



분석함수 ✨(과제)

📄 언어	SQL
📄 유형	수업
🕒 생성 일시	@2023년 3월 31일 오후 2:17
⚙️ 이해 여부	Not started
≡ 책 페이지	
☑️ 최종 정리 여부	<input type="checkbox"/>
☑️ 필기 작성 여부	<input type="checkbox"/>
☑️ 확인 필요 여부	<input type="checkbox"/>
↗️ 공부 완료 여부	
↗️ 날짜별	
📅 복습/숙제 일자	
📎 파일과 미디어	

분석함수 기초

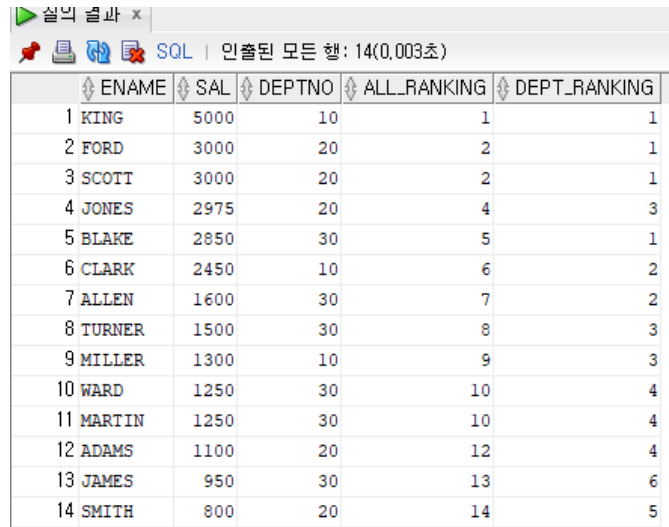
```
SELECT  
분석함수 OVER(인자값, 인자값, 인자값)  
FROM 테이블;
```

- 데이터를 통계적으로 갖고 오는 것이 필요
- OVER 함수 꼭 함께 사용

- rank

--분석함수

```
SELECT ename, sal, deptno,  
       RANK() OVER(ORDER BY sal DESC) ALL_RANKING,  
       RANK() OVER(PARTITION BY deptno ORDER BY SAL DESC) dept_ranking  
FROM emp;
```



	ENAME	SAL	DEPTNO	ALL_RANKING	DEPT_RANKING
1	KING	5000	10	1	1
2	FORD	3000	20	2	1
3	SCOTT	3000	20	2	1
4	JONES	2975	20	4	3
5	BLAKE	2850	30	5	1
6	CLARK	2450	10	6	2
7	ALLEN	1600	30	7	2
8	TURNER	1500	30	8	3
9	MILLER	1300	10	9	3
10	WARD	1250	30	10	4
11	MARTIN	1250	30	10	4
12	ADAMS	1100	20	12	4
13	JAMES	950	30	13	6
14	SMITH	800	20	14	5

- PARTITION BY : 그룹핑 역할



구분하기

- OVER함수 안에 인자값으로 order by 넣었을 때
: OVER가 제일 먼저 돌아감
- over 함수 내에 partition by를 넣었을 때에 GROUP BY와의 차이

오라클에서 분석함수를 사용할 때 PARTITION BY를 사용하여 그룹으로 묶어서 연산을 할 수 있다. GROUP BY 절을 사용하지 않고, 조회된 각 행에 그룹으로 집계된 값을 표시할 때 OVER 절과 함께 PARTITION BY 절을 사용하면 된다.

```
SELECT job
      , SUM(sal)
FROM emp
WHERE job IN ('MANAGER', 'SALESMAN')
GROUP BY job
ORDER BY job
```

JOB	SUM(SAL)
MANAGER	8275
SALESMAN	5600

```
SELECT empno
      , ename
      , job
      , SUM(sal) OVER(PARTITION BY job)
FROM emp
WHERE job IN ('MANAGER', 'SALESMAN')
ORDER BY job
```

EMPNO	ENAME	JOB	SUM(SAL)OVER(PARTITIONBYJOB)
7566	JONES	MANAGER	8275
7782	CLARK	MANAGER	8275
7698	BLAKE	MANAGER	8275
7654	MARTIN	SALESMAN	5600
7521	WARD	SALESMAN	5600
7844	TURNER	SALESMAN	5600
7499	ALLEN	SALESMAN	5600

오라클 PARTITION BY 절

GROUP BY : 그룹당 값이 한개씩 나옴

PARTITION BY : 모든 ROW별로 값이 하나씩 추출됨

:: 모든 사원마다 GROUPING 처리하기 힘들다, PARTITION으로 처리 가능



윈도우는 분석함수가 적용되는 하나의 데이터 덩어리

집계 값을 구할 때 보통은 그룹 쿼리를 사용하는데, 이때 GROUP BY 절에 의해 최종 쿼리 결과는 그룹별로 row 수가 줄어든다.

이에 반해, 집계 함수를 사용하면 row의 손실 없이도 그룹별 집계 값을 산출해 낼 수 있다.

분석 함수에서 사용하는 row별 그룹을 **윈도우**라 부름.

```
분석 함수 (매개 변수) OVER
(PARTITION BY expr1, expr2, ...
ORDER BY expr3, expr4 ...
window 절)
```

분석을 진행할 전체를 윈도우로 지칭



실행 순서

OVER 절 실행 순서

FROM – WHERE – GROUP BY – HAVING – SELECT – ORDER BY – OVER

--테이블 1번 사용(정렬 문제)

```
SELECT ENAME, SAL, ROWNUM FROM EMP ORDER BY SAL DESC;
```

--테이블 2번 사용

```
SELECT ENAME, SAL, ROWNUM
FROM(SELECT * FROM EMP ORDER BY SAL DESC) ;
```

--분석함수 사용

```
SELECT ENAME, SAL,  
       ROW_NUMBER() OVER(ORDER BY SAL DESC) RANKING  
FROM EMP;
```

분석함수 예제

```
SELECT empno,ename, deptno, sal,  
       RANK() OVER(PARTITION BY deptno ORDER BY sal DESC) rank  
FROM emp;
```

```
SELECT empno,ename, deptno, sal,  
       DENSE_RANK() OVER(PARTITION BY deptno ORDER BY sal DESC) rank  
FROM emp;
```

```
SELECT empno,ename, sal, hiredate,  
       ROW_NUMBER() OVER(ORDER BY sal DESC, hiredate asc) as "순번"  
FROM EMP;
```

```
SELECT ENAME, SAL, NTILE(4) OVER (ORDER BY SAL)  
FROM EMP;
```