

아래 쿼리는 인라인 뷰(t2)와 메인 쿼리(t3)에 스칼라 서브 쿼리를 사용했다. 최종 SELECT 목록 순서(t3 -> t2)대로 스칼라 서브 쿼리가 표시되는 것을 확인할 수 있다.

```

1  -- 2
2  SELECT (SELECT y.c2
3          FROM t3 y
4          WHERE y.c1 = a.c1) AS t3
5  FROM (SELECT a.*
6          , (SELECT x.c2
7              FROM t2 x
8              WHERE x.c1 = a.c1) AS t2
9          FROM t1 a) a;
11
12  -----
13  | Id | Operation                                | Name |
14  -----
15  | 0 | SELECT STATEMENT                        |      |
16  | 1 | TABLE ACCESS BY INDEX ROWID          | T3    |
17  | * 2 | INDEX RANGE SCAN                      | T3_X1 |
18  | 3 | TABLE ACCESS BY INDEX ROWID          | T2    |
19  | * 4 | INDEX RANGE SCAN                      | T2_X1 |
20  | 5 | TABLE ACCESS FULL                    | T1    |
21  -----
22
23  Predicate Information (identified by operation id):
24  -----
25  2 - access("Y"."C1"=:B1)
26  4 - access("X"."C1"=:B1)

```

↑ 스칼라 서브쿼리에 대한 operation은 'SELECT STATEMENT' operation 직전에 실시된다.

아래 쿼리는 스칼라 서브 쿼리를 중첩했다. 중첩 스칼라 서브 쿼리의 안쪽(t3)부터 표시되는 것을 확인할 수 있다.

```

1  -- 3
2  SELECT a.*
3  , (SELECT (SELECT y.c2
4              FROM t3 y
5              WHERE y.c1 = x.c1) AS t4
6      FROM t2 x
7      WHERE x.c1 = a.c1
8      AND x.c2 > 0) AS t3
9  FROM t1 a;
11
12  -----
13  | Id | Operation                                | Name |
14  -----
15  | 0 | SELECT STATEMENT                        |      |
16  | 1 | TABLE ACCESS BY INDEX ROWID          | T3    |
17  | * 2 | INDEX RANGE SCAN                      | T3_X1 |
18  | * 3 | TABLE ACCESS BY INDEX ROWID          | T2    |
19  | * 4 | INDEX RANGE SCAN                      | T2_X1 |
20  | 5 | TABLE ACCESS FULL                    | T1    |
21  -----
22
23  Predicate Information (identified by operation id):
24  -----
25  2 - access("Y"."C1"=:B1)
26  3 - filter("X"."C2">0)
27  4 - access("X"."C1"=:B1)

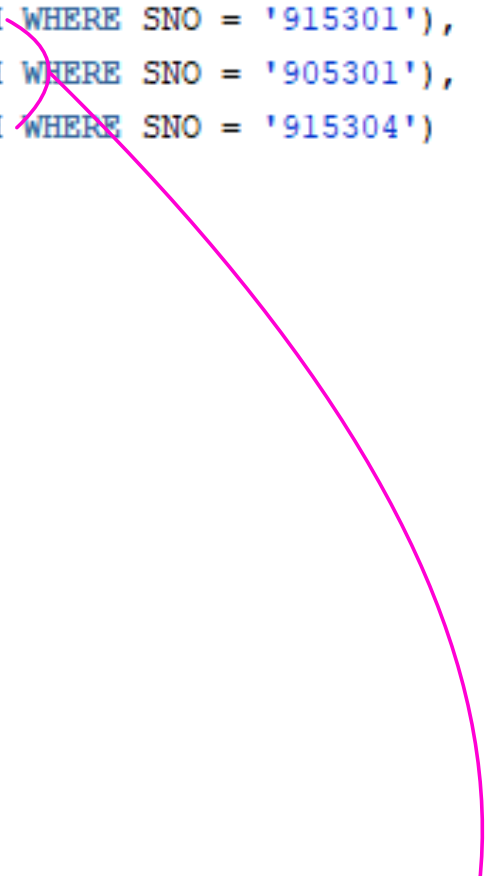
```

쿼리 2, 3에서 얻을 수 있는 규칙은 단순하다. 스칼라 서브 쿼리는 최종 SELECT 목록 순서로 표시되며, 중첩 스칼라 서브 쿼리 안쪽부터 표시된다. 규칙을 통해 대략의 위치를 파악하고, Predicate Information을 통해 확인하는 방식을 사용하면 스칼라 서브 쿼리를 쉽게 식별할 수 있다.

```
WITH FIRST_WITH AS (
SELECT /*+ INLINE NO_MERGE */
      CNO,
      CNAME,
      PNO
FROM COURSE
),

SECOND_WITH AS (
SELECT /*+ INLINE NO_MERGE */
      SNO,
      SNAME
FROM STUDENT
WHERE SNO LIKE '%1%'
)

SELECT (SELECT CNAME FROM FIRST_WITH WHERE CNAME = '일반화학실험'),
      (SELECT SNAME FROM SECOND_WITH WHERE SNO = '915301'),
      (SELECT SNAME FROM SECOND_WITH WHERE SNO = '905301'),
      (SELECT SNAME FROM SECOND_WITH WHERE SNO = '915304')
FROM DUAL;
```



Id	Operation	Name	E-Rows
0	SELECT STATEMENT		
1	VIEW		1
* 2	TABLE ACCESS FULL	COURSE	1
3	VIEW		1
* 4	TABLE ACCESS FULL	STUDENT	1
5	VIEW		1
* 6	TABLE ACCESS FULL	STUDENT	1
7	VIEW		1
* 8	TABLE ACCESS FULL	STUDENT	1
9	FAST DUAL		1

여러 스칼라 서브쿼리가
같은 table에
동일한 로직을
적용하더라도,
각 스칼라 서브쿼리가
별로 실행된다.

Predicate Information (identified by operation id):

2 - filter("CNAME"='일반화학실험')

4 - filter(("SNO"='915301' AND "SNO" LIKE '%1%' AND "SNO" IS NOT
NULL))

6 - filter(("SNO"='905301' AND "SNO" LIKE '%1%' AND "SNO" IS NOT
NULL))

8 - filter(("SNO"='915304' AND "SNO" LIKE '%1%' AND "SNO" IS NOT
NULL))