

서브쿼리

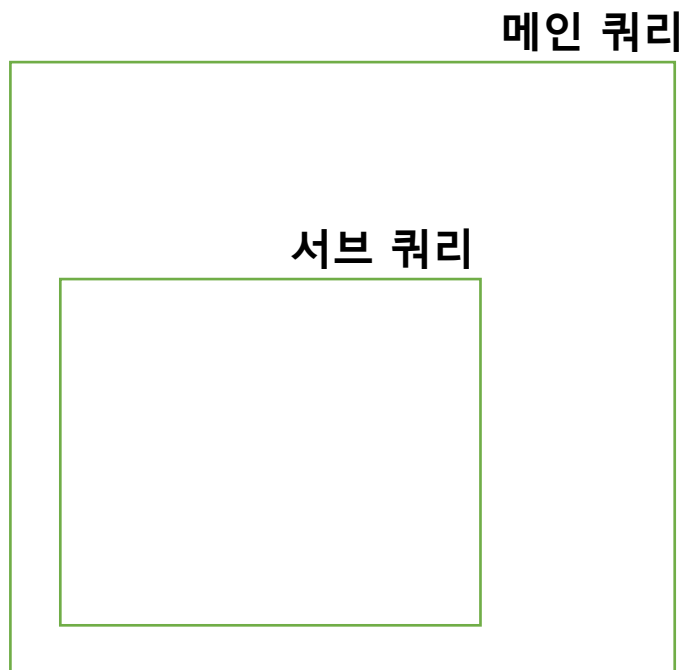
(SUB QUERY)

현업 사용 정도 : ★ ★ ★ ★ ★

1. 서브쿼리의 개념
2. 스칼라 서브쿼리 (사용 : SELECT)
3. 인라인 뷰(사용 : FROM)
4. 중첩서브쿼리 (사용 : WHERE)
& 단일행/다중행 연산자

서브쿼리(SUB QUERY)란?

작성된 쿼리 내부에 다른 쿼리를 삽입하여 보다 다양하게 데이터를 출력하는 방법을 제공합니다.



메인 쿼리

```
SELECT A.MEMBER_ID AS 회원ID
      , A.MEMBER_NAME AS 회원이름
      , (
          SELECT TEL_NO
            FROM TB_MEMBER_TEL
           WHERE TEL_DV_CD = '휴대폰'
              AND MEMBER_ID = A.MEMBER_ID ) AS 휴대폰번호
FROM TB_MEMBER A
WHERE A.MEMBER_ID IN ('AAAAA' , 'BBBBB' , 'CCCCC' , 'DDDDD') ;
```

서브 쿼리

The diagram shows a SQL query with a subquery. The main query is enclosed in a red box and labeled '메인 쿼리'. The subquery, which selects the phone number for a specific member, is enclosed in a smaller red box and labeled '서브 쿼리'. The subquery is identified by the label 'AS 휴대폰번호'.

2. 스칼라 서브쿼리 (사용 : SELECT)

스칼라 서브쿼리 (Scala Subquery)

- **SELECT** 에서 사용되는 서브쿼리
- 하나의 컬럼에 대해 하나의 행만 반환하는 특징을 보유
- 출력되는 값이 없다면 NULL을 반환

```
SELECT A.MEMBER_ID           AS 회원ID
      , A.MEMBER_NAME        AS 회원이름
      , (
          SELECT TEL_NO
            FROM TB_MEMBER_TEL
           WHERE TEL_DV_CD = '휴대폰'
             AND MEMBER_ID = A.MEMBER_ID ) AS 휴대폰번호
FROM TB_MEMBER A
WHERE A.MEMBER_ID IN ('AAAAA' , 'BBBBB' , 'CCCCC' , 'DDDDD') ;
```

회원ID	회원이름	휴대폰번호
AAAAA	사용자A	010-1231-1231
BBBBB	사용자B	010-5555-8888
CCCCC	사용자C	(null)
DDDDD	사용자D	(null)

2. 스칼라 서브쿼리 (사용 : SELECT)

스칼라 서브쿼리의 실행 원리

SELECT가 실행되는 횟수만큼 각각의 인스턴스에 반복 대입 / 출력 됩니다.

```
SELECT A.MEMBER_ID          AS 회원ID
      , A.MEMBER_NAME       AS 회원이름
      , (
          SELECT TEL_NO
            FROM TB_MEMBER_TEL
           WHERE TEL_DV_CD = '휴대폰'
             AND MEMBER_ID = A.MEMBER_ID ) AS 휴대폰번호
FROM TB_MEMBER A
WHERE A.MEMBER_ID IN ('AAAAA' , 'BBBBB' , 'CCCCC' , 'DDDDD') ;
```

- (1) 가장 먼저 FROM 이 실행이 됩니다.
- (2) WHERE 이 실행되면서 출력될 행의 수가 4행으로 한정이 됩니다.
- (3) SELECT 가 4번이 실행이 되며 , 이 때 스칼라 서브쿼리도 총 4번 실행됩니다.

2. 스칼라 서브쿼리 (사용 : SELECT)

스칼라 서브쿼리의 실행 원리

SELECT가 실행되는 횟수만큼 각각의 인스턴스에 반복 대입 / 출력 됩니다.

```
SELECT A.MEMBER_ID          AS 회원ID      AAAAA
      , A.MEMBER_NAME       AS 회원이름    사용자A
      , (
        SELECT TEL_NO        (3) 010-1231-1231 출력
          FROM TB_MEMBER_TEL
         WHERE TEL_DV_CD = '휴대폰'
           AND MEMBER_ID = A.MEMBER_ID ) AS 휴대폰번호
      FROM TB_MEMBER A
      WHERE A.MEMBER_ID IN ('AAAAA' , 'BBBBB' , 'CCCCC' , 'DDDDD') ;
```

회원ID	회원이름	휴대폰번호
AAAAA	사용자A	010-1231-1231
BBBBB	사용자B	010-5555-8888
CCCCC	사용자C	(null)
DDDDD	사용자D	(null)

(2) 서브쿼리 실행

```
SELECT TEL_NO
      FROM TB_MEMBER_TEL
     WHERE TEL_DV_CD = '휴대폰'
       AND MEMBER_ID = 'AAAAA' ;
```

TEL_NO
010-1231-1231

2. 스칼라 서브쿼리 (사용 : SELECT)

스칼라 서브쿼리의 실행 원리

SELECT가 실행되는 횟수만큼 각각의 인스턴스에 반복 대입 / 출력 됩니다.

```
SELECT A.MEMBER_ID          AS 회원ID      BBBBB
      , A.MEMBER_NAME        AS 회원이름    사용자B
      , (
        SELECT TEL_NO        (3) 010-5555-8888 출력
          FROM TB_MEMBER_TEL
         WHERE TEL_DV_CD = '휴대폰'
               AND MEMBER_ID = A.MEMBER_ID ) AS 휴대폰번호
      FROM TB_MEMBER A
      WHERE A.MEMBER_ID IN ('AAAAA' , 'BBBBB' , 'CCCCC' , 'DDDDD') ;
```

회원ID	회원이름	휴대폰번호
AAAAA	사용자A	010-1231-1231
BBBBB	사용자B	010-5555-8888
CCCCC	사용자C	(null)
DDDDD	사용자D	(null)

(2) 서브쿼리 실행

```
SELECT TEL_NO
      FROM TB_MEMBER_TEL
     WHERE TEL_DV_CD = '휴대폰'
           AND MEMBER_ID = 'BBBBB' ;
```

TEL_NO
010-5555-8888

2. 스칼라 서브쿼리 (사용 : SELECT)

스칼라 서브쿼리의 실행 원리

SELECT가 실행되는 횟수만큼 각각의 인스턴스에 반복 대입 / 출력 됩니다.

```
SELECT A.MEMBER_ID          AS 회원ID      CCCCC
      , A.MEMBER_NAME        AS 회원이름    사용자C
      , (
        SELECT TEL_NO (3) NULL 출력
          FROM TB_MEMBER_TEL
         WHERE TEL_DV_CD = '휴대폰'
           AND MEMBER_ID = A.MEMBER_ID ) AS 휴대폰번호
      FROM TB_MEMBER A
      WHERE A.MEMBER_ID IN ('AAAAA' , 'BBBBB' , 'CCCCC' , 'DDDDD') ;
```

회원ID	회원이름	휴대폰번호
AAAAA	사용자A	010-1231-1231
BBBBB	사용자B	010-5555-8888
CCCCC	사용자C	(null)
DDDDD	사용자D	(null)

(2) 서브쿼리 실행

```
SELECT TEL_NO
FROM TB_MEMBER_TEL
WHERE TEL_DV_CD = '휴대폰'
      AND MEMBER_ID = 'CCCCC' ;
```

SQL | 인출된 모든 행: 0(0.003초)

TEL_NO

2. 스칼라 서브쿼리 (사용 : SELECT)

스칼라 서브쿼리의 실행 원리

SELECT가 실행되는 횟수만큼 각각의 인스턴스에 반복 대입 / 출력 됩니다.

```
SELECT A.MEMBER_ID                AS 회원ID      DDDDD
      , A.MEMBER_NAME              AS 회원이름    사용자D
      , (
        SELECT TEL_NO              (3) NULL 출력
          FROM TB_MEMBER_TEL
         WHERE TEL_DV_CD = '휴대폰'
               AND MEMBER_ID = A.MEMBER_ID ) AS 휴대폰번호
FROM TB_MEMBER A                  (1) DDDDD 대입
WHERE A.MEMBER_ID IN ('AAAAA' , 'BBBBB' , 'CCCCC' , 'DDDDD' ) ;
```

회원ID	회원이름	휴대폰번호
AAAAA	사용자A	010-1231-1231
BBBBB	사용자B	010-5555-8888
CCCCC	사용자C	(null)
DDDDD	사용자D	(null)

(2) 서브쿼리 실행

```
SELECT TEL_NO
FROM TB_MEMBER_TEL
WHERE TEL_DV_CD = '휴대폰'
      AND MEMBER_ID = 'DDDDD' ;
```

SQL | 인출된 모든 행: 0(0.003초)
TEL_NO

2. 스칼라 서브쿼리 (사용 : SELECT) – 사용시 주의사항

스칼라 서브쿼리로 출력되는 **행(튜플)**은 1개 혹은 NULL이어야 합니다.

```
SELECT A.MEMBER_ID          AS 회원ID
      , A.MEMBER_NAME       AS 회원이름
      , (
          SELECT TEL_NO
            FROM TB_MEMBER_TEL
           WHERE 1=1 -- TEL_DV_CD = '휴대폰'
              AND MEMBER_ID = A.MEMBER_ID ) AS 휴대폰번호
FROM TB_MEMBER A
WHERE A.MEMBER_ID IN ('AAAAA' , 'BBBBB' , 'CCCCC' , 'DDDDD') ;
```

MEMBER_ID 가 AAAAA , BBBBB 인 계정은
여러 연락처를 보유하고 있음 (2개 이상!)

ORA-01427: single-row subquery returns more than one row
01427, 00000 - "single-row subquery returns more than one row"

```
SELECT TEL_NO
FROM TB_MEMBER_TEL
WHERE 1=1 -- TEL_DV_CD = '휴대폰'
AND MEMBER_ID = 'AAAAA' ;
```

TEL_NO
062-123-1234
02-9999-9999
010-1231-1231

```
SELECT TEL_NO
FROM TB_MEMBER_TEL
WHERE 1=1 -- TEL_DV_CD = '휴대폰'
AND MEMBER_ID = 'BBBBB' ;
```

TEL_NO
062-555-7777
010-5555-8888

2. 스칼라 서브쿼리 (사용 : SELECT) – 사용시 주의사항

스칼라 서브쿼리로 출력되는 컬럼은 1개여야 합니다.

```
SELECT A.MEMBER_ID AS 회원ID
      , A.MEMBER_NAME AS 회원이름
      , (
        SELECT TEL_NO , TEL_DV_CD
        FROM TB_MEMBER_TEL
        WHERE TEL_DV_CD = '휴대폰'
        AND MEMBER_ID = A.MEMBER_ID ) AS 휴대폰번호
FROM TB_MEMBER A
WHERE A.MEMBER_ID IN ('AAAAA' , 'BBBBB' , 'CCCCC' , 'DDDDD') ;
```

스칼라 서브쿼리에서 출력되는 컬럼이 2개임

ORA-00913: too many values
00913, 00000 - "too many values"

```
SELECT TEL_NO , TEL_DV_CD
FROM TB_MEMBER_TEL
WHERE TEL_DV_CD = '휴대폰'
AND MEMBER_ID = 'AAAAA' ;
```

TEL_NO	TEL_DV_CD
010-1231-1231	휴대폰

2. 스칼라 서브쿼리 (사용 : SELECT)

문제1)

TB_MEMBER 테이블에는 GRADE_CD 라는 컬럼이 있습니다.

스칼라 서브쿼리를 이용해서 TB_GRADE 테이블에 있는 GRADE_NAME 값을 가져와 데이터를 출력해주세요 .

힌트 :

```
SELECT [redacted] AS 회원ID
, [redacted] AS 회원이름
, A.GRADE_CD AS 등급코드
, (
    SELECT [redacted]
    FROM [redacted]
    WHERE GRADE_CD = [redacted] ) AS 등급이름
FROM [redacted] A ;
```

출력결과

회원ID	회원이름	등급코드	등급이름
AAAAA	사용자A	1	브론즈
BBBBB	사용자B	2	실버
CCCCC	사용자C	1	브론즈
DDDDD	사용자D	3	골드
EEEEE	사용자E	1	브론즈
FFFFF	사용자F	3	골드
GGGGG	사용자G	2	실버
HHHHH	사용자H	5	VVIP
IIIII	사용자I	4	VIP

2. 스칼라 서브쿼리 (사용 : SELECT)

문제 2)

TB_ORDER 테이블에는 각각 MEMBER_ID 와 PRD_ID 컬럼을 이용해 TB_MEMBER 테이블과 TB_PRD 테이블과 연결되어 있습니다.
TB_ORDER 테이블의 MEMBER_ID 컬럼을 이용하면 TB_MEMBER 테이블에서 MEMBER_NAME을 가져올 수 있습니다.
TB_ORDER 테이블의 PRD_ID 컬럼을 이용하면 TB_PRD 테이블에서 PRD_NAME 을 가져올 수 있습니다.

아래와 같이 2가지 방법을 이용하여 출력결과를 뽑아주세요.

- (1) 세 테이블을 조인을 해서 가져오는 방법
- (2) 스칼라 서브쿼리를 이용해서 각 정보를 가져오는 방법

** 조인 풀이 시 힌트

각 테이블에서 가져올 컬럼 정보
[TB_ORDER 테이블 => ORDER_NO , MEMBER_ID , PRD_ID]
[TB_MEMBER 테이블 => MEMBER_NAME]
[TB_PRD 테이블 => PRD_NAME]

** 스칼라 서브쿼리 풀이 시 힌트

```
SELECT A. [블록] AS 주문번호
, A. [블록] AS 회원ID
, (
    SELECT [블록]
    FROM TB_MEMBER
    WHERE [블록] ) AS 주문자이름
, A. [블록] AS 상품ID
, (
    SELECT [블록]
    FROM TB_PRD
    WHERE [블록] ) AS 상품명
FROM TB_ORDER A ;
```

출력결과

주문번호	회원ID	주문자이름	상품ID	상품명
1	AAAAA	사용자A	P0002	에어컨
2	AAAAA	사용자A	P0003	세탁기
3	AAAAA	사용자A	P0004	건조기
4	AAAAA	사용자A	P0020	수건

2. 스칼라 서브쿼리 (사용 : SELECT)

문제 1 답)

```
SELECT A.MEMBER_ID          AS 회원ID
      , A.MEMBER_NAME       AS 회원이름
      , A.GRADE_CD          AS 등급코드
      , (
          SELECT GRADE_NAME
            FROM TB_GRADE
           WHERE GRADE_CD = A.GRADE_CD ) AS 등급이름
FROM TB_MEMBER A ;
```

출력결과

회원ID	회원이름	등급코드	등급이름
AAAAA	사용자A	1	브론즈
BBBBB	사용자B	2	실버
CCCCC	사용자C	1	브론즈
DDDDD	사용자D	3	골드
EEEEE	사용자E	1	브론즈
FFFFF	사용자F	3	골드
GGGGG	사용자G	2	실버
HHHHH	사용자H	5	VVIP
IIIII	사용자I	4	VIP

2. 스칼라 서브쿼리 (사용 : SELECT)

문제 2 답)

--조인 풀이 방법

```
SELECT A.ORDER_NO      AS 주문번호
      , A.MEMBER_ID    AS 회원ID
      , B.MEMBER_NAME  AS 주문자이름
      , A.PRD_ID       AS 상품ID
      , C.PRD_NAME     AS 상품명
FROM TB_ORDER A
     , TB_MEMBER B
     , TB_PRD C
WHERE A.MEMBER_ID = B.MEMBER_ID
     AND A.PRD_ID = C.PRD_ID ;
```

--스칼라 서브 쿼리 풀이 방법

```
SELECT A.ORDER_NO      AS 주문번호
      , A.MEMBER_ID    AS 회원ID
      , (
          SELECT MEMBER_NAME
            FROM TB_MEMBER
           WHERE MEMBER_ID = A.MEMBER_ID ) AS 주문자이름
      , A.PRD_ID       AS 상품ID
      , (
          SELECT PRD_NAME
            FROM TB_PRD
           WHERE PRD_ID = A.PRD_ID ) AS 상품명
FROM TB_ORDER A ;
```

출력결과

	주문번호	회원ID	주문자이름	상품ID	상품명
1	1	AAAAA	사용자A	P0002	에어컨
2	2	AAAAA	사용자A	P0003	세탁기
3	3	AAAAA	사용자A	P0004	건조기
4	4	AAAAA	사용자A	P0020	수건

2. 스칼라 서브쿼리 (사용 : SELECT)

스칼라 서브쿼리의 장단점

장점

- (1) **조인할 테이블의 개수를 줄여**서 이해하기 쉽게 만들 수 있다.
- (2) **캐싱 기능**을 이용해서 성능을 증가시킬 수 있다.

단점

- (1) SELECT 횟수만큼 실행하므로 **성능이 떨어질 수 있다.** (차라리 조인이 성능이 더 좋을 수 있음)
- (2) **하나의 컬럼, 하나의 값만 가져와야** 한다는 제약이 있다. (조인은 제약 없음)

3. 인라인 뷰 (사용 : FROM)

인라인 뷰 (Inline View)

- **FROM** 부분에서 사용되는 서브 쿼리
- FROM 절에 쿼리를 작성해 결과를 **가상의 테이블인 것 처럼** 사용할 수 있음

```
SELECT A.MEMBER_ID
      , A.MEMBER_NAME
      , A.GRADE_CD
      , A.AGE
      , B.등급별최고나이
FROM TB_MEMBER A
      , (
        SELECT GRADE_CD
              , MAX(AGE) AS 등급별최고나이
        FROM TB_MEMBER
        GROUP BY GRADE_CD
      ) B -- 등급별로 가장 나이가 많은 정보
WHERE A.GRADE_CD = B.GRADE_CD
      AND A.AGE = B.등급별최고나이;
```

서브쿼리

인라인 뷰를 이용해 가상의 테이블 처럼 결과를 생성



GRADE_CD	등급별최고나이
1	27
2	39
5	44
4	52
3	35

3. 인라인 뷰 (사용 : FROM)

인라인 뷰 (Inline View)

인라인 뷰 실행원리 이해하기

```
SELECT A.MEMBER_ID
      , A.MEMBER_NAME
      , A.GRADE_CD
      , A.AGE
      , B.등급별최고나이
FROM TB_MEMBER A
      , (
    SELECT GRADE_CD
          , MAX(AGE) AS 등급별최고나이
    FROM TB_MEMBER
    GROUP BY GRADE_CD
  ) B -- 등급별로 가장 나이가 많은 정보
WHERE A.GRADE_CD = B.GRADE_CD
      AND A.AGE = B.등급별최고나이;
```

TB_MEMBER

MEMBER_ID	MEMBER_NAME	GRADE_CD	AGE
CCCCC	사용자C	1	27
AAAAA	사용자A	1 (null)	
EEEEEE	사용자E	1 (null)	
BBBBBB	사용자B	2	25
GGGGG	사용자G	2	39
DDDDD	사용자D	3	30
FFFFF	사용자F	3	35
IIIII	사용자I	4	52
HHHHH	사용자H	5	44

인라인 뷰로 만든 가상 테이블

GRADE_CD	등급별최고나이
1	27
2	39
5	44
4	52
3	35

실행 결과

MEMBER_ID	MEMBER_NAME	GRADE_CD	AGE	등급별최고나이
CCCCC	사용자C	1	27	27
GGGGG	사용자G	2	39	39
HHHHH	사용자H	5	44	44
IIIII	사용자I	4	52	52
FFFFF	사용자F	3	35	35

4. 중첩서브쿼리 (사용 : WHERE)

중첩서브쿼리 (Nested Sub Query)

- **WHERE** 부분에서 사용되는 서브 쿼리
- 메인쿼리의 컬럼을 쓰냐 안쓰냐에 따라 상관서브쿼리 , 비상관서브쿼리로 나뉨

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX (PRD_PRICE)  
                    FROM TB_PRD  
                    WHERE PRD_TYPE = A.PRD_TYPE  
                  ) ;
```

상관서브쿼리 예시

실행 순서 : 메인 -> 서브

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX (PRD_PRICE)  
                    FROM TB_PRD  
                  ) ;
```

비상관서브쿼리 예시

실행 순서 : 서브 -> 메인

4. 중첩서브쿼리 (사용 : WHERE)

비상관 서브쿼리 (Sub query)

서브쿼리에서 메인쿼리의 컬럼을 사용하지 않는 서브쿼리를 의미합니다. (메인쿼리와 연관 X)

이 경우 서브쿼리가 먼저 실행되고 메인 쿼리가 실행됩니다.

(2)

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = 2000000;
```

```
SELECT *
```

```
FROM TB_PRD A
```

```
WHERE PRD_PRICE = ( SELECT MAX (PRD_PRICE)  
FROM TB_PRD  
);
```

(1)

MAX(PRD_PRICE)
2000000

4. 중첩서브쿼리 (사용 : WHERE)

비상관 서브쿼리 문제를 풀어봅시다

(1) TB_MEMBER 테이블에서 회원들 중에 **평균 AGE 기준 이상인** AGE 정보를 가진 대상만 출력해주세요.

MEMBER_NAME	GRADE_CD	AGE
사용자G	2	39
사용자H	5	44
사용자I	4	52

(2) TB_PRD 테이블에서 **가장 싼 가격**을 가진 상품 정보를 출력해주세요.

PRD_NAME	PRD_INFO	PRD_PRICE
수세미	욕실 청소용 수세미입니다.	5000
수건	호텔에서도 사용되는 고급소재의 친환경 타올 소재의 수건입니다.	5000

4. 중첩서브쿼리 (사용 : WHERE)

비상관 서브쿼리 문제를 풀어봅시다 (답)

(1) TB_MEMBER 테이블에서 회원들 중에 **평균 AGE 기준 이상인** AGE 정보를 가진 대상만 출력해주세요.

답

```
SELECT MEMBER_NAME
      , GRADE_CD
      , AGE
FROM TB_MEMBER
WHERE AGE >= ( SELECT AVG (AGE)
                FROM TB_MEMBER ) ;
```

MEMBER_NAME	GRADE_CD	AGE
사용자G	2	39
사용자H	5	44
사용자I	4	52

(2) TB_PRD 테이블에서 **가장 싼 가격**을 가진 상품 정보를 출력해주세요.

답

```
SELECT PRD_NAME
      , PRD_INFO
      , PRD_PRICE
FROM TB_PRD
WHERE PRD_PRICE = ( SELECT MIN (PRD_PRICE)
                    FROM TB_PRD ) ;
```

PRD_NAME	PRD_INFO	PRD_PRICE
수세미	욕실 청소용 수세미입니다.	5000
수건	호텔에서도 사용되는 고급소재의 친환경 타올 소재의 수건입니다.	5000

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 (Sub query)

서브쿼리에서 메인쿼리의 컬럼을 사용하는 서브쿼리를 의미합니다. (메인쿼리와 연관 O)

메인쿼리에서 출력되는 튜플(행) 수만큼 서브쿼리가 실행되고 **그 결과를 대입해 메인쿼리를 실행**합니다.

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX (PRD_PRICE)  
                    FROM TB_PRD  
                    WHERE PRD_TYPE = A.PRD_TYPE  
                  ) ;
```


서브쿼리에 메인쿼리 컬럼(A.PRD_TYPE)이 있으므로 먼저 실행되지 못함

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 실행 원리를 알아봅시다.

(1) 메인쿼리의 행 개수만큼 값을 반복 대입/비교하여 조건을 확인합니다.

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX(PRD_PRICE)  
                    FROM TB_PRD  
                    WHERE PRD_TYPE = A.PRD_TYPE  
                  ) ;
```



PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0001	헤어드라이기	가전	30000
P0002	에어컨	가전	1500000
P0003	세탁기	가전	600000
P0004	건조기	가전	800000
P0009	갤럭시s23	스마트폰	1500000
P0008	애플14	스마트폰	1200000
P0010	조아샴푸	욕실용품	20000
P0019	런스	욕실용품	20000
P0018	샤워기	욕실용품	50000
P0017	공팡이제거제	욕실용품	10000
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000
P0013	냄비	주방용품	30000
P0012	전기밥솥	주방용품	80000
P0011	주전자	주방용품	20000
P0014	칼	주방용품	15000
P0005	노트북	컴퓨터	1500000
P0006	데스크탑	컴퓨터	2000000
P0007	태블릿	컴퓨터	800000

따로 WHERE 조건이 없으므로

메인쿼리에서 출력될 행의 횟수만큼 대입/비교를 실행

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 실행 원리를 알아봅시다.

(2) 서브쿼리에 있는 조건에 맞춰 (A.PRD_TYPE) 값을 대입/비교합니다.

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX(PRD_PRICE)  
                    FROM TB_PRD  
                    WHERE PRD_TYPE = A.PRD_TYPE  
                  ) ;
```

가전

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0001	헤어드라이기	가전	30000
P0002	에어컨	가전	1500000
P0003	세탁기	가전	600000
P0004	건조기	가전	800000
P0009	갤럭시S23	스마트폰	1500000
P0008	애플14	스마트폰	1200000
P0010	조아샴푸	욕실용품	20000
P0019	런스	욕실용품	20000
P0018	샤워기	욕실용품	50000
P0017	곰팡이제거제	욕실용품	10000
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000
P0013	냄비	주방용품	30000
P0012	전기밥솥	주방용품	80000
P0011	주전자	주방용품	20000
P0014	칼	주방용품	15000
P0005	노트북	컴퓨터	1500000
P0006	데스크탑	컴퓨터	2000000
P0007	태블릿	컴퓨터	800000

X

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 실행 원리를 알아봅시다.

(2) 서브쿼리에 있는 조건에 맞춰 (A.PRД_TYPE) 값을 대입/비교합니다.

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX(PRD_PRICE)  
                    FROM TB_PRD  
                    WHERE PRD_TYPE = A.PRД_TYPE  
                  ) ;
```

가전

<출력결과>

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0002	에어컨	가전	1500000

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0001	헤어드라이기	가전	30000
P0002	에어컨	가전	1500000
P0003	세탁기	가전	600000
P0004	건조기	가전	800000
P0009	갤럭시S23	스마트폰	1500000
P0008	애플14	스마트폰	1200000
P0010	조아샴푸	욕실용품	20000
P0019	런스	욕실용품	20000
P0018	샤워기	욕실용품	50000
P0017	공팡이제거제	욕실용품	10000
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000
P0013	냄비	주방용품	30000
P0012	전기밥솥	주방용품	80000
P0011	주전자	주방용품	20000
P0014	칼	주방용품	15000
P0005	노트북	컴퓨터	1500000
P0006	데스크탑	컴퓨터	2000000
P0007	태블릿	컴퓨터	800000

X
O

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 실행 원리를 알아봅시다.

(2) 서브쿼리에 있는 조건에 맞춰 (A.PRД_TYPE) 값을 대입/비교합니다.

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX(PRD_PRICE)  
                    FROM TB_PRD  
                    WHERE PRD_TYPE = A.PRД_TYPE  
                  ) ;
```

가전

<출력결과>

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0002	에어컨	가전	1500000

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0001	헤어드라이기	가전	30000
P0002	에어컨	가전	1500000
P0003	세탁기	가전	600000
P0004	건조기	가전	800000
P0009	갤럭시S23	스마트폰	1500000
P0008	애플14	스마트폰	1200000
P0010	조아샴푸	욕실용품	20000
P0019	런스	욕실용품	20000
P0018	샤워기	욕실용품	50000
P0017	공팡이제거제	욕실용품	10000
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000
P0013	냄비	주방용품	30000
P0012	전기밥솥	주방용품	80000
P0011	주전자	주방용품	20000
P0014	칼	주방용품	15000
P0005	노트북	컴퓨터	1500000
P0006	데스크탑	컴퓨터	2000000
P0007	태블릿	컴퓨터	800000

X
O
X

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 실행 원리를 알아봅시다.

(2) 서브쿼리에 있는 조건에 맞춰 (A.PRД_TYPE) 값을 대입/비교합니다.

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX(PRD_PRICE)  
                    FROM TB_PRD  
                    WHERE PRD_TYPE = 스마트폰  
                    ) ;
```

<출력결과>

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0002	에어컨	가전	1500000
P0009	갤럭시S23	스마트폰	1500000

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0001	헤어드라이기	가전	30000
P0002	에어컨	가전	1500000
P0003	세탁기	가전	600000
P0004	건조기	가전	800000
P0009	갤럭시S23	스마트폰	1500000
P0008	애플14	스마트폰	1200000
P0010	조아샴푸	욕실용품	20000
P0019	런스	욕실용품	20000
P0018	샤워기	욕실용품	50000
P0017	공팡이제거제	욕실용품	10000
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000
P0013	냄비	주방용품	30000
P0012	전기밥솥	주방용품	80000
P0011	주전자	주방용품	20000
P0014	칼	주방용품	15000
P0005	노트북	컴퓨터	1500000
P0006	데스크탑	컴퓨터	2000000
P0007	태블릿	컴퓨터	800000

X
O
X
X
O
X

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 실행 원리를 알아봅시다.

(2) 서브쿼리에 있는 조건에 맞춰 (A.PRД_TYPE) 값을 대입/비교합니다.

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX(PRD_PRICE)  
                    FROM TB_PRD  
                    WHERE PRD_TYPE = A.PRД_TYPE  
                  ) ;
```

욕실용품

<출력결과>

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0002	에어컨	가전	1500000
P0009	갤럭시S23	스마트폰	1500000
P0018	샤워기	욕실용품	50000

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0001	헤어드라이기	가전	30000
P0002	에어컨	가전	1500000
P0003	세탁기	가전	600000
P0004	건조기	가전	800000
P0009	갤럭시S23	스마트폰	1500000
P0008	애플14	스마트폰	1200000
P0010	조아샴푸	욕실용품	20000
P0019	런스	욕실용품	20000
P0018	샤워기	욕실용품	50000
P0017	공팡이제거제	욕실용품	10000
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000
P0013	냄비	주방용품	30000
P0012	전기밥솥	주방용품	80000
P0011	주전자	주방용품	20000
P0014	칼	주방용품	15000
P0005	노트북	컴퓨터	1500000
P0006	데스크탑	컴퓨터	2000000
P0007	태블릿	컴퓨터	800000

X
O
X
X
O
X
X
O
X
O

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 실행 원리를 알아봅시다.

(2) 서브쿼리에 있는 조건에 맞춰 (A.PRД_TYPE) 값을 대입/비교합니다.

```
SELECT *  
FROM TB_PRD A  
WHERE PRD_PRICE = ( SELECT MAX(PRD_PRICE)  
                    FROM TB_PRD  
                    WHERE PRD_TYPE = A.PRД_TYPE  
                  ) ;
```

컴퓨터

<출력결과>

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0002	에어컨	가전	1500000
P0009	갤럭시S23	스마트폰	1500000
P0018	샤워기	욕실용품	50000
P0012	전기밥솥	주방용품	80000
P0006	데스크탑	컴퓨터	2000000

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0001	헤어드라이기	가전	30000
P0002	에어컨	가전	1500000
P0003	세탁기	가전	600000
P0004	건조기	가전	800000
P0009	갤럭시S23	스마트폰	1500000
P0008	애플14	스마트폰	1200000
P0010	조아샴푸	욕실용품	20000
P0019	런스	욕실용품	20000
P0018	샤워기	욕실용품	50000
P0017	공팡이제거제	욕실용품	10000
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000
P0013	냄비	주방용품	30000
P0012	전기밥솥	주방용품	80000
P0011	주전자	주방용품	20000
P0014	칼	주방용품	15000
P0005	노트북	컴퓨터	1500000
P0006	데스크탑	컴퓨터	2000000
P0007	태블릿	컴퓨터	800000

X
O
X
X
O
X
X
O
X
X
X
O
X
X
O
X
O
X

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 문제를 풀어봅시다

(3) TB_PRD 테이블에서 상품 타입 별로 가장 싼 가격의 상품 정보를 출력해주세요.

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0001	헤어드라이기	가전	30000
P0007	태블릿	컴퓨터	800000
P0008	애플14	스마트폰	1200000
P0014	칼	주방용품	15000
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000

4. 중첩서브쿼리 (사용 : WHERE)

상관 서브쿼리 문제를 풀어봅시다(답)

(3) TB_PRD 테이블에서 상품 타입 별로 가장 싼 가격의 상품 정보를 출력해주세요.

답

```
SELECT PRD_ID
      , PRD_NAME
      , PRD_TYPE
      , PRD_PRICE
FROM TB_PRD A
WHERE PRD_PRICE = ( SELECT MIN(PRD_PRICE)
                    FROM TB_PRD
                    WHERE PRD_TYPE = A.PRD_TYPE
                  ) ;
```

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0001	헤어드라이기	가전	30000
P0007	태블릿	컴퓨터	800000
P0008	애플14	스마트폰	1200000
P0014	칼	주방용품	15000
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000

4. 중첩서브쿼리 (사용 : WHERE) – 단일행 연산자 / 다중행 연산자

WHERE 절에서 서브 쿼리를 결과를 받을 때

단일행을 받을 수 있는 연산자와 다중행을 받을 수 있는 연산자가 있습니다.

```
SELECT *  
  FROM TB_MEMBER  
 WHERE GRADE_CD = ( SELECT GRADE_CD  
                   FROM TB_GRADE ) ;
```

```
SELECT *  
  FROM TB_MEMBER  
 WHERE GRADE_CD IN ( SELECT GRADE_CD  
                   FROM TB_GRADE ) ;
```

둘 중 오류가 나는 쿼리는 무엇일까요?

4. 중첩서브쿼리 (사용 : WHERE) – 단일행 연산자 / 다중행 연산자

```
SELECT *  
FROM TB_MEMBER  
WHERE GRADE_CD = ( SELECT GRADE_CD  
FROM TB_GRADE ) ;
```

ORA-01427: 단일 행 하위 질의에 2개 이상의 행이 리턴되었습니다.
01427, 00000 - "single-row subquery returns more than one row"

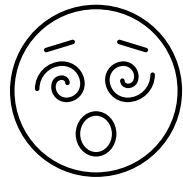
<- 오류

```
SELECT *  
FROM TB_MEMBER  
WHERE GRADE_CD IN ( SELECT GRADE_CD  
FROM TB_GRADE ) ;
```

<- 정상

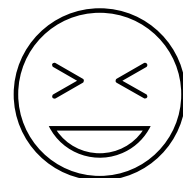
GRADE_CD
1
2
3
4
5

이중에서 뭐를 = 로 비교해야 하지?



GRADE_CD
1
2
3
4
5

GRADE_CD IN (1,2,3,4,5) 가능하지!



4. 중첩서브쿼리 (사용 : WHERE) – 단일행 연산자 / 다중행 연산자

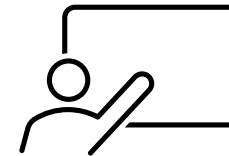
다중행을 받을 수 있는 연산자 : IN , ANY , ALL , EXISTS , NOT EXISTS 등

```
SELECT *  
  FROM TB_MEMBER  
 WHERE GRADE_CD IN ( SELECT GRADE_CD  
                     FROM TB_GRADE ) ;
```

단일행을 받을 수 있는 연산자 : = , >= , < , != 등

```
SELECT *  
  FROM TB_MEMBER  
 WHERE GRADE_CD = 1 ;
```

기존의 비교연산자들은
단일행 연산자입니다.



4. 중첩서브쿼리 (사용 : WHERE) – 단일행 연산자 / 다중행 연산자

다중행 연산자 예시를 들어봅시다.

(IN , ANY , ALL)

EXISTS 와 NOT EXISTS 는 이후에 다룹니다

4. 중첩서브쿼리 (사용 : WHERE) – 단일행 연산자 / 다중행 연산자

IN 은 입력된 다중 행에 대해 일치하는 값을 출력합니다.

```
SELECT PRD_ID
      , PRD_NAME
      , PRD_TYPE
      , PRD_PRICE
FROM TB_PRD A
WHERE PRD_PRICE IN ( SELECT MAX(PRD_PRICE)
                     FROM TB_PRD
                     GROUP BY PRD_TYPE
                     ) ;
```



MAX(PRD_PRICE)
2000000
80000
1500000
1500000
50000



```
SELECT PRD_ID
      , PRD_NAME
      , PRD_TYPE
      , PRD_PRICE
FROM TB_PRD A
WHERE PRD_PRICE IN ( 2000000,80000,1500000,1500000,50000 ) ;
```

4. 중첩서브쿼리 (사용 : WHERE) – 단일행 연산자 / 다중행 연산자

ANY 는 입력된 다중 행 중에서 **하나라도 일치하면** 출력합니다.

```
SELECT PRD_ID
      , PRD_NAME
      , PRD_TYPE
      , PRD_PRICE
FROM TB_PRD A
WHERE PRD_PRICE = ANY ( SELECT MAX(PRD_PRICE)
                        FROM TB_PRD
                        GROUP BY PRD_TYPE
                        ) ;
```



MAX(PRD_PRICE)
2000000
80000
1500000
1500000
50000



```
SELECT PRD_ID
      , PRD_NAME
      , PRD_TYPE
      , PRD_PRICE
FROM TB_PRD A
WHERE PRD_PRICE = ANY ( 2000000,80000,1500000,1500000,50000 ) ;
```

4. 중첩서브쿼리 (사용 : WHERE) – 단일행 연산자 / 다중행 연산자

ALL 은 입력된 다중 행에 대해 **조건이 모두 일치해야** 출력합니다.

```
SELECT PRD_ID
      , PRD_NAME
      , PRD_TYPE
      , PRD_PRICE
FROM TB_PRD A
WHERE PRD_PRICE >= ALL ( SELECT MAX(PRD_PRICE)
                        FROM TB_PRD
                        GROUP BY PRD_TYPE
                        ) ;
```



MAX(PRD_PRICE)
2000000
80000
1500000
1500000
50000



```
SELECT PRD_ID
      , PRD_NAME
      , PRD_TYPE
      , PRD_PRICE
FROM TB_PRD A
WHERE PRD_PRICE >= ALL ( 2000000,80000,1500000,1500000,50000 ) ;
```

문제를 풀어봅시다

1. 오른쪽 테이블에 대해서 다음 중 오류가 발생하는 쿼리는 무엇인가?

MEMBER_ID	AGE	GRADE_CD
AAAAA	(null)	1
BBBBB	25	2
CCCCC	27	1
DDDDD	30	3
EEEEE	(null)	1
FFFFF	35	3
GGGGG	39	2
HHHHH	44	5
IIIII	52	4

(1)

```
SELECT *  
FROM TB_MEMBER  
WHERE AGE = ( SELECT AGE  
               FROM TB_MEMBER  
               WHERE ROWNUM = 1  
             ) ;
```

(2)

```
SELECT *  
FROM TB_MEMBER  
WHERE AGE = ( SELECT MAX(AGE)  
               FROM TB_MEMBER  
             ) ;
```

(3)

```
SELECT *  
FROM TB_MEMBER  
WHERE AGE >= ( SELECT AVG(AGE)  
               FROM TB_MEMBER  
               GROUP BY GRADE_CD ) ;
```

(4)

```
SELECT *  
FROM TB_MEMBER  
WHERE AGE = ANY ( SELECT AVG(AGE)  
                  FROM TB_MEMBER  
                  GROUP BY GRADE_CD ) ;
```

서브쿼리 END