

# 데이터 분석을 위한 중급 SQL - CASE 해커랭크 문제풀이

Equilateral : 정삼각형 (3 개의 변의 길이가 같다.)

Isosceles : 이등변 삼각형 (2 개의 변의 길이가 같다.)

Scalene : 변의 길이가 다르다.

Not A Triangle : 삼각형 성립 x (밑변 + 높이가 빗변보다 작다. 밑변 + 높이 < 빗변)

해당 **TRIANGLES** 데이터셋은 다음과 같다.

A	B	C
20	20	20
20	21	22
13	14	30

문제의 답을 요구하는 데이터셋은 다음과 같다.

TypeOfTriangle
Equilateral
Scalene
Triangle

## 나의 풀이

```
SELECT CASE
    WHEN A = B AND B = C AND C = A THEN 'Equilateral'
    WHEN (A = B OR B = C OR C = A) AND (A + B > C AND A + C > B AND B
+ C > A) THEN 'Isosceles'
    WHEN (A != B AND B != C AND C != A) AND (A + B > C AND A + C > B
AND B + C > A) THEN 'Scalene'
    ELSE 'Not A Triangle'
END
FROM TRIANGLES
```

정삼각형, 이등변 삼각형, 삼각형, Not A Triangle 순으로 SELECT CASE 조건문을 정의했다. 쿼리 구문 가독성이 떨어지며, 내부 구동 시간이 더 오래 걸릴 것임.

## 선미님의 답안

---

```
SELECT CASE
    WHEN A = B AND B = C AND C = A THEN 'Equilateral'
    WHEN A + B <= C OR A + C <= B OR B + C <= A THEN 'Not A Triangle'
    WHEN A = B OR B = C OR A = C THEN 'Isosceles'
    ELSE 'Scalene'
END
FROM TRIANGLES;
```

정삼각형, Not A Triangle, 이등변 삼각형, 그냥 삼각형 순으로 정의했다. 쿼리 구문이 간결하며, 내부 구동 시간이 훨씬 단축 될 것이다.

## 깨달은 점

---

SELECT CASE 조건문에 대해서 중복되는 조건에 대해 위로 올리면 가시성도 좋을 뿐더러 내부 구동 시간이 단축 될 것이다. Devide and Conquer and optimization, 나눠서 생각하고 정복하고 정복한 것을 최적화 시킨다. IT에서는 이 세가지를 꼭 기억하며 문제들을 풀어내면 좋을 것이다.