

sql-study-day9

07 - 1 하나의 열에 출력 결과를 담은 다중행 함수

- 다중행 함수는 여러 행을 바탕으로 하나의 결과 값을 도출해내기 위해 사용하는 함수이다.

```
SELECT SAL  
FROM EMP;
```

```
SELECT SUM(SAL)  
FROM EMP;
```

	SAL
1	800
2	1600
3	1250
4	2975
5	1250
6	2850
7	2450
8	3000
9	5000
10	1500
11	1100
12	950
13	3000
14	1300

	SUM(SAL)
1	29025

SUM 함수는 SELECT문으로 조회된 행에 지정한 열 값을 모두 더한 값을 반환해 주는 함수이다.

함수	설명
SUM	지정한 데이터의 합 반환
COUNT	지정한 데이터의 개수 반환
MAX	지정한 데이터 중 최댓값 반환

함수	설명
MIN	지정한 데이터 중 최솟값 반환
AVG	지정한 데이터의 평균값 반환

합계를 구하는 SUM함수

SUM([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)]
[합계를 구할 열이나 연산자, 함수를 사용한 데이터(필수)]) // 1.

번호	설명
1.	합계를 구할 데이터를 지정합니다.

SUM함수를 분석하는 용도로 사용한다면 다음과 같이 함수를 작성한 후 OVER절을 사용할 수도 있다.

SUM([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)]
[합계를 구할 열이나 연산자, 함수를 사용한 데이터(필수)])
OVER(분석을 위한 여러 문법을 지정)(선택)

데이터 개수를 구해 주는 COUNT 함수

- COUNT 함수는 데이터 개수를 출력하는 데 사용한다.

COUNT([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)]
[개수를 구할 열이나 연산자, 함수를 사용한 데이터(필수)])
OVER(분석을 위한 여러 문법 지정)(선택) // 1.

번호	설명
1.	결과 행의 개수를 출력합니다.

```
SELECT *
FROM EMP;
```

```
SELECT COUNT(*)
FROM EMP;
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	80/12/17	800	(null)	20
2	7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
3	7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
4	7566	JONES	MANAGER	7839	81/04/02	2975	(null)	20
5	7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
6	7698	BLAKE	MANAGER	7839	81/05/01	2850	(null)	30
7	7782	CLARK	MANAGER	7839	81/06/09	2450	(null)	10
8	7788	SCOTT	ANALYST	7566	87/04/19	3000	(null)	20
9	7839	KING	PRESIDENT	(null)	81/11/17	5000	(null)	10
10	7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
11	7876	ADAMS	CLERK	7788	87/05/23	1100	(null)	20
12	7900	JAMES	CLERK	7698	81/12/03	950	(null)	30
13	7902	FORD	ANALYST	7566	81/12/03	3000	(null)	20
14	7934	MILLER	CLERK	7782	82/01/23	1300	(null)	10

	COUNT(*)
1	14

최댓값과 최솟값을 구하는 MAX, MIN 함수

MAX 함수와 MIN 함수는 단어 의미 그대로 입력 데이터 중 최댓값과 최솟값을 반환하는 함수이다.

MAX([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)]
[최댓값을 구할 열이나 연산자, 함수를 사용한 데이터(필수)])
OVER(분석을 위한 여러 문법 지정)(선택) // 1.

번호	설명
1.	결과 행에서 최댓값을 출력합니다.

MIN([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)]
[최솟값을 구할 열이나 연산자, 함수를 사용한 데이터(필수)])
OVER(분석을 위한 여러 문법을 지정)(선택) // 1.

번호	설명
1.	결과 행에서 최솟값을 출력합니다.

평균 값을 구하는 AVG 함수

AVG 함수는 입력 데이터의 평균 값을 구하는 함수이다.

AVG([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)])
 [평균 값을 구할 열이나 연산자, 함수를 사용한 데이터(필수)]
 OVER(분석을 위한 여러 문법을 지정)(선택) // 1.

번호	설명
1.	결과 행의 평균 값을 반환합니다.

07 - 2 결과 값을 원하는 열로 묶어 출력하는 GROUP BY 절

GROUP BY절의 기본 사용법

SELECT문에서는 GROUP BY절을 작성하여 데이터를 그룹화할 수 있다.

```
SELECT  [조회할 열1 이름], [열2 이름], ..., [열N 이름]
FROM    [조회할 테이블 이름]
WHERE   [조회할 행을 선별하는 조건식]
GROUP BY[그룹화할 열을 지정(여러 개 지정 가능)] // 1.
ORDER BY[정렬하려는 열 지정]
```

번호	키워드	필수 요소	선택 요소	설명
1.	GROUP BY	그룹화할 열 또는 데이터 지정	-	특정 열 또는 데이터를 기준으로 데이터를 그룹으로 묶습니다.

GROUP BY절을 사용할 때 유의점

GROUP BY절을 사용하여 출력 데이터를 그룹화할 경우 유의해야 할 점이 있는데 바로 다중행 함수를 사용하지 않은 일반 열은 GROUP BY절에 명시하지 않으면 SELECT절에서 사용할 수 없다는 것이다.

```
SELECT ENAME, DEPTNO, AVG(SAL)
FROM EMP
GROUP BY DEPTNO;
```

```
ORA-00979: GROUP BY 표현식이 아닙니다.
00979, 00000 - "not a GROUP BY expression"
*Cause:
*Action:
1행, 8열에서 오류 발생
```

07 - 3 GROUP BY절에 조건을 줄 때 사용하는 HAVING절

- HAVING절은 SELECT문에 GROUP BY절이 존재할 때만 사용할 수 있다. 그리고 GROUP BY절을 통해 그룹화된 결과 값의 범위를 제한하는 데 사용한다.

HAVING절의 기본 사용법

```
SELECT [조회할 열1 이름], [열2 이름], ..., [열N 이름]
FROM [조회할 테이블 이름]
WHERE [조회할 행을 선별하는 조건식]
GROUP BY [그룹화할 열 지정(여러 개 지정 가능)]
HAVING[출력 그룹을 제한하는 조건식] // 1.
ORDER BY [정렬하려는 열 지정];
```

번호	키워드	필수 요소	선택 요소	설명
1.	HAVING	조건식	-	GROUP BY절을 사용해 그룹화된 결과 중 출력 그룹을 선별하는 조건식을 지정합니다.

HAVING절을 사용할 때 유의점

조건식을 지정한다는 점에서 HAVING절이 WHERE절과 비슷하다고 생각할 수도 있다. 하지만 WHERE절은 출력 대상 행을 제한하고, HAVING절은 그룹화된 대상을 출력에서 제한하므로 쓰임새는 전혀 다르다.

WHERE절과 HAVING절의 차이점

- WHERE절을 사용하지 않고 HAVING 절만 사용한 경우

```
SELECT DEPTNO, JOB, AVG(SAL)
FROM EMP
GROUP BY DEPTNO, JOB
HAVING AVG(SAL) >= 2000
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	AVG(SAL)
1	10	MANAGER	2450
2	10	PRESIDENT	5000
3	20	ANALYST	3000
4	20	MANAGER	2975
5	30	MANAGER	2850

- WHERE절과 HAVING절을 모두 사용한 경우

```
SELECT DEPTNO, JOB, AVG(SAL)
FROM EMP
WHERE SAL <= 3000
GROUP BY DEPTNO, JOB
HAVING AVG(SAL) >= 2000
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	AVG(SAL)
1	10	MANAGER	2450
2	20	ANALYST	3000
3	20	MANAGER	2975
4	30	MANAGER	2850