

[Database]Oracle SQL ORDER BY 절, GROUP BY 절 함수 사용법

노트북: [TIL-MY]

만든 날짜: 2020-07-16 오전 9:01

URL: <https://continuous-development.tistory.com/22>

나무늘보의 개발 블로그



Database

[Database]Oracle SQL ORDER BY 절, GROUP BY 절 함수 사용법

| 꾸까꾸 | 2020. 7. 16. 00:01 | 수정 | 삭제

ORDER BY

검색된 데이터를 순차적으로 정렬해서 보여주는 함수

```
SELECT  
FROM  
WHERE  
ORDER BY 기준1 [ASC/DESC]
```

DESC - 내림차수(내려가면서 작아진다)

ASC - 오름 차수(내려가면서 커진다.)

```
-- ORDER BY절
```

```
SELECT EMP_NAME,
        SALARY
FROM EMPLOYEE
WHERE DEPT_ID='50'
OR DEPT_ID IS NULL
ORDER BY SALARY DESC;
```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 8(0.002초)

	EMP_NAME	SALARY
1	김순이	3400000
2	박하일	2600000
3	나승원	2300000
4	심하균	2300000
5	김예수	2100000
6	성해교	1900000
7	염정하	1500000
8	정지현	1500000

연봉순으로 내림차수로 결과를 보여준다.

```
SELECT EMP_NAME,
        HIRE_DATE,
        DEPT_ID
FROM EMPLOYEE
WHERE HIRE_DATE > TO_DATE('20030101', 'YYYYMMDD')
ORDER BY DEPT_ID DESC NULLS LAST,
        HIRE_DATE,
        EMP_NAME;
```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 11(0.003초)

	EMP_NAME	HIRE_DATE	DEPT_ID
1	강중훈	04/04/30	90
2	한선기	90/04/01	90
3	염정하	04/07/21	80
4	성해교	03/08/16	50
5	정지현	04/07/15	50
6	박하일	04/11/10	50
7	이중기	04/10/01	20
8	감우섭	05/07/31	20
9	고승우	03/04/11	10
10	염정하	03/09/17	(null)
11	심하균	04/09/30	(null)

이건 DEPT_ID 순으로 내림차수를 한데 그냥 출력하면 NULL이 제일 위에 뜬다.

이걸 내리기 위에 DESC에 NULLS LAST을 넣어준다.

이건 NULL을 제일 마지막에 출력한다는 옵션이다.

반대로 NULLS FIRST 도 있다.

그리고 order by 는 인덱스나 AS 로 정의한 별칭으로도 가능하다.

```
-- 별칭으로 ORDER BY
SELECT EMP_NAME AS 이름,
       HIRE_DATE AS 입사일,
       DEPT_ID AS 부서코드
FROM EMPLOYEE
WHERE HIRE_DATE > TO_DATE('20030101','YYYYMMDD')
ORDER BY 부서코드 DESC,
         입사일,
         이름;

-- 인덱스로 ORDER BY
SELECT EMP_NAME AS 이름,
       HIRE_DATE AS 입사일,
       DEPT_ID AS 부서코드
FROM EMPLOYEE
WHERE HIRE_DATE > TO_DATE('20030101','YYYYMMDD')
ORDER BY 3 DESC,
         2,
         1;
```

스크립트 출력 x | 질의 결과 x

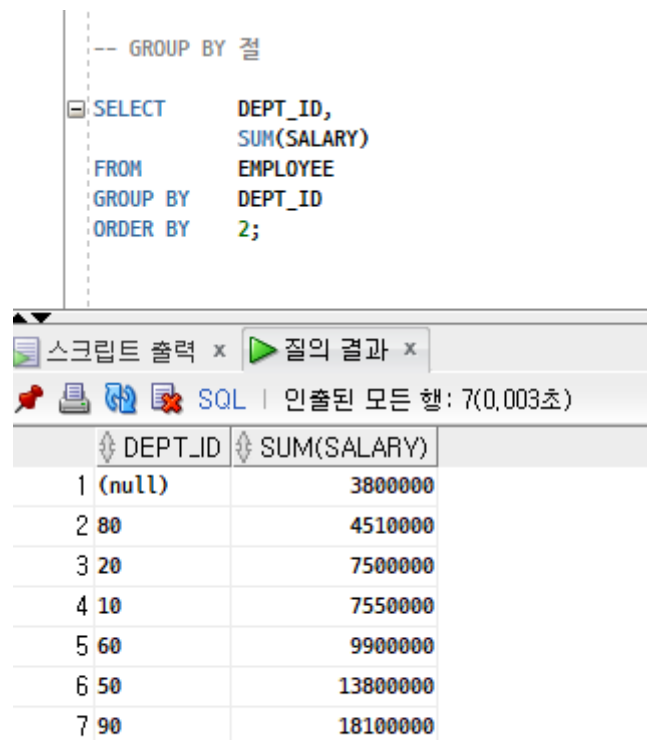
SQL | 인출된 모든 행: 11(0.003초)

	이름	입사일	부서코드
1	염정하	03/09/17	(null)
2	심하균	04/09/30	(null)
3	강중훈	04/04/30	90
4	한선기	90/04/01	90
5	염정하	04/07/21	80
6	성해교	03/08/16	50
7	정지현	04/07/15	50
8	박하일	04/11/10	50
9	이중기	04/10/01	20
10	감우섭	05/07/31	20
11	고승우	03/04/11	10

GROUP BY

데이터 들을 그룹으로 묶어주는 함수

```
SELECT  
FROM  
WHERE  
GROUP BY column_name | expr
```



The screenshot shows a SQL IDE window with a query editor and a results pane. The query in the editor is:

```
-- GROUP BY 절  
SELECT      DEPT_ID,  
            SUM(SALARY)  
FROM        EMPLOYEE  
GROUP BY    DEPT_ID  
ORDER BY    2;
```

The results pane shows the output of the query, with columns DEPT_ID and SUM(SALARY). The data is as follows:

DEPT_ID	SUM(SALARY)
1 (null)	3800000
2 80	4510000
3 20	7500000
4 10	7550000
5 60	9900000
6 50	13800000
7 90	18100000

첫번째 쿼리는 DEPT_ID 라는 부서 번호로 그룹을 묶었다. GROUP BY 를 쓸때 주의 할 점은

그룹으로 묶여있는 것을 출력 할때는 다른 것들도 그룹으로 묶여서 출력 할수 있게끔 해야한다.

왜냐하면 이미 묶어서 출력을 하기 때문에 다른 것들이 묶여서 보여지지 않고 개별로 나온다면 에러가나기 때문이다.

SUM - 총합 계산

AVG - 평균 계산

MIN - 최소값 반환

MAX - 최대값 반환

COUNT - Result Set 전체 행 수 반환

이것들은 컬럼에 대한 값을 묶어서 출력해주는 간단한 함수이다. 이런 식으로 GROUP BY 절로 묶은 컬럼과

```
-- GROUP BY 절

SELECT DEPT_ID,
       SUM(SALARY)
FROM   EMPLOYEE
GROUP BY DEPT_ID
ORDER BY 2;
```

```
SELECT JOB_ID,
       TRUNC(AVG(SALARY))
FROM   EMPLOYEE
GROUP BY JOB_ID
ORDER BY 2;
```

스크립트 출력 x 질의 결과 x		
SQL 인출된 모든 행: 8(0,005초)		
	JOB_ID	TRUNC(AVG(SALARY))
1	J7	1600000
2	(null)	2300000
3	J5	2333333
4	J4	2525000
5	J6	2640000
6	J3	3566666
7	J2	4550000
8	J1	9000000

GROUP BY 는 ORDER BY 처럼 인덱스나 별칭으로 사용 할 수 없다.

```

--성별에 따른 급여 평균
SELECT      DECODE(SUBSTR(EMP_NO,8,1),
                  '1','남자',
                  '3','남자','여자') AS "성별 임금 평균",
            TRUNC(AVG(SALARY))
FROM        EMPLOYEE
GROUP BY    DECODE(SUBSTR(EMP_NO,8,1),
                  '1','남자',
                  '3','남자','여자')
ORDER BY    2;

```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 2(0.003초)

성별 임금 평균	TRUNC(AVG(SALARY))
1 여자	2257500
2 남자	3364285

ROLLUP 중간중간에 대한 소계를 넣어준다.

```

-- 중간 중간에 대한 소계를 넣어주는 함수라고 보면 된다.
SELECT      DEPT_ID,
            EMP_NAME,
            COUNT(*)
FROM        EMPLOYEE
GROUP BY    ROLLUP(DEPT_ID,
                  EMP_NAME);

```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 30(0.003초)

DEPT_ID	EMP_NAME	COUNT(*)
1 (null)	심하균	1
2 (null)	염정하	1
3 (null)	(null)	2
4 10	고승우	1
5 10	권상후	1
6 10	임영애	1
7 10	(null)	3
8 20	감우섭	1
9 20	김술오	1
10 20	이중기	1
11 20	(null)	3
12 50	김순이	1
13 50	김예스	1

HAVING 절

그룹에 대한 제한을 하기 위해 사용한다.

GROUP BY 에 대한 조건을 걸기 위해 HAVING을 사용한다.

HAVING은 그룹에 대한 조건이고

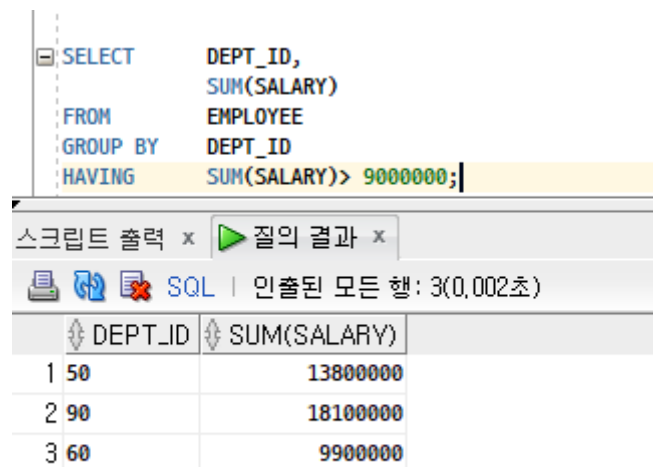
WHERE는 테이블에 대한 조건이다.

SELECT

FROM

GROUP BY

HAVING



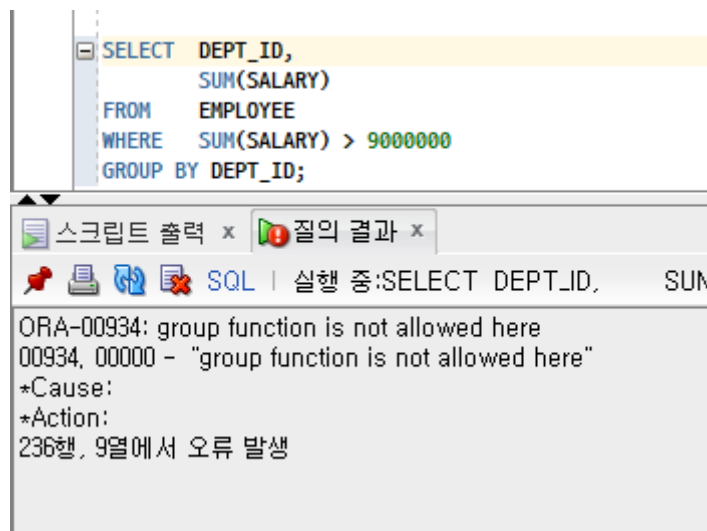
```
SELECT DEPT_ID, SUM(SALARY)
FROM EMPLOYEE
GROUP BY DEPT_ID
HAVING SUM(SALARY) > 9000000;
```

스크립트 출력 x 실행 결과 x

SQL | 인출된 모든 행: 3(0,002초)

	DEPT_ID	SUM(SALARY)
1	50	13800000
2	90	18100000
3	60	9900000

SELECT 부분에 보면 SELECT 의 SUM 에대한 조건은 HAVING 절에 들어간다.



WHERE에 그룹의 조건을 넣어주면 에러가 난다.

'Database' 카테고리의 다른 글

- [Database] ORACLE & ANSI JOIN 절 (0) 08:57:15
- [Database]Oracle SQL ORDER BY 절,GROUP BY 절 함수 사용법 (0) 00:01:13
- [Database] Oracle SQL DECODE & CASE 함수 사용법 (0) 2020.07.15
- [Database]Oracle SQL NVL,NVL2 함수 사용방법 (0) 2020.07.15
- [Database] 데이터 타입 변환 TO_DATE,TO_NUMBER,TO_CHAR 사용법 (0) 2020.07.15
- [Database] 날짜함수 SYSDATE,ADD_MONTHS,MONTHS_BETWEEN 함수 사용법 (0) 2020.07.14

태그 #GROUP BY, #GROUP BY 사용, #GROUP BY 함수, #Oracle, #order by, #ORDER BY 사용, #ORDER BY 함수, #SQL GROUP BY, #SQL ORDER BY

'Database' Related Articles

FROM EMPLOYEE E, DEPARTMENT D WHERE E.DEPT_ID = D.DEPT_ID;

스크립트 출력

실행의 결과

SQL | 인출된 모든 행: 20(0.002초)

EMP_NAME	DEPT_NAME
1 환선기	해외영업3팀
2 강종훈	해외영업3팀
3 최만석	해외영업3팀
4 정도연	기술지원팀
5 안석규	기술지원팀

SQL | 인출된 모든 행: 60(0.004초)

EMP_NAME	EMP_NO	SUBSTR(EMP_NO,8,1)	GENDER	GENDER2
1환	641231-2269880	2	여자	여자
1수	651122-2592938	2	여자	여자
1명	871204-1540881	1	남자	남자
1이	741122-2515789	2	여자	여자
1교	640524-2148339	2	여자	여자
1일	891225-1869181	1	남자	남자

SQL | 인출된 모든 행: 40(0.001초)

EMP_NAME	SALARY	NVL(BONUS,PCT*3)
1 환선기	3500000	0.2
2 강종훈	3500000	0
3 최만석	3500000	0
4 조재형	3800000	0

SQL | 인출된 모든 행: 40(0.001초)

EMP_NAME	SALARY	NVL(BONUS,PCT*3)
1 환선기	3500000	0.2
2 강종훈	3500000	0
3 최만석	3500000	0
4 조재형	3800000	0

SELECT EMP_NAME, SALARY, NVL(BONUS,PCT*3) FROM EMPLOYEE WHERE SALARY > 3500000;

스크립트 출력

실행의 결과

SQL | 인출된 모든 행: 40(0.001초)

EMP_NAME	SALARY	NVL(BONUS,PCT*3)
1 환선기	3500000	0.2
2 강종훈	3500000	0
3 최만석	3500000	0
4 조재형	3800000	0

SQL | 인출된 모든 행: 40(0.001초)

EMP_NAME	SALARY	NVL(BONUS,PCT*3)
1 환선기	3500000	0.2
2 강종훈	3500000	0
3 최만석	3500000	0
4 조재형	3800000	0

