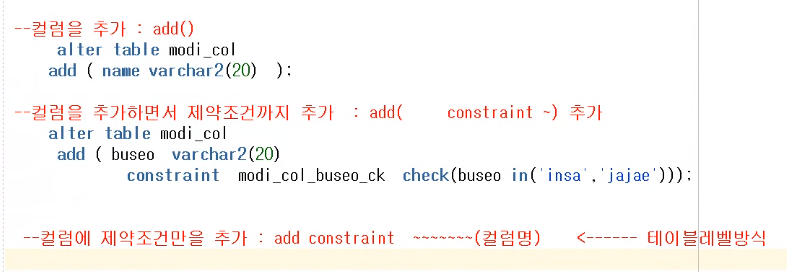
값이 있는데 줄이면 에러가 발생한다.



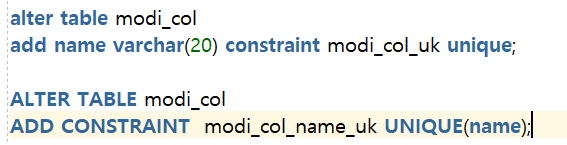


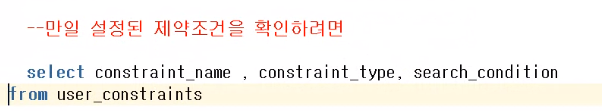


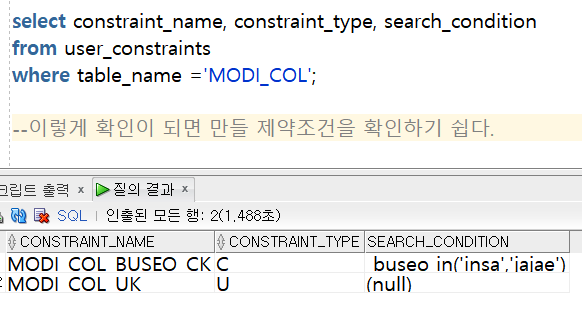
**0618**

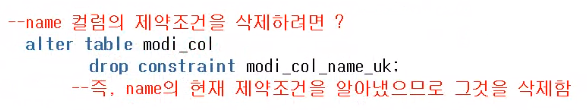


Name 컬럼에 unique를 추가해보자

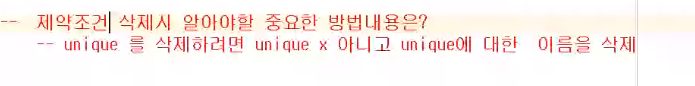








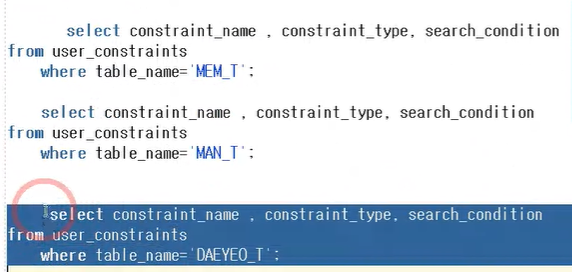
나는 modi\_col\_uk로 추가해서 modi\_col\_uk로 바꿔서 치면 된다. 그 이름대로



alter table man\_t drop primary key cascade;





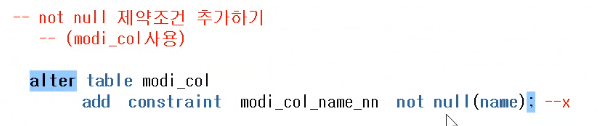


제약 조건이 사라진 것을 볼 수 있다.

제약조건의 이름을 지우면 제약 조건이 사라진다.

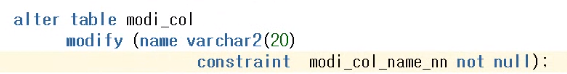
Not null은 컬럼 방식만 된다.

칼럼 추가, 제약 조건 추가는 add 인데, 제약 조건의 변경은 modify이다.

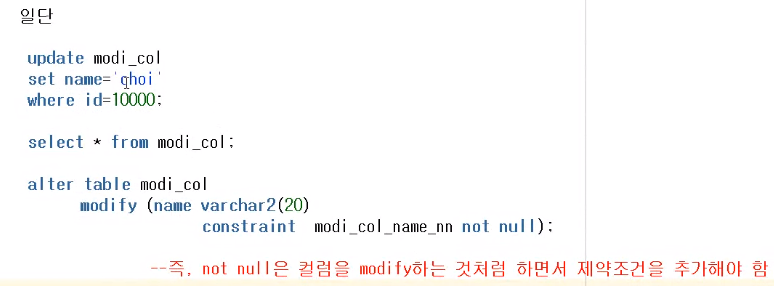




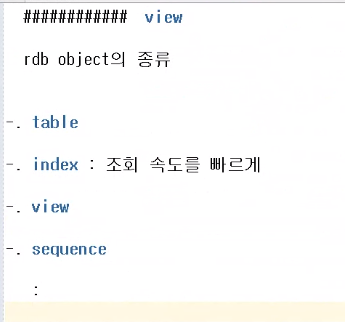


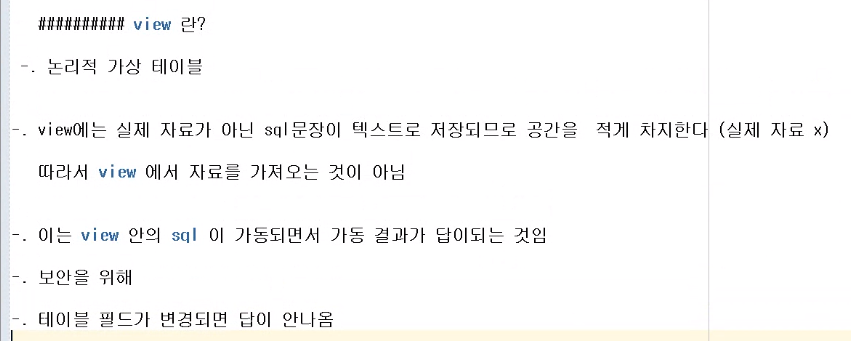


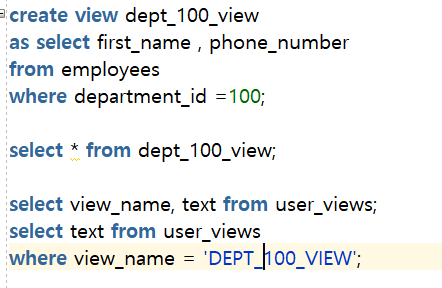


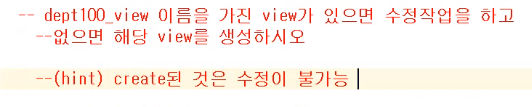


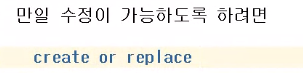




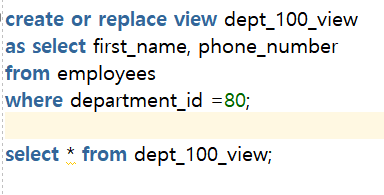


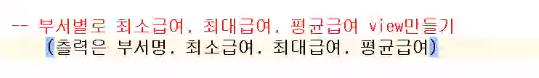


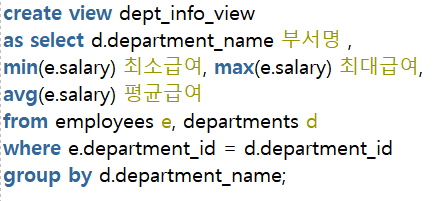


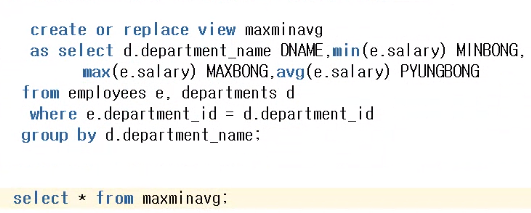


있으면 수정하고. Replace. 없으면 새로 만들게 된다. Create

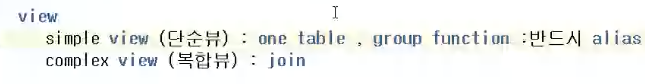




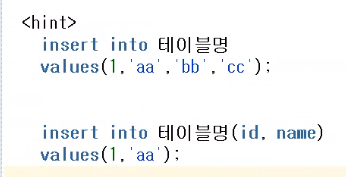


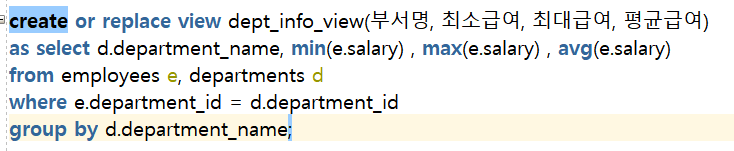


컬럼별 alias를 쓰지 않으면 에러가 난다.

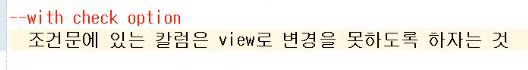


Group function이 들어가면 복합뷰로 본다.

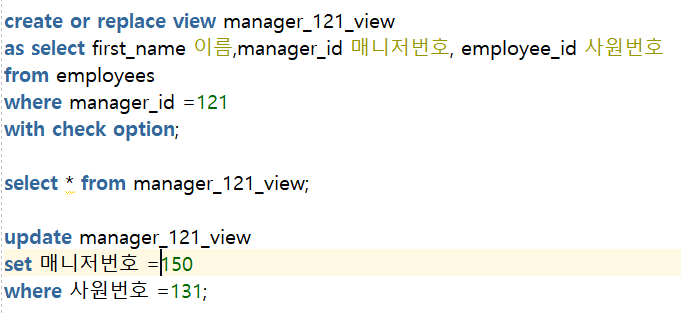


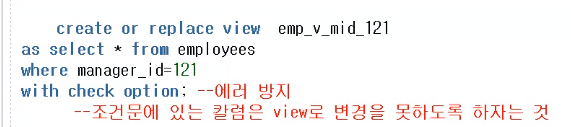


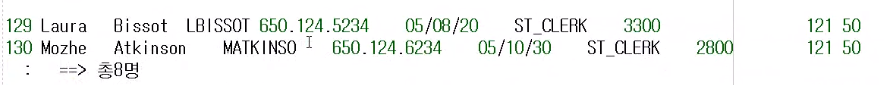
Alias를 다르게 쓰는 방법.

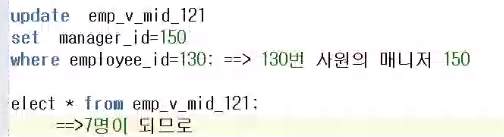


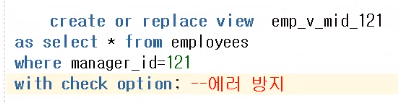




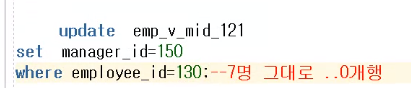




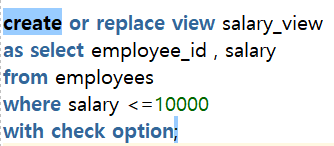




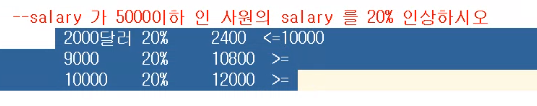
조건문에 있는 column은 view로 변경하지 못하게 막는다.



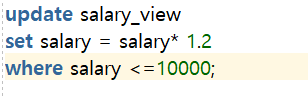




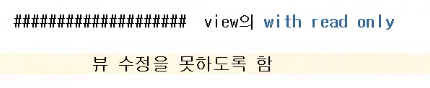


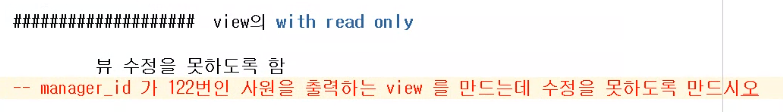


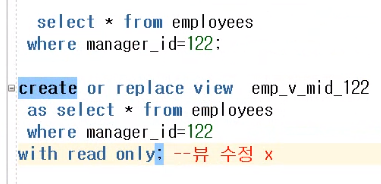
20%인상했을 때 10000을 넘어가는 사원이 존재한다.



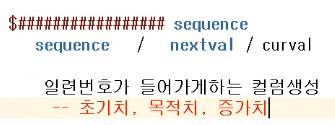
따라서 해당 update 명령문에서 에러가 발생한다.









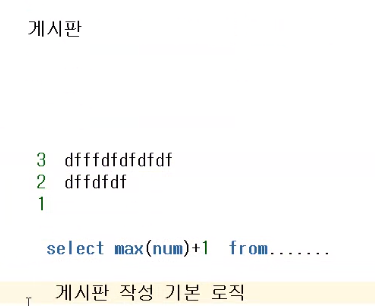
 curval -> currval

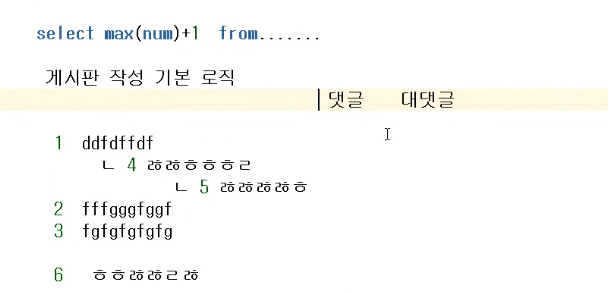




Nextval 그 다음 값, currval: 현재까지 변화된 값

Sequence는 일련번호로 써야지 현재 개수가 몇 인지를 묻는 용도로 쓰면 안 된다.





기존의 글, 댓글, 대댓글 count.

