

데이터 분석에 유용한 분석쿼리 사용하기

1. 개선된 서브쿼리 CTE 사용하기

(1) CTE 란

- CTE(Common Table Expression)
 - 공통 테이블 표현식
 - WITH으로 시작되는 SELECT 문(WITH 절이라고도 함)
 - 서브쿼리를 여러 개 선언해 사용

```
WITH cte1 AS (SELECT ... FROM ...),  
      cte2 AS (SELECT ... FROM ...),  
      ...  
SELECT ...  
  FROM cte1, cte2 ...  
WHERE ...
```

1. 개선된 서브쿼리 CTE 사용하기

(1) CTE 란

- 파생 테이블과 CTE 사용 쿼리 비교

파생 테이블

```
SELECT a.dept_no, a.dept_name,  
       mng.emp_no, mng.first_name, mng.last_name  
FROM departments a,  
     (SELECT b.dept_no, b.emp_no,  
          c.first_name, c.last_name  
      FROM dept_manager b, employees c  
      WHERE b.emp_no = c.emp_no  
            AND SYSDATE() BETWEEN b.from_date  
                                AND b.to_date  
      ) mng  
WHERE a.dept_no = mng.dept_no  
ORDER BY 1;
```

서브쿼리

CTE

```
WITH mng AS  
(SELECT b.dept_no, b.emp_no,  
         c.first_name, c.last_name  
 FROM dept_manager b, employees c  
 WHERE b.emp_no = c.emp_no  
       AND SYSDATE() BETWEEN b.from_date  
                             AND b.to_date  
)  
SELECT a.dept_no, a.dept_name,  
       b.emp_no, b.first_name, b.last_name  
FROM departments a, mng b  
WHERE a.dept_no = b.dept_no  
ORDER BY 1;
```

1. 개선된 서브쿼리 CTE 사용하기

(2) CTE로 또 다른 서브쿼리 참조하기

- CTE를 사용하면 한 서브쿼리 안에서 또 다른 서브쿼리 참조가 가능
- 파생 테이블을 또 다른 파생 테이블의 서브쿼리에서 참조 불가능

2. 윈도우 함수로 다양한 집계 값 산출하기

(1) 윈도우 함수란

- 윈도우(window) : 특정 칼럼 값을 기준으로 지정한 **로우의 그룹**
- 윈도우 함수 : 윈도우를 대상으로 연산하는 함수

함수부 **OVER** (**PARTITION BY** 칼럼1, 칼럼2 ...
ORDER BY ...)

- PARTITION BY : 윈도우 지정
- ORDER BY : 윈도우로 지정된 로우의 순서

2. 윈도우 함수로 다양한 집계 값 산출하기

(2) 윈도우 함수 사용하기

함수	설명(반환값)
ROW_NUMBER()	로우의 순번
RANK()	순위
DENSE_RANK()	누적 순위
PERCENT_RANK()	비율 순위
LAG()	현재 로우의 바로 앞 로우 값
LEAD()	현재 로우의 다음 로우 값
CUME_DIST()	누적 분포 값
NTILE()	분할 버킷 수
FIRST_VALUE()	지정된 범위에서 첫 번째 로우의 값
LAST_VALUE()	지정된 범위에서 마지막 로우의 값
NTH_VALUE()	지정된 범위에서 N번째 로우의 값

2. 윈도우 함수로 다양한 집계 값 산출하기

(3) 프레임 절로 집계 범위 조정하기

- 프레임(frame) : PARTITION BY로 지정된 파티션을 다시 나눈 하위 집합

ROWS 또는 RANGE BETWEEN frame_start AND frame_end

- ROWS : 현재 로우를 기준으로 로우 단위로 대상 프레임 지정
- RANGE : 현재 로우를 기준으로 값의 범위 단위로 대상 프레임 지정
- BETWEEN frame_start AND frame_end : frame_start와 frame_end 사이에 있는 로우 지정

2. 윈도우 함수로 다양한 집계 값 산출하기

(3) 프레임 절로 집계 범위 조정하기

- frame_start, frame_end 옵션 값
 - CURRENT ROW 현재 로우
 - UNBOUNDED PRECEDING 파티션의 첫 번째 로우
 - UNBOUNDED FOLLOWING 파티션의 마지막 로우
 - n PRECEDING
 - ROWS 사용: 현재 로우 기준 n번째 앞 로우
 - RANGE 사용: 현재 로우의 값에서 n을 뺀 값을 가진 로우
 - n FOLLOWING
 - ROWS 사용: 현재 로우 기준 n번째 뒤 로우
 - RANGE 사용: 현재 로우의 값에서 n을 더한 값을 가진 로우

2. 윈도우 함수로 다양한 집계 값 산출하기

(3) 프레임 절로 집계 범위 조정하기

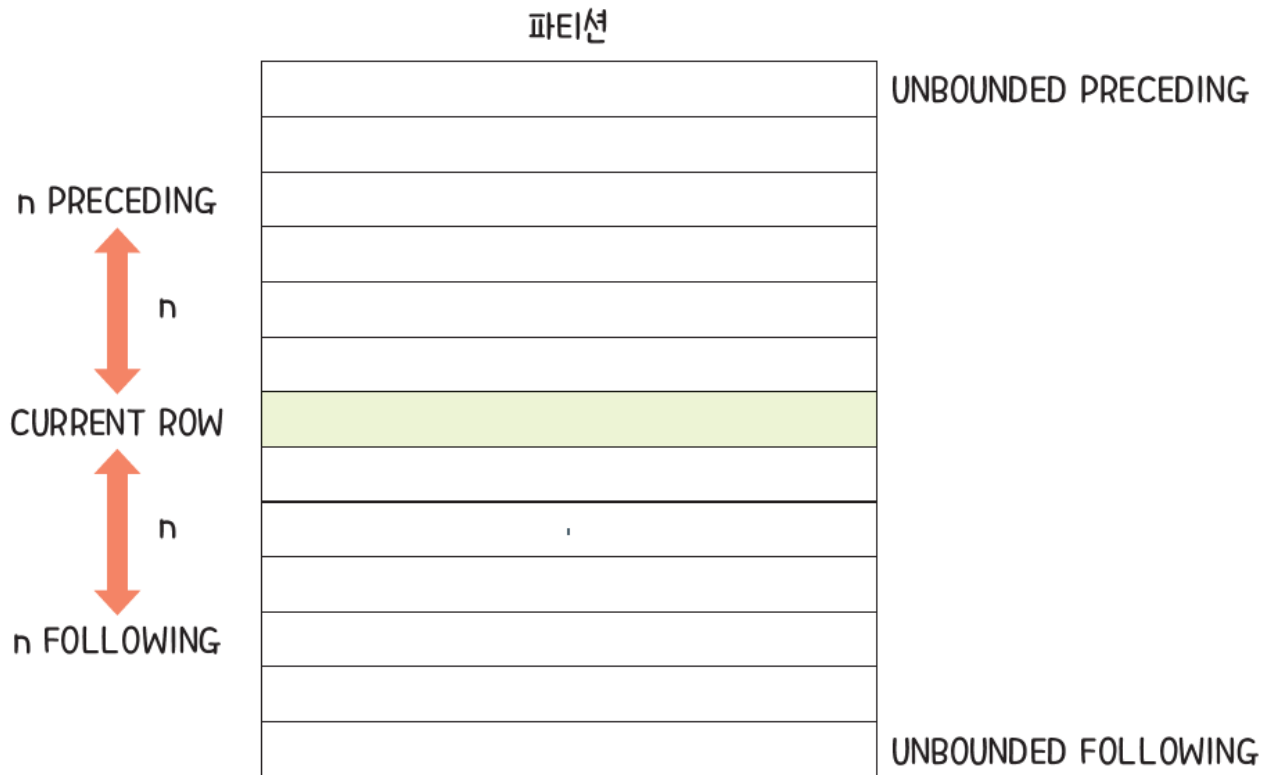
- 조합 예시

조합 예	파티션 로우 범위
UNBOUNDED PRECEDING	파티션의 첫 번째 로우부터 현재 로우까지
BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW	파티션의 첫 번째 로우부터 현재 로우까지
BETWEEN 2 PRECEDING AND CURRENT ROW	현재 로우 기준 두 번째 앞 로우부터 현재 로우까지
BETWEEN 2 PRECEDING AND 2 FOLLOWING	현재 로우 기준 두 번째 앞 로우부터 두 번째 뒤 로우까지
BETWEEN 2 PRECEDING AND UNBOUNDED FOLLOWING	현재 로우 기준 두 번째 앞 로우부터 파티션의 마지막 로우까지

2. 윈도우 함수로 다양한 집계 값 산출하기

(3) 프레임 절로 집계 범위 조정하기

- 프레임 범위



2. 윈도우 함수로 다양한 집계 값 산출하기

(4) 프레임 절과 윈도우 함수 함께 사용하기

- FIRST_VALUE(), LAST_VALUE() : 주어진 프레임의 첫 번째 로우 값과 마지막 로우 값 반환
- NTH_VALUE() : 주어진 프레임에서 매개변수 expr의 n번째 로우에 있는 값 반환

2. 윈도우 함수로 다양한 집계 값 산출하기

(5) 윈도우 별칭 사용하기

- 테이블이나 칼럼에 대한 별칭을 만들듯이 윈도우에 별칭을 지정해 OVER 절에서 사용

```
SELECT employee_id, emp_name, dept_name, salary,  
       FIRST_VALUE(salary) OVER wa firstvalue,  
       LAST_VALUE(salary) OVER wa lastvalue  
FROM emp_hierarchy  
WHERE dept_name IN ('IT', 'Finance')  
WINDOW wa AS (PARTITION BY dept_name  
              ORDER BY salary DESC  
              ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING  
              )  
ORDER BY 3, 4 DESC;
```

3. 뷰로 복잡한 쿼리 간단하게 사용하기

(1) 뷰 생성하고 사용하기

- 뷰(View)
 - 하나 이상의 테이블을 조회한 결과 집합인 독립적인 데이터베이스 객체
 - 테이블처럼 로우와 칼럼으로 구성
 - 데이터를 저장하지 않고 SELECT 문을 저장

```
CREATE [OR REPLACE] VIEW 뷰명 AS
```

```
SELECT ...;
```

3. 뷰로 복잡한 쿼리 간단하게 사용하기

(2) 뷰 수정하고 삭제하기

- 뷰 수정

```
ALTER VIEW 뷰명 AS
```

```
SELECT ...;
```

- 뷰 삭제

```
DROP VIEW 뷰명;
```