SQL Pogramming

- Day 19 -

2023. 04

목차

Day 19. 분석함수1

■ 분석함수(Analytic Function) 정의

- ▶ 윈도우함수(Window Function)이라고도 함
- ▶ 기존의 함수들은 행(레코드) 단위로 자기 자신의 컬럼 값들만 참조할 수 있었음
- ▶ 행(레코드)과 행(레코드) 간의 관계를 정의하거나, 행과 행 간을 비교 및 연산하는 것을 하나의 SQL 문으로 처리하는 것은 매우 어려운 문제였음
- ▶ 부분적이나마 행(레코드)과 행(레코드) 간의 관계를 쉽게 정의하기 위해 만든 함수가 바로 분석 함수임
- ▶ 분석 함수를 이용하면 복잡한 프로그램을 하나의 SQL 문장으로 쉽게 해결할 수 있음
- ▶ 분석 함수는 기존에 사용하던 집계 함수도 있고, 새로이 분석 함수 전용으로 만들어진 기능도 있음
- ▶ 분석 함수는 다른 함수와는 달리 중첩해서 사용하지 못함
- ▶ 분석 함수는 서브쿼리에서는 사용할 수 있음

■ 분석함수(Analytic Function) 종류

- ▶ 순위 함수
  - ○ RANK, DENSE_RANK, ROW_NUMBER
- ▶ 집계 함수
  - ○ SUM, MAX, MIN, AVG, COUNT
- ▶ 행 순서 함수 (오라클)
  - ○ FIRST_VALUE, LAST_VALUE, LAG, LEAD
- ▶ 비율 함수
  - ○ CUME_DIST, PERCENT_RANK
  - ○ NTILE
  - ○ RATIO_TO_REPORT (오라클)
- ▶ 선형 분석을 포함한 통계 함수

Day 19. 분석함수1

■ 분석함수(Analytic Function) Syntax

SELECT **분석함수** (인수들) **OVER**
([**PARTITION BY** 절]
[**ORDER BY** 절] [WINDOWING 절])
FROM 테이블명

▶ 분석함수 → 이전 페이지에서 기술한 함수들
▶ 인수들 → 함수에 따라 0 ~N개의 인수가 지정될 수 있음
▶ OVER → 분석 함수에서는 OVER가 반드시 포함됨
▶ PARTITION BY 절 → 전체 집합을 기준에 의해서 소그룹으로 나눌 수 있음
▶ ORDER BY 절 → 어떤 항목들에 대해 정렬할 지 지정
▶ WINDOWING 절 → 행(레코드) 기준의 범위를 지정
○ ROWS : 물리적인 결과 행의 수
○ RANGE : 논리적인 값에 의한 범위
○ 둘 중 선택 가능

■ RANK

▶ ORDER BY를 포함한 QUERY 문에서 특정 항목(컬럼)에 대한 순위를 구하는 함수

▶ 특정 범위 내에서 순위를 구할 수도 있고, 전체 범위의 데이터에 대한 순위를 구할 수도 있음

▶ 동일한 값에 대해서는 동일한 순위를 부여함

```
SELECT JOB, ENAME, SAL,
 RANK() OVER (ORDER BY SAL DESC) ALL_RANK,
 RANK() OVER (PARTITION BY JOB ORDER BY SAL DESC) JOB_RANK
FROM EMP;
```

JOB	ENAME	SAL	ALL_RANK	JOB_RANK
PRESIDENT	KING	5000	1	1
ANALYST	FORD	3000	2	1
ANALYST	SCOTT	3000	2	1
MANAGER	JONES	2975	4	1
MANAGER	BLAKE	2850	5	2
MANAGER	CLARK	2450	6	3
SALESMAN	ALLEN	1600	7	1
SALESMAN	TURNER	1500	8	2
CLERK	MILLER	1300	9	1
SALESMAN	WARD	1250	10	3
SALESMAN	MARTIN	1250	10	3
CLERK	ADAMS	1100	12	2
CLERK	JAMES	950	13	3
CLERK	SMITH	800	14	4

<실습 대상>

```
SELECT INVOICE_NO
      ,LINE_NO
      ,ITEM_NM
      ,ORDER_QTY
  FROM LO_OUT_D
 WHERE INVOICE_NO IN ('346724705050', '346724705061')
 ORDER BY INVOICE_NO, LINE_NO
```

■ DENSE_RANK

▶ RANK 함수와 모든 면에서 유사

▶ 동일한 순위를 하나의 건 수로 취급하는 것이 다름

```
SELECT JOB, ENAME, SAL,
 RANK() OVER (ORDER BY SAL DESC) RANK,
 DENSE_RANK() OVER (ORDER BY SAL DESC) DENSE_RANK
FROM EMP;
```

JOB	ENAME	SAL	RANK	DENSE_RANK
PRESIDENT	KING	5000	1	1
ANALYST	FORD	3000	2	2
ANALYST	SCOTT	3000	2	2
MANAGER	JONES	2975	4	3
MANAGER	BLAKE	2850	5	4
MANAGER	CLARK	2450	6	5
SALESMAN	ALLEN	1600	7	6
SALESMAN	TURNER	1500	8	7
CLERK	MILLER	1300	9	8
SALESMAN	WARD	1250	10	9
SALESMAN	MARTIN	1250	10	9
CLERK	ADAMS	1100	12	10
CLERK	JAMES	950	13	11
CLERK	SMITH	800	14	12

■ ROW_NUMBER

▶ RANK나 DENSE_RANK 함수가 동일한 값에 대해서는 동일한 순위를 부여하는데 반해, 동일한 값이라도 유니크한 순위를 부여함

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

JOB	ENAME	SAL	RANK	ROW_NUMBER
PRESIDENT	KING	5000	1	1
ANALYST	FORD	3000	2	2
ANALYST	SCOTT	3000	2	3
MANAGER	JONES	2975	4	4
MANAGER	BLAKE	2850	5	5
MANAGER	CLARK	2450	6	6
SALESMAN	ALLEN	1600	7	7
SALESMAN	TURNER	1500	8	8
CLERK	MILLER	1300	9	9
SALESMAN	WARD	1250	10	10
SALESMAN	MARTIN	1250	10	11
CLERK	ADAMS	1100	12	12
CLERK	JAMES	950	13	13
CLERK	SMITH	800	14	14

■ MAX

▶ 파티션별 최대값을 구함

```
SELECT MGR, ENAME, SAL,
       MAX(SAL) OVER (PARTITION BY MGR) AS MGR_MAX
FROM   EMP;
```

MGR	ENAME	SAL	MGR_MAX
7566	FORD	3000	3000
7566	SCOTT	3000	3000
7698	JAMES	950	1600
7698	ALLEN	1600	1600
7698	WARD	1250	1600
7698	TURNER	1500	1600
7698	MARTIN	1250	1600
7782	MILLER	1300	1300
7788	ADAMS	1100	1100
7839	BLAKE	2850	2975
7839	JONES	2975	2975
7839	CLARK	2450	2975
7902	SMITH	800	800
	KING	5000	5000

■ MIN

▶ 파티션별 최소값을 구함

▶ RANGE UNBOUNDED PRECEDING

○ 현재 행을 기준으로 파티션 내의 첫 번째 행까지의 범위를 지정함

```
SELECT MGR, ENAME, HIREDATE, SAL,
       MIN(SAL) OVER(PARTITION BY MGR ORDER BY HIREDATE
                RANGE UNBOUNDED PRECEDING) AS MGR_MIN
FROM   EMP;
```

MGR	ENAME	HIREDATE	SAL	MGR_MIN
7566	FORD	1981-12-03	3000	3000
7566	SCOTT	1987-07-13	3000	3000
7698	ALLEN	1981-02-20	1600	1600
7698	WARD	1981-02-22	1250	1250
7698	TURNER	1981-09-08	1500	1250
7698	MARTIN	1981-09-28	1250	1250
7698	JAMES	1981-12-03	950	950
7782	MILLER	1982-01-23	1300	1300
7788	ADAMS	1987-07-13	1100	1100
7839	JONES	1981-04-02	2975	2975
7839	BLAKE	1981-05-01	2850	2850
7839	CLARK	1981-06-09	2450	2450
7902	SMITH	1980-12-17	800	800
	KING	1981-11-17	5000	5000

■ SUM

▶ 파티션별 합계값을 구함

▶ RANGE UNBOUNDED PRECEDING

○ 현재 행을 기준으로 파티션 내의 첫 번째 행까지의 범위를 지정함

```
SELECT MGR, ENAME, SAL,
       SUM(SAL) OVER
      (PARTITION BY MGR ORDER BY SAL RANGE UNBOUNDED PRECEDING)
       AS MGR_SUM
FROM   EMP
```

MGR	ENAME	SAL	MGR_SUM	참조
7566	SCOTT	3000	6000	
7566	FORD	3000	6000	
7698	JAMES	950	950	
7698	WARD	1250	3450	**
7698	MARTIN	1250	3450	**
7698	TURNER	1500	4950	
7698	ALLEN	1600	6550	
7782	MILLER	1300	1300	
7788	ADAMS	1100	1100	
7839	CLARK	2450	2450	
7839	BLAKE	2850	5300	
7839	JONES	2975	8275	
7902	SMITH	800	800	
	KING	5000	5000	

■ COUNT

▶ 파티션별 건 수를 구함

▶ RANGE BETWEEN 50 PRECEDING AND 150 FOLLOWING

○ 현재 행의 값을 기준으로 파티션 내에서 -50에서 150 사이 값을 가진 모든 행이 대상이 됨

○ RANGE는 현재 행의 데이터 값을 기준으로 앞뒤 데이터 값의 범위를 말하는 것임

```
SELECT ENAME, SAL,
       COUNT(*) OVER (ORDER BY SAL
       RANGE BETWEEN 50 PRECEDING AND 150 FOLLOWING) AS MOV_COUNT
FROM   EMP;
```

ENAME	SAL	MOV_CNT	참조,범위값
SMITH	800	2	(750~ 950)
JAMES	950	2	(900~1100)
ADAMS	1100	3	** (1050~1250)
WARD	1250	3	(1200~1400)
MARTIN	1250	3	(1200~1400)
MILLER	1300	3	(1250~1450)
TURNER	1500	2	(1450~1650)
ALLEN	1600	1	(1550~1750)
CLARK	2450	1	(2400~2600)
BLAKE	2850	4	(2800~3000)
JONES	2975	3	(2925~3125)
SCOTT	3000	3	(2950~3100)
FORD	3000	3	(2950~3100)
KING	5000	1	(4950~5100)

■ LAG / LEAD

▶ 이전(이후) 몇 번째 행의 값을 가져올 수 있음

▶ 3개의 인자들을 가질 수 있음

○ 첫 번째 인자 → 표시할 항목(컬럼)

○ 두 번째 인자 → 몇 번째 행을 가져올 지 결정 (디폴트 1)

○ 세 번째 인자 → NULL일 경우의 대체값

```
SELECT ENAME, HIREDATE, SAL,
       LAG(SAL) OVER
      (ORDER BY HIREDATE)
       AS PREV_SAL
FROM   EMP
WHERE  JOB = 'SALESMAN' ;
```

ENAME	HIREDATE	SAL	PREV_SAL
ALLEN	1981-02-20	1600	
WARD	1981-02-22	1250	1600
TURNER	1981-09-08	1500	1250
MARTIN	1981-09-28	1250	1500

```
SELECT ENAME, HIREDATE, SAL,
       LAG(SAL, 2, 0) OVER
      (ORDER BY HIREDATE)
       AS PREV_SAL
FROM   EMP
WHERE  JOB = 'SALESMAN' ;
```

ENAME	HIREDATE	SAL	PREV_SAL
ALLEN	1981-02-20	1600	0
WARD	1981-02-22	1250	0
TURNER	1981-09-08	1500	1600
MARTIN	1981-09-28	1250	1250

Day 19. 분석함수1

실전문제① ▶ 분석함수 MIN/MAX 사용법 익히기				
《테이블》	■ LO_OUT_D(출고주문상세)			
《조건》	■ INVOICE_NO(송장번호)	▶ 346724703845 또는 346724706214 또는 346724706225		
《정렬》	■			
《특징》	■ INVOICE_NO(송장번호) 단위의 ORDER_QTY(출고수량) 최소값/최대값 ■ 전체에서 ORDER_QTY(출고수량) 최소값/최대값			

결과 ▼ 총 건수 : 7건

INVOICE_NO	LINE_NO	ITEM_NM	ORDER_QTY	MIN_1	MIN_2	MAX_1	MAX_2
346724703845	1	Dole 곤약젤리 얼그레이자몽 130ml*10입	1	1	1	1	36
346724703845	2	Dole 곤약젤리 자스민복숭아 130ml*10입	1	1	1	1	36
346724706214	1	뉴트리플랜 건강프로젝트-장 160g	36	8	1	36	36
346724706214	2	뉴트리플랜 건강프로젝트-항산화 160g	8	8	1	36	36
346724706214	4	뉴트리플랜 건강프로젝트-피부모질 160g	36	8	1	36	36
346724706225	2	뉴트리플랜 건강프로젝트-항산화 160g	28	28	1	36	36
346724706225	3	뉴트리플랜 건강프로젝트-체중조절 160g	36	28	1	36	36

Thank you !

ASETEC Location http://www.asetec.co.kr

본사. 경기도 성남시 분당구 성남대로 331번길 8, 킨스타워 2201호 TEL.031.609.7000 FAX.031.609.7009
부산. 부산광역시 해운대구 센텀동로 99 TEL.051.506.6352 FAX.051.504.8794