

ROWNUM

- 내장 함수는 아니지만 자주 사용되는 문법임.
- 오라클에서 내부적으로 생성되는 가상 컬럼으로 SQL 조회 결과의 순번을 나타냄.
- 자료를 일부분만 확인하여 처리할 때 유용함.

질의. 고객 목록에서 고객번호, 이름, 전화번호를 앞의 두 명만 보이시오.

```
SELECT    ROWNUM "순번", custid, name, phone
FROM      Customer
WHERE     ROWNUM <=2;
```

순번	CUSTID	NAME	PHONE
1	1	박지성	000-5000-0001
2	2	김연아	000-6000-0001

부속질의

- 스칼라 부속질의 – SELECT 부속질의
- 인라인 뷰 – FROM 부속질의
- 중첩질의 – WHERE 부속질의

부속질의

■ 부속질의(subquery)란?

- 하나의 SQL 문 안에 다른 SQL 문이 중첩된 nested 질의를 말함.
- 다른 테이블에서 가져온 데이터로 현재 테이블에 있는 정보를 찾거나 가공할 때 사용함.
- 보통 데이터가 대량일 때 데이터를 모두 합쳐서 연산하는 조인보다 필요한 데이터만 찾아서 공급해주는 부속질의가 성능이 더 좋음.
- 주질의(main query, 외부질의)와 부속질의(sub query, 내부질의)로 구성됨.

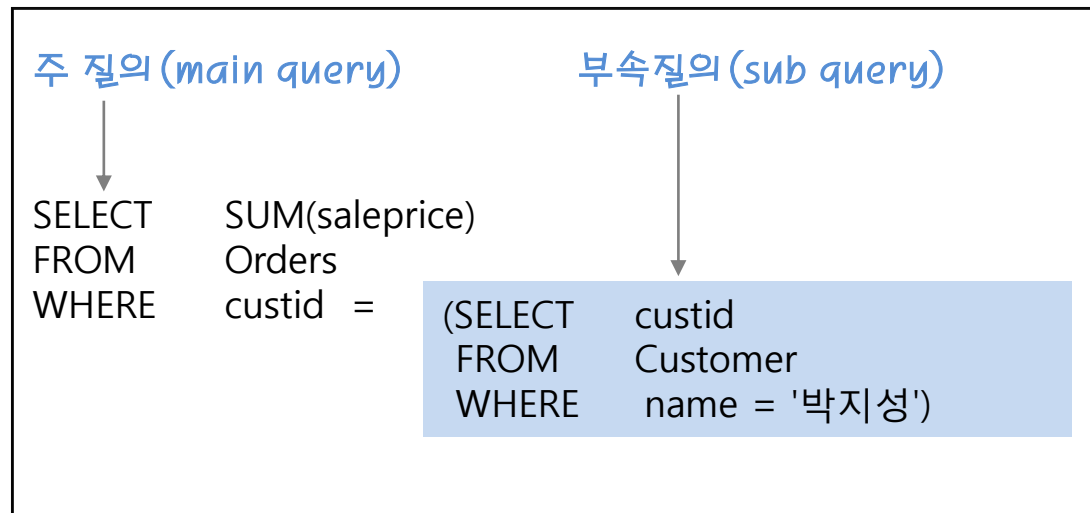


그림 4-2 부속질의

02 부속질의

부속질의의 종류

명칭	위치	영문 및 동의어	설명
스칼라 부속질의	SELECT 절	scalar subquery	SELECT 절에서 사용되며 단일 값을 반환하기 때문에 스칼라 부속질의라고 함.
인라인 뷰	FROM 절	inline view, table subquery	FROM 절에서 결과를 뷰(view) 형태로 반환하기 때문에 인라인 뷰라고 함.
중첩질의	WHERE 절	nested subquery, predicate subquery	WHERE 절에 술어와 같이 사용되며 결과를 한정시키기 위해 사용됨. 상관 혹은 비상관 형태.

2.1 스칼라 부속질의 – SELECT 부속질의

■ 스칼라 부속질의(scalar subquery)란?

- SELECT 절에서 사용되는 부속질의로, 부속질의의 결과 값을 단일 행, 단일 열의 스칼라 값으로 반환함.
- 스칼라 부속질의는 원칙적으로 스칼라 값이 들어갈 수 있는 모든 곳에 사용 가능하며, 일반적으로 SELECT 문과 UPDATE SET 절에 사용됨.
- 주질의와 부속질의와의 관계는 상관/비상관 모두 가능함.

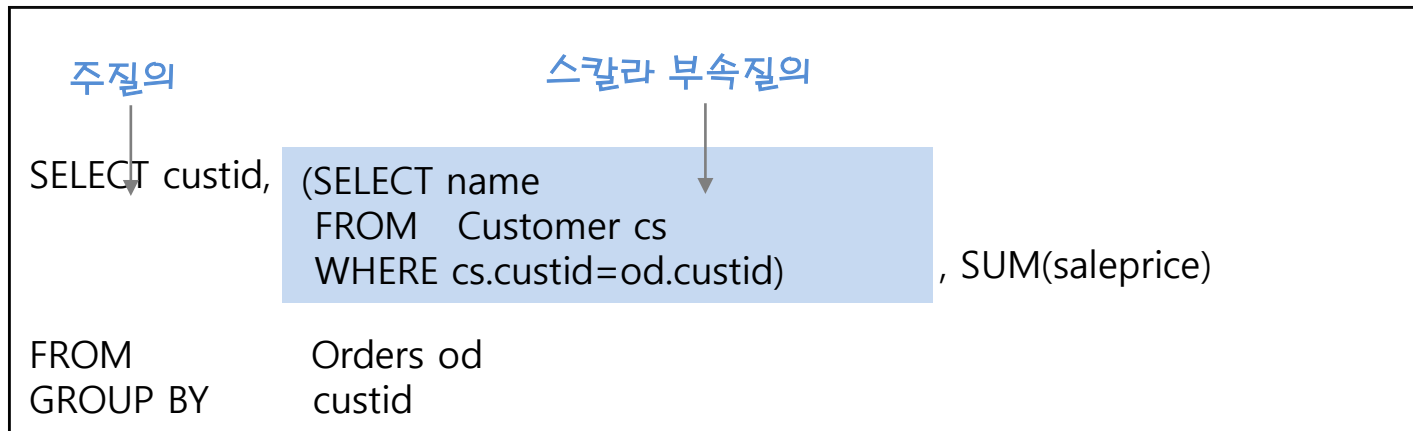


그림 4-3 스칼라 부속질의

2.1 스칼라 부속질의 - SELECT 부속질의

질의. 마당서점의 고객별 판매액을 보이시오(결과는 고객이름과 고객별 판매액을 출력).

```
SELECT      ( SELECT name
                FROM Customer cs
                WHERE cs.custid=od.custid ) "name",
            SUM(saleprice) "total"
FROM        Orders od
GROUP BY    od.custid;
```

⚡ name	⚡ total
박지성	39000
김연아	15000
추신수	33000
장미란	31000

2.1 스칼라 부속질의 – SELECT 부속질의

```
SELECT    custid,  
          SUM(saleprice) "total"  
FROM      Orders od  
GROUP BY custid ;
```



CUSTID	total
1	39000
2	15000
4	33000
3	31000

```
SELECT    name  
FROM      Customer cs  
WHERE     cs.custid = od.custid
```

custid 1의 이름은?



```
SELECT    (SELECT    name  
            FROM      Customer cs  
            WHERE     cs.custid = od.custid "name",  
          SUM(saleprice) "total"  
FROM      Orders od  
GROUP BY od.custid ;
```



name	total
박지성	39000
김연아	15000
추신수	33000
장미란	31000

그림 4-4 마당서점의 고객별 판매액

2.1 스칼라 부속질의 - SELECT 부속질의

질의 . Orders 테이블에 각 주문에 맞는 도서이름을 입력하시오.

```
UPDATE Orders
SET bookname = ( SELECT bookname
                  FROM Book
                  WHERE Book.bookid=Orders.bookid );
```

ORDERID	CUSTID	BOOKID	SALEPRICE	ORDERDATE	BOOKNAME
1	1	1	6000	14/07/01	축구의 역사
2	1	3	21000	14/07/03	축구의 이해
3	2	5	8000	14/07/03	피겨 교본
4	3	6	6000	14/07/04	역도 단계별기술
5	4	7	20000	14/07/05	야구의 추억
6	1	2	12000	14/07/07	축구하는 여자
7	4	8	13000	14/07/07	야구를 부탁해
8	3	10	12000	14/07/08	Olympic Champions
9	2	10	7000	14/07/09	Olympic Champions
10	3	8	13000	14/07/10	야구를 부탁해

2.2 인라인 뷰- FROM 부속질의

■ 인라인 뷰(inline view)란?

- FROM 절에서 사용되는 부속질의.
- 테이블 이름 대신 인라인 뷰 부속질의를 사용하면 보통의 테이블과 같은 형태로 사용할 수 있음.
- 부속질의 결과 반환되는 데이터는 다중 행, 다중 열이어도 상관없음.
- 다만 가상의 테이블인 뷰 형태로 제공되어 상관 부속질의로 사용될 수는 없음.

질의 고객번호가 2 이하인 고객의 판매액을 보이시오
(결과는 고객이름과 고객별 판매액 출력)

```
SELECT  cs.name, SUM(od.saleprice) "total"
FROM    (SELECT custid, name
         FROM Customer
         WHERE custid <= 2) cs,
         Orders od
WHERE    cs.custid=od.custid
GROUP BY cs.name;
```

NAME	total
박지성	39000
김연아	15000

2.2 인라인 뷰- FROM 부속질의

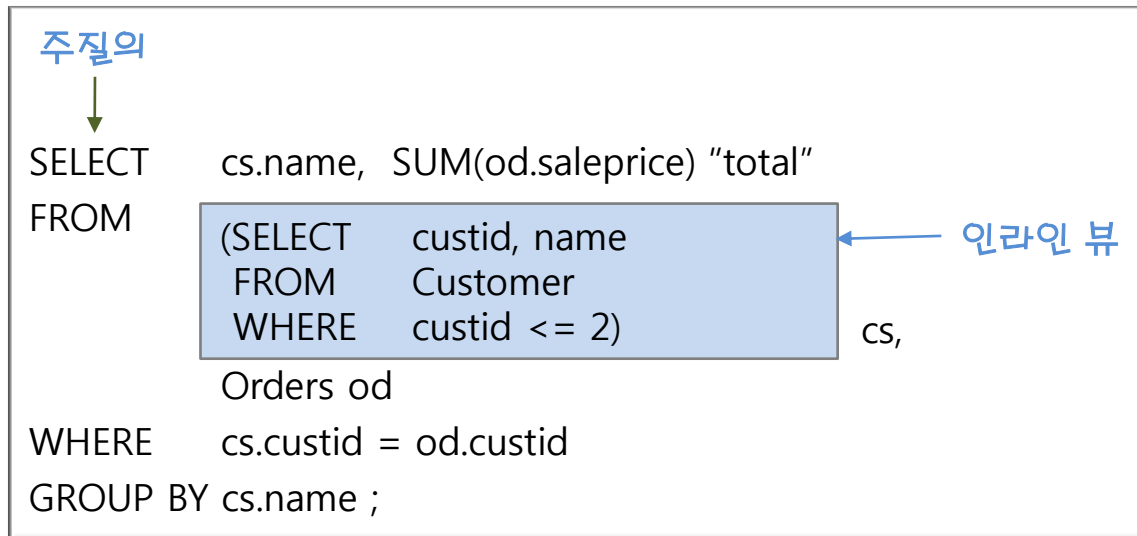


그림 4-5 인라인 뷰

2.3 중첩질의 – WHERE 부속질의

- 중첩질의(nested subquery)는 WHERE 절에서 사용되는 부속질의.
- WHERE 절은 보통 데이터를 선택하는 조건 혹은 술어(predicate)와 같이 사용됨. 그래서 중첩질의를 술어 부속질의(predicate subquery)라고도 함.

중첩질의 연산자의 종류

술어	연산자	반환 행	반환 열	상관
비교	=, >, <, >=, <=, < >	단일	단일	가능
집합	IN, NOT IN	다중	단일	가능
한정(quantified)	ALL, SOME(ANY)	다중	단일	가능
존재	EXISTS, NOT EXISTS	다중	다중	필수

2.3 중첩질의 – WHERE 부속질의

■ 비교 연산자

부속질의가 반드시 단일 행, 단일 열을 반환해야 하며, 아닐 경우 질의를 처리할 수 없음.

질의. 평균 주문금액 이하의 주문에 대해서 주문번호와 금액을 보이시오.

```
SELECT   orderid, saleprice
FROM      Orders
WHERE     saleprice <= (SELECT AVG(saleprice)
                        FROM Orders);
```

ORDERID	SALEPRICE
1	6000
3	8000
4	6000
9	7000

질의. 각 고객의 평균 주문금액보다 큰 금액의 주문 내역에 대해서 주문번호, 고객번호, 금액을 보이시오.

```
SELECT    orderid, custid, saleprice
FROM      Orders od
WHERE     saleprice > (SELECT AVG(saleprice)
                        FROM Orders so
                        WHERE od.custid=so.custid);
```

ORDERID	CUSTID	SALEPRICE
2	1	21000
3	2	8000
5	4	20000
8	3	12000
10	3	13000

2.3 중첩질의 – WHERE 부속질의

■ IN, NOT IN

IN 연산자는 주질의 속성 값이 부속질의에서 제공한 결과 집합에 있는지 확인(check)하는 역할을 함. IN 연산자는 부속질의의 결과 다중 행을 가질 수 있음. 주질의는 WHERE 절에 사용되는 속성 값을 부속질의의 결과 집합과 비교해 하나라도 있으면 참이 된다. NOT IN은 이와 반대로 값이 존재하지 않으면 참이 됨.

질의. 대한민국에 거주하는 고객에게 판매한 도서의 총판매액을 구하시오.

```
SELECT    SUM(saleprice) "total"
FROM      Orders
WHERE     custid IN (SELECT custid
                     FROM Customer
                     WHERE address LIKE '%대한민국%');
```

total
46000

2.3 중첩질의 – WHERE 부속질의

■ ALL, SOME(ANY)

- ALL은 모두, SOME(ANY)은 어떠한(최소한 하나라도)이라는 의미를 가짐.
- 구문 구조

scalar_expression {비교연산자(= | < | > | != | > | >= | !=> | < | <= | !=<)}
{ALL | SOME | ANY} (부속질의)

질의. 3번 고객이 주문한 도서의 최고 금액보다 더 비싼 도서를 구입한 주문의 주문번호와 금액을 보이시오.

```
SELECT   orderid, saleprice
FROM      Orders
WHERE     saleprice > ALL (SELECT saleprice
                           FROM    Orders
                           WHERE   custid='3');
```

ORDERID	SALEPRICE
5	20000
2	21000

2.3 중첩질의 – WHERE 부속질의

■ EXISTS, NOT EXISTS

- 데이터의 존재 유무를 확인하는 연산자
- 주질의에서 부속질의로 제공된 속성의 값을 가지고 부속질의에 조건을 만족하여 값이 존재하면 참이 되고, 주질의는 해당 행의 데이터를 출력함.
- NOT EXISTS의 경우 이와 반대로 동작함.
- 구문 구조

WHERE [NOT] EXISTS (부속질의)

질의. EXISTS 연산자로 대한민국에 거주하는 고객에게 판매한 도서의 총 판매액을 구하시오.

```
SELECT    SUM(saleprice) "total"
FROM      Orders od
WHERE     EXISTS (SELECT *
                  FROM Customer cs
                  WHERE address LIKE '%대한민국%' AND cs.custid=od.custid);
```

total
46000