

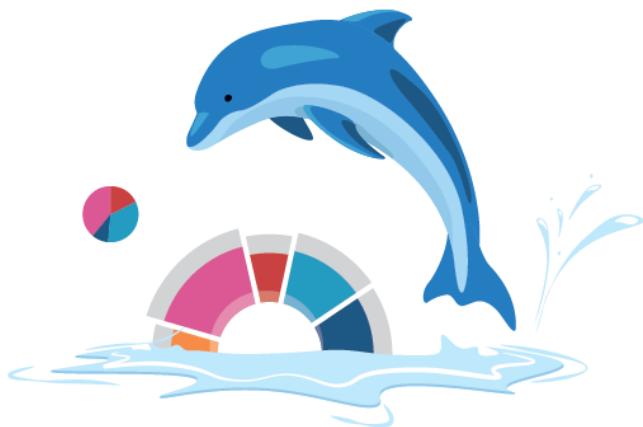
난생처음

MySQL



Chapter 06

서브쿼리



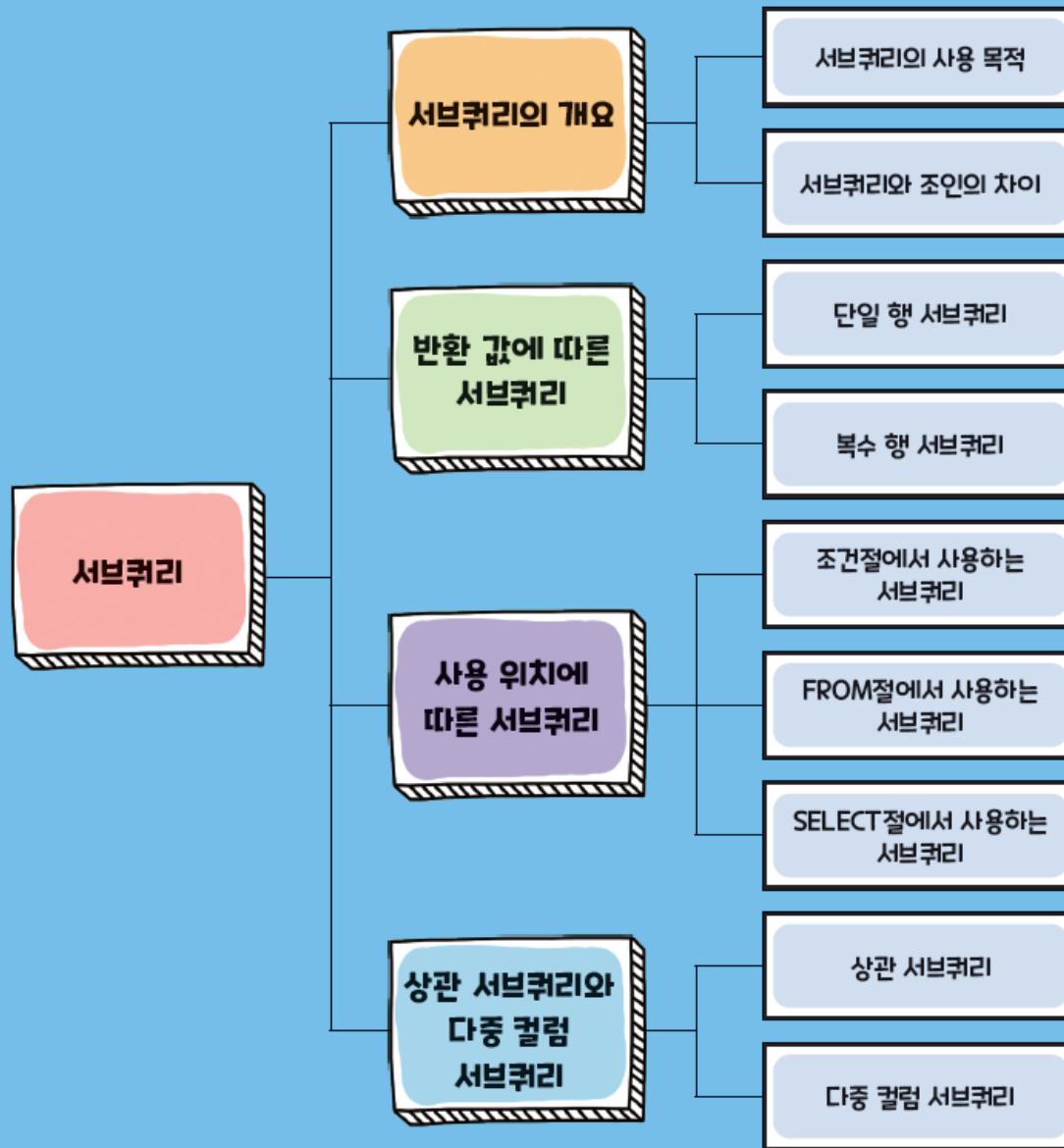
목차

1. 서브쿼리의 개요
2. 반환 값에 따른 서브쿼리
3. 사용 위치에 따른 서브쿼리
4. 상관 서브쿼리와 다중 컬럼 서브쿼리

학습목표

- 서브쿼리에 대해 설명할 수 있습니다.
- 서브쿼리의 종류를 이해하고, 작성할 수 있습니다.

Preview



Section 01

서브쿼리의 개요

1. 서브쿼리의 사용 목적

■ 서브쿼리(SubQuery)

- SQL문 내부에서 사용하는 SELECT문
- 서브쿼리는 복잡한 데이터 추출 및 조작 작업에 유용하며, SQL문의 유연성과 기능 확장에 도움됨
- 서브쿼리는 괄호 안에 기술해야 하며, WHERE절, SELECT절, FROM절, HAVING절 등에 넣어 사용할 수 있음
- 일반적으로 서브쿼리가 먼저 실행된 후 메인 쿼리가 실행되는 순서로 작업이 이루어짐

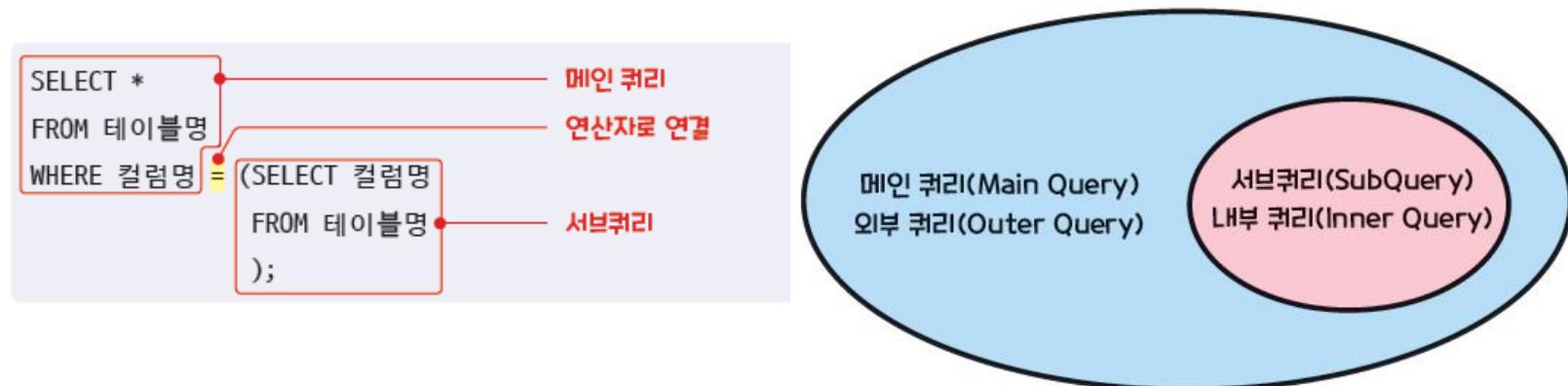


그림 6-1 서브쿼리의 구조

1. 서브쿼리의 사용 목적

■ 서브쿼리의 사용 목적

- 데이터 필터링
- 집계 및 계산
- 비교 및 검증
- 중첩된 데이터 추출
- EXISTS 절

2. 서브쿼리와 조인의 차이점

■ 서브쿼리와 조인의 차이점

- 데이터를 추출하는 접근 방식이 서로 다름
 - ✓ 서브쿼리 : 한 문장 안에 1개의 메인 쿼리와 1개 이상의 서브쿼리가 있는 구조
 - ✓ 조인 : 1개의 쿼리문 안에 2개 이상의 테이블을 연결한 후 필요한 컬럼을 조회
- 서브쿼리
 - ✓ 작은 데이터셋이나 간단한 필터링 작업에 효율적
 - ✓ 간단한 쿼리에 적합
 - ✓ 데이터 필터링, 계산에 적합
- 조인
 - ✓ 큰 테이블이면서 적절한 인덱스가 있는 경우에 효율적
 - ✓ 복잡한 쿼리에 적합
 - ✓ 여러 테이블 간의 관계 처리에 적합

Section 02

반환 값에 따른 서브쿼리

1. 단일 행 서브쿼리

■ 단일 행 서브쿼리(Single-Row SubQuery)

- 서브쿼리의 결과로 단일 행을 반환함
- =, <, <=, >, >=, <> 등의 단일 행 비교 연산자를 사용하여 메인 쿼리와 서브 쿼리를 연결할 수 있음

■ [예제 6-1] 최고 마일리지를 보유한 고객의 정보를 보이시오.

```
SELECT 고객번호  
      ,고객회사명  
      ,담당자명  
      ,마일리지  
FROM 고객  
WHERE 마일리지 = (SELECT MAX(마일리지)  
                      FROM 고객  
                     );
```

▶ 실행 결과

고객번호	고객회사명	담당자명	마일리지
ATRAN	대정 인터내셔널	배순용	128790

1. 단일 행 서브쿼리

■ [예제 6-2] 주문번호 'H0250'을 주문한 고객에 대해 고객회사명과 담당자명을 보이시오.

- 서브쿼리 사용

```
SELECT 고객회사명  
      ,담당자명  
FROM 고객  
WHERE 고객번호 = (SELECT 고객번호  
                     FROM 주문  
                     WHERE 주문번호 = 'H0250'  
                   );
```

- 조인 사용

```
SELECT 고객회사명  
      ,담당자명  
FROM 고객  
INNER JOIN 주문  
ON 고객.고객번호 = 주문.고객번호  
WHERE 주문번호 = 'H0250';
```

▶ 실행결과

고객회사명	담당자명
청운유통	손창민

1. 단일 행 서브쿼리

■ [예제 6-3] '부산광역시'고객의 최소 마일리지보다 더 큰 마일리지를 가진 고객 정보를 보이시오

```
SELECT 담당자명  
      ,고객회사명  
      ,마일리지  
FROM 고객  
WHERE 마일리지 > (SELECT MIN(마일리지)  
                      FROM 고객  
                     WHERE 도시 = '부산광역시'  
                      );
```

▶ 실행결과

담당자명	고객회사명	마일리지
이은광	굿모닝서울	15911
김성민	씨엔그룹	8788
최지수	오리안무역	6119

2. 복수 행 서브쿼리

복수 행 서브쿼리(Multi-Row SubQuery)

- 서브쿼리의 결과가 여러 행이 나오는 쿼리
- IN, ALL, ANY, SOME, EXISTS와 같은 복수 행 비교 연산자를 사용하여 메인 쿼리와 서브쿼리를 연결함

표 6-1 복수 행 비교 연산자

연산자	설명
IN	메인 쿼리의 비교 조건이 서브쿼리 결과 중 일치하는 것이 하나라도 있으면 참이다. 서브쿼리의 각 결괏값을 = 연산자로 비교한다.
ANY, SOME	서브쿼리의 각 결괏값을 >, <, >=, <= 등의 비교 연산자로 비교하여 하나 이상 일치하면 참이다. 즉, 서브쿼리의 최소 결괏값과 비교한다.
ALL	서브쿼리의 각 결괏값을 >, <, >=, <= 등의 비교 연산자로 비교하여 모두 일치하면 참이다. 즉, 서브쿼리의 최대 결괏값과 비교한다.
EXISTS	서브쿼리에 비교 조건을 만족하는 결과가 존재하면 참이다. 컬럼으로 비교하지 않고 행의 존재 여부로 비교하기 때문에 EXISTS 연산자 앞에는 컬럼명을 넣지 않는다.

2. 복수 행 서브쿼리

■ [예제 6-4] '부산광역시' 고객이 주문한 주문건수를 보이시오.

```
SELECT COUNT(*) AS 주문건수  
FROM 주문  
WHERE 고객번호 IN (SELECT 고객번호  
                     FROM 고객  
                     WHERE 도시 = '부산광역시'  
                     );
```

▶ 실행결과

주문건수

51

- IN은 서브쿼리의 각 결과마다 = 연산자를 사용하여 비교함

고객번호 = 'LIDBO'
OR
고객번호 = 'LLIWE'
OR
고객번호 = 'NGOHU'
OR
고객번호 = 'RICPE'
...
란 의미이니 서브쿼리에서
IN을 사용하면 되겠군!



'부산광역시' 고객의 고객번호

고객번호
LIDBO
LLIWE
NGOHU
RICPE
TTMBO

그림 6-2 복수 행 비교 연산자 IN

2. 복수 행 서브쿼리

■ [예제 6-5] '부산광역시' 전체 고객의 마일리지보다 마일리지가 큰 고객의 정보를 보이시오.

```
SELECT 담당자명  
      ,고객회사명  
      ,마일리지  
FROM 고객  
WHERE 마일리지 > ANY (SELECT 마일리지  
                          FROM 고객  
                          WHERE 도시 = '부산광역시'  
                          );
```

▶ 실행결과

담당자명	고객회사명	마일리지
이은광	굿모닝서울	15911
김성민	씨엔그룹	8788
최지수	오리안무역	6119
배순율	대정 인터내셔널	128790
박지혜	제원인테내쇼날	47865

- ANY를 사용하여 서브쿼리의 각 결과에 대해 > 연산자를 사용하여 비교한 후, 하나 이상 만족하는 고객의 정보를 반환함

마일리지
2819
806
7795
1177
7329

(a) 부산광역시 고객의 마일리지

고객회사명	마일리지	비교	결과
굿모닝서울	15911	15911>2819 OR 15911>806 OR 15911>7795 OR 15911>1177 OR 15911>7329	TRUE
엘케이 상사	406	406>2819 OR 406>806 OR 406>7795 OR 406>1177 OR 406>7329	FALSE
씨엔그룹	8788	8788>2819 OR 8788>806 OR 8788>7795 OR 8788>1177 OR 8788>7329	TRUE
오리안무역	6119	6119>2819 OR 6119>806 OR 6119>7795 OR 6119>1177 OR 6119>7329	TRUE

(b) 고객 테이블의 각 고객 마일리지에 대하여 부산광역시 고객의 마일리지와 각각 비교한 후 결과 반환

그림 6-3 ANY를 사용한 서브쿼리의 처리 방식

2. 복수 행 서브쿼리

- [예제 6-6] 각 지역의 어느 평균 마일리지보다도 마일리지가 큰 고객의 정보를 보이시오.

```
SELECT 담당자명  
      ,고객회사명  
      ,마일리지  
  FROM 고객  
 WHERE 마일리지 > ALL (SELECT AVG(마일리지)  
                           FROM 고객  
                          GROUP BY 지역  
                         );
```

▶ 실행결과

담당자명	고객회사명	마일리지
이은광	굿모닝서울	15911
배순율	대정 인터내셔널	128790
박지혜	제원인터내쇼날	47865
홍소현	그린로더스	81479
조호주	대화유피스	78001
지승관	조흥식품	22260
조가영	태흥푸드	36960
한현구	활인터넷프라이시스	116898
박건희	탑푸드코리아	57746

2. 복수 행 서브쿼리

■ [예제 6-6] 각 지역의 어느 평균 마일리지보다도 마일리지가 큰 고객의 정보를 보이시오.

- ALL은 서브쿼리의 결과로 나온 각 지역의 평균 마일리지 중 최대 값보다 큰 마일리지를 가진 고객의 정보를 반환함

지역	AVG(마일리지)
	11025.5152
경기도	2290.7059
경상남도	8758.0000
경상북도	7065.0000
전라북도	943.0000
강원도	321.5000

(a) 지역별 평균 마일리지

고객회사명	마일리지	비교	결과
굿모닝서울	15911	15911 > 11025 AND 15911 > 2290 AND 15911 > 8758 AND 15911 > 7065 ...	TRUE
엘케이 상사	406	406 > 11025 AND 406 > 2290 AND 406 > 8758 AND 406 > 7065 ...	FALSE
씨엔그룹	8788	8788 > 11025 AND 8788 > 2290 AND 8788 > 8758 AND 8788 > 7065 ...	FALSE
오리안무역	6119	6119 > 11025 AND 6119 > 2290 AND 6119 > 8758 AND 6119 > 7065 ...	FALSE

(b) 고객 테이블의 각 고객 마일리지에 대하여 지역별 평균 마일리지를 각각 비교한 후 결과 반환

그림 6-4 ALL을 사용한 서브쿼리의 처리 방식

2. 복수 행 서브쿼리

■ [예제 6-7] 한 번이라도 주문한 적이 있는 고객의 정보를 보이시오.

```
SELECT 고객번호  
      ,고객회사명  
FROM 고객  
WHERE EXISTS (SELECT *  
               FROM 주문  
              WHERE 고객번호 = 고객.고객번호  
                );
```

▶ 실행결과

고객번호	고객회사명
ACDDR	굿모닝서울
ADHTR	엘케이 상사
AIHTR	씨엔그룹
ANKFR	오리안무역
ANRFR	남해종합식품
ANSFR	서보엠엔에프
ATRAN	대정 인터내셔널
	제워이드

- IN 사용

```
SELECT 고객번호  
      ,고객회사명  
FROM 고객  
WHERE 고객번호 IN (SELECT DISTINCT 고객번호  
                     FROM 주문  
                     );
```

2. 복수 행 서브쿼리

■ [예제 6-7] 한 번이라도 주문한 적이 있는 고객의 정보를 보이시오.

- 조인 사용

```
SELECT DISTINCT 고객.고객번호  
          ,고객회사명  
FROM 고객  
INNER JOIN 주문  
ON 고객.고객번호 = 주문.고객번호;
```

▶ 실행결과

고객번호	고객회사명
ACDDR	굿모닝서울
ADHTR	엘케이 상사
AIHTR	씨엔그룹
ANKFR	오리안무역
ANRFR	남해종합식품
ANSFR	서보암엔에프
ATRAN	대정 인터내셔널
	제원아이엔씨

2. 복수 행 서브쿼리

확인문제

다음 보기의 SQL 중에서 동일한 결과를 보이는 SQL문끼리 연결된 것을 2개 고르시오.

(A)

```
SELECT COUNT(*)
FROM 고객
WHERE 마일리지 > ANY (SELECT 마일리지
                           FROM 고객
                           WHERE 도시 = '부산광역시');
```

(B)

```
SELECT COUNT(*)
FROM 고객
WHERE 마일리지 > ALL (SELECT 마일리지
                           FROM 고객
                           WHERE 도시 = '부산광역시');
```

(C)

```
SELECT COUNT(*)
FROM 고객
WHERE 마일리지 > (SELECT MAX(마일리지)
                           FROM 고객
                           WHERE 도시 = '부산광역시');
```

(D)

```
SELECT COUNT(*)
FROM 고객
WHERE 마일리지 > (SELECT MIN(마일리지)
                           FROM 고객
                           WHERE 도시 = '부산광역시');
```

① (A) – (D)

② (A) – (C)

③ (B) – (D)

④ (B) – (C)

정답

①, ④

Section 03

사용 위치에 따른 서브쿼리

1. 조건절에서 사용하는 서브쿼리

- [예제 6-8] 고객 전체의 평균마일리지보다 평균마일리지가 큰 도시에 대해 도시명과 도시의 평균마일리지를 보이시오.

```
SELECT 도시
      ,AVG(마일리지) AS 평균마일리지
  FROM 고객
 GROUP BY 도시
 HAVING AVG(마일리지) > (SELECT AVG(마일리지)
                               FROM 고객
                             );

```

▶ 실행결과

도시	평균마일리지
서울특별시	11137.0800
김해시	8758.0000
인천광역시	13088.0000
대전광역시	14822.0000
제주시	8838.0000

2. FROM절에서 사용하는 서브쿼리

■ 인라인 뷰(Inline View)

- FROM절에서도 서브쿼리를 사용할 수 있는데, 이를 인라인 뷰라고 함
- 뷰에는 반드시 별명을 지정해주어야 함
- 인라인 뷰의 별명은 테이블명처럼 사용할 수 있음

2. FROM절에서 사용하는 서브쿼리

■ [예제 6-9] 담당자명, 고객회사명, 마일리지, 도시, 해당 도시의 평균마일리지를 보이시오. 그리고 고객이 위치하는 도시의 평균마일리지와 각 고객의 마일리지 간의 차이도 함께 보이시오.

```
SELECT 담당자명
      ,고객회사명
      ,마일리지
      ,고객.도시 ● ①
      ,도시_평균마일리지
      ,도시_평균마일리지 - 마일리지 AS 차이
FROM 고객
     ,(SELECT 도시
           ,AVG(마일리지) AS 도시_평균마일리지
      FROM 고객
     GROUP BY 도시
    ) AS 도시별요약 ● 별명
WHERE 고객.도시 = 도시별요약.도시;
```

▶ 실행결과

담당자명	고객회사명	마일리지	도시	도시_평균마일리지	차이
이은광	굿모닝서울	15911	서울특별시	11137.0800	-4773.9200 ● 11,137-15,911
김병현	엘케이 상사	406	서울특별시	11137.0800	10731.0800 ● 11,137-406
김성민	씨언그룹	8788	광명시	4587.5000	-4200.5000
최지수	오리안무역	6119	서울특별시	11137.0800	5018.0800
강준	남희종합식품	106	구리시	106.0000	0.0000
	서보액에어		서울특별시		10408.0000

3. SELECT절에서 사용하는 서브쿼리

■ 스칼라 서브쿼리(Scalar SubQuery)

- SELECT절 내에서도 서브쿼리를 사용할 수 있음. 이때 서브쿼리가 하나의 값을 생성하는 형태를 스칼라 서브쿼리라고 함
- 쿼리를 실행하여 반환되는 값은 메인 쿼리에서 사용됨
- 만약 서브쿼리의 결과로 행을 0개 반환한다면 메인 쿼리의 결과는 NULL이고, 행이 2개 이상 반환되면 오류가 발생함

3. SELECT절에서 사용하는 서브쿼리

■ [예제 6-10] 고객번호, 담당자명과 고객의 최종 주문일을 보이시오

```
SELECT 고객번호
      ,담당자명
      ,(SELECT MAX(주문일)
        FROM 주문
       WHERE 주문.고객번호 = 고객.고객번호
      ) AS 최종주문일
     FROM 고객;
```

▶ 실행결과

고객번호	담당자명	최종주문일
ACDDR	이은광	2022-01-11
ADHTR	김병현	2021-09-27
AIHTR	김성민	2021-09-16
ANKFR	최지수	2021-12-17
AMNCO	강준	2021-12-01

4. CTE

■ CTE(Common Table Expression)

- 쿼리로 만든 임시 데이터셋으로 WITH절에서 정의함
- 인라인 뷰와 마찬가지로 파생 테이블(Derived Table)처럼 사용할 수 있음
- 하나의 쿼리를 논리적인 블록으로 나눌 수 있어 가독성이 좋음
- 쿼리 내에서 여러 번 참조할 수 있기 때문에 재사용성이 좋음
- 형식

```
WITH CTE명 AS  
(  
    SELECT문  
)
```

4. CTE

■ [예제 6-11] [예제 6-9]를 CTE를 사용하여 작성하시오.

```
WITH 도시별요약 AS
(
    SELECT 도시
        ,AVG(마일리지) AS 도시_평균마일리지
    FROM 고객
    GROUP BY 도시
)

SELECT 담당자명
    ,고객회사명
    ,마일리지
    ,고객.도시
    ,도시_평균마일리지
    ,도시_평균마일리지 - 마일리지 AS 차이
FROM 고객
    ,도시별요약
WHERE 고객.도시 = 도시별요약.도시;
```

4. CTE

확인문제

다음 내용이 맞으면 ○, 틀리면 ×를 표시하시오.

1. 서브쿼리는 WHERE절에서만 작성이 가능하다. ()
2. 메인 쿼리의 SELECT절 내에 SELECT문을 작성할 수 있다. ()

정답

1. × 2. ○

Section 04

상관 서브쿼리와
다중 컬럼 서브쿼리

1. 상관 서브쿼리

■ 상관 서브쿼리(Correlated SubQuery)

- 메인 쿼리와 서브쿼리 간의 상관관계를 포함하는 형태의 쿼리
- 상관 서브쿼리는 메인 쿼리를 한 행씩 처리함
- 이때 연관이 있는 컬럼을 기준으로 서브쿼리에서 값을 찾음
- 이로 인해 데이터가 많아지면 속도가 느려질 수 있으니 주의해야 함
- 상관 서브쿼리는 SELECT절, FROM절, WHERE절 등에서 사용할 수 있음

■ [예제 6-12] 사원 테이블에서 사원번호, 사원의 이름, 상사의 사원번호, 상사의 이름을 보이시오.

```
SELECT 사원번호  
      ,이름  
      ,상사번호  
      ,(SELECT 이름  
          FROM 사원 AS 상사  
         WHERE 상사.사원번호 = 사원.상사번호  
      ) AS 상사이름  
      FROM 사원;
```

▶ 실행결과

사원번호	이름	상사번호	상사이름
E01	이소미	E06	이현진
E02	배재용		HULL
E03	유대현	E06	이현진
E04	최소민	E06	이현진
E05	안주홍	E09	유가을
	이혁진		안주홍

2. 다중 컬럼 서브쿼리

■ 다중 컬럼 서브쿼리(Multi-Column SubQuery)

- 서브쿼리에서 여러 개의 컬럼을 사용하여 다중 비교를 하는 쿼리
- 다시 말해 서브쿼리의 결과로 나오는 여러 컬럼을 메인 쿼리의 값과 비교하여 결과를 출력함

■ [예제 6-13] 각 도시마다 최고 마일리지를 보유한 고객의 정보를 보이시오

```
SELECT 도시
      ,담당자명
      ,고객회사명
      ,마일리지
  FROM 고객
 WHERE (도시, 마일리지) IN (SELECT 도시, MAX(마일리지)
                               FROM 고객
                               GROUP BY 도시);
```

순서와 개수가 같아야 함

▶ 실행결과

도시	담당자명	고객회사명	마일리지
광명시	김성민	씨엔그룹	8788
구리시	강준	남해종합식품	106
서울특별시	배순용	대정 인터내셔널	128790
김해시	오선주	케이리할리스	8758
평택시	이두원	진가식품	122
인천광역시	홍소현	그린로더스	81479
상주시	권기호	태원 산업	7065
전주시	박동찬	시드니식품	943
등두천시	김경태	선인 상사	631
춘천시	신민주	성민식품상사	382
부천시	최해주	세미월푸드	7106
과천시	조희진	진영무역	8607
제주시	주민영	오뚜락	8838
오산시	이수호	뉴서울제과	409
수원시	조연희	시대 통상	7135
제천시	손창민	청운유통	3852
안산시	홍지우	신푸드	1149
부산광역시	오유진	케이티에스씨	7795

2. 다중 컬럼 서브쿼리

■ [예제 6-13] 각 도시마다 최고 마일리지를 보유한 고객의 정보를 보이시오

고객 테이블

담당자명	고객회사명	도시	마일리지
임재민	오케이식품	공주시	416
김지은	대림에스엠	과천시	1064
조희진	진영무역	과천시	8607
이승연	은혜식품	광명시	387
김성민	씨엔그룹	광명시	8788
강준	남해종합식품	구리시	106

서브쿼리

도시	MAX(마일리지)
공주시	416
과천시	8607
광명시	8798
구리시	106
김해시	8758

결과

도시	담당자명	고객회사명	마일리지
과천시	조희진	진영무역	8607
광명시	김성민	씨엔그룹	8788

과천시
최고
마일리지

2. 다중 컬럼 서브쿼리

확인문제

다음 SQL에서 오류가 발생하는 부분을 찾고 오류가 발생하는 이유를 적으시오.

- ① SELECT 도시
 ,담당자명
 ,고객회사명
 ,마일리지
- ② FROM 고객
- ③ WHERE (마일리지, 도시) IN (SELECT 도시, MAX(마일리지)
 ④ FROM 고객
 ⑤ GROUP BY 도시);

정답

- ③ 컬럼의 순서가 다르기 때문에 오류가 발생한다.

점검문제

점검문제

문제 1

‘배재용’ 사원의 부서명을 보이시오.

▶ 실행결과

부서명
기획부

문제 2

한 번도 주문한 적이 없는 제품의 정보를 보이시오.

▶ 실행결과

제품번호	제품명	포장단위	단가	재고	재고금액
78	웰몬트 망고 주스	0.5l bottles	3000	80	240000

점검문제

문제 3

담당자명, 고객회사명, 주문건수, 최초주문일과 최종주문일을 보이시오.

▶ 실행결과

담당자명	고객회사명	주문건수	최초주문일	최종주문일
이은광	굿모닝서울	6	2020-08-04	2022-01-11
김병현	엘케이 상사	6	2020-05-06	2021-09-27
김성민	씨엔그룹	3	2021-02-25	2021-09-16
최지수	오리안무역	15	2020-04-06	2021-12-17
강준	남해종합식품	3	2021-05-26	2021-12-01
김민경	서보암엔에프	6	2020-09-30	2022-01-07
배순율	대정 인터내셔널	4	2020-05-27	2021-11-11
박지혜	제원 인터내셔널	7	2020-12-16	2022-01-06

Thank you!

