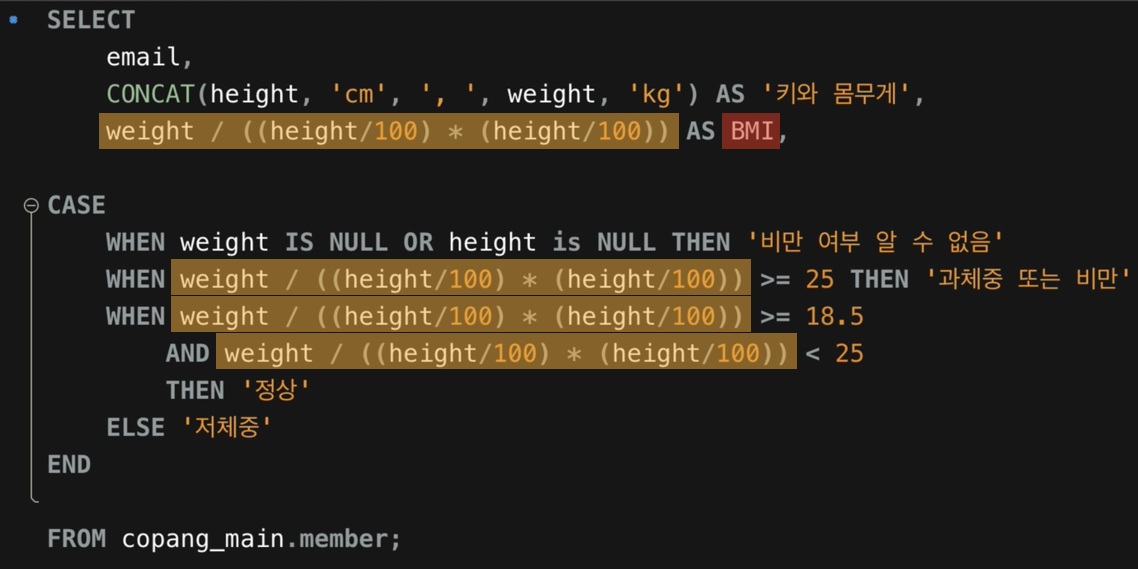
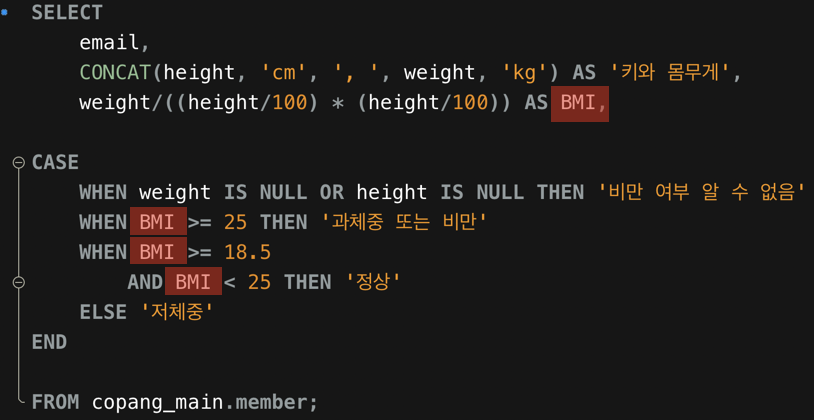
자, 이제 SQL 문을 쓸 때 서브쿼리를 어떻게 활용할 수 있을지 감이 오셨나요?

그럼 잠깐 우리가 이전 챕터에서 해결하지 못했던 내용을 이제 한번 해결해봅시다. 예전에 우리는 [챕터 4의 'alias를 붙이고 바로 쓸 수 없는 이유'라는 노트](https://www.codeit.kr/learn/3191/)를 잠깐 다시 살펴보고 오세요. 이때 우리는

SELECT 절에서 어떤 컬럼에 붙인 **alias**를, **같은 SELECT 절 안에서 재사용하지 못한 문제**가 있었는데요. 그러니까 이렇게

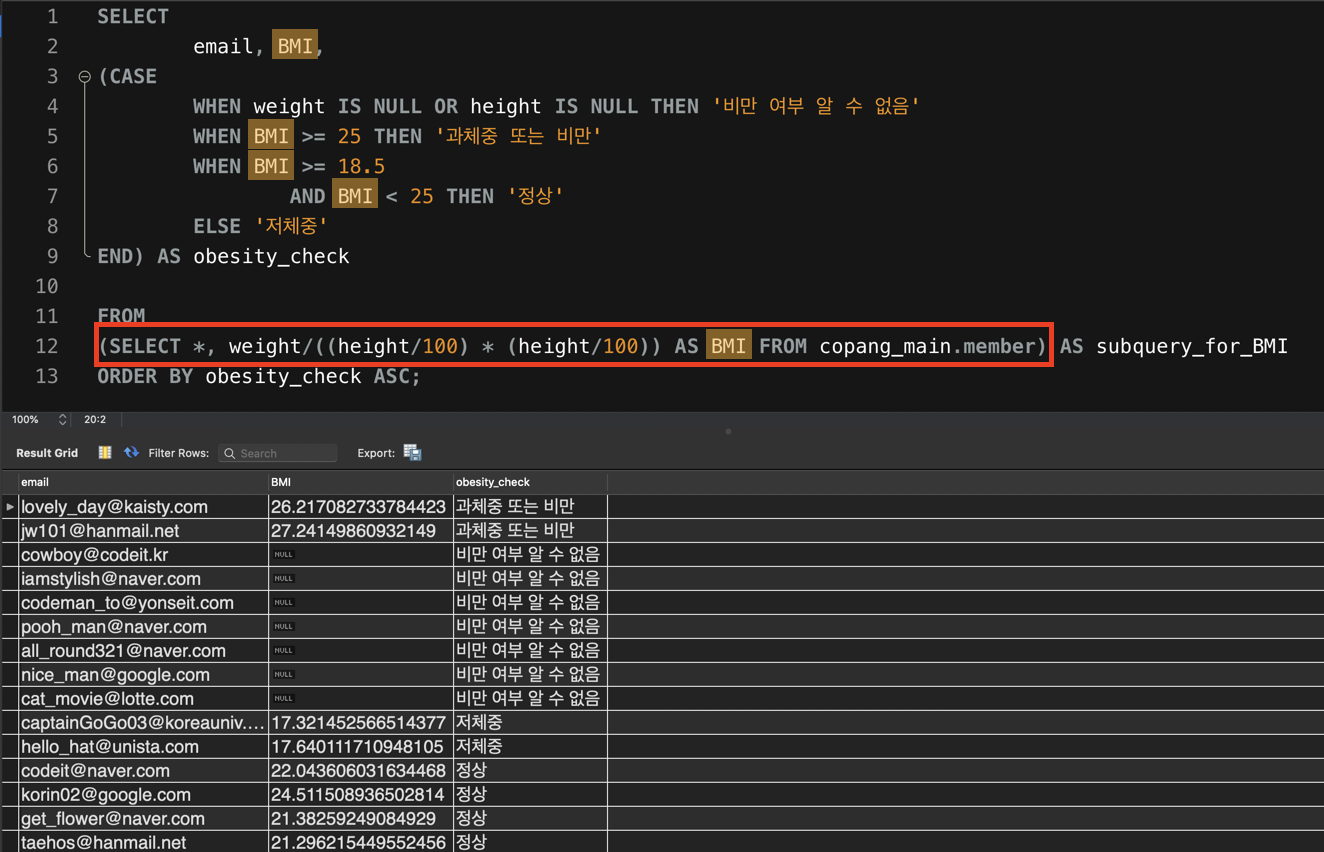


SELECT 절 안에서 설정했던 **BMI**라는 alias를,



동일한 SELECT 절에서 재활용하지 못했습니다. 그래서 어쩔 수 없이 가독성이 좋지 않은 SQL 문을 작성할 수 밖에 없었죠.

하지만 이제 서브쿼리를 배웠으니 그 문제를 해결할 수 있습니다. 아래 그림을 보세요.



지금 BMI라는 alias를 붙인 SELECT 문을 서브쿼리로 만들었습니다. 이 서브쿼리는 FROM 뒤에 있으니까 derived table로 인식되겠죠? 지금 그 derived table에 **subquery\_for\_BMI**라는 alias를 붙인 상태네요.

이제 **subquery\_for\_BMI**는 마치 원래 존재하던 테이블인 것처럼 자유롭게 사용할 수 있습니다. 그래서 지금 보면 outer query에서 BMI라는 단어를 자유롭게 사용하는 것을 볼 수 있습니다.

이렇게 쓰면 마치, 이미 BMI라는 컬럼이 있는 테이블에서 조회를 하는 것과 같기 때문에 이전과는 달리 에러가 발생하지 않습니다. 서브쿼리를 배웠더니 읽기 쉬운 SQL 문을 작성하는데 도움이 되는군요!