

< 다중행 함수와 데이터 그룹화 >

• 하나의 열에 출력 결과를 담는 다중행 함수

그룹 함수 또는 복수행 함수라고도 불리는 다중행 함수. 여러 행을 바탕으로 하나의 결과 값을 도출해 내기 위해 사용하는 함수. 다중행 함수를 사용한 SELECT절에는 기본적으로 여러 행이 결과로 나올 수 있는 열(함수 : 연산자가 사용된 데이터도 포함)을 함께 사용할 수 없음.

※ ORA-00937 오류 : SELECT절에 다중행 함수를 사용하여 결과 값이 한 행으로 나온 데이터와 여러 행이 나올 수 있는 데이터를 함께 명시했을 때 발생

자주 사용하는 다중행 함수들

함수	설명
SUM	지정한 데이터의 합 반환
COUNT	지정한 데이터의 개수 반환
MAX	지정한 데이터 중 최댓값 반환
MIN	지정한 데이터 중 최솟값 반환
AVG	지정한 데이터의 평균값 반환

- 합계를 구하는 SUM 함수 : 합계를 구할 데이터를 지정함.

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

- 데이터 개수를 구해주는 COUNT 함수 : 결과 행의 개수를 출력함.

COUNT([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)]
[개수를 구할 열이나 연산자, 함수를 사용한 데이터(필수)])

- 최댓값과 최솟값을 구하는 MAX, MIN 함수

MAX([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)]
[최댓값을 구할 열이나 연산자, 함수를 사용한 데이터(필수)])

MIN([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)]
[최솟값을 구할 열이나 연산자, 함수를 사용한 데이터(필수)])

• 날짜 데이터에 MAX, MIN 함수 사용하기

- 평균값을 구하는 AVG 함수

AVG([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)]
[평균값을 구할 열이나 연산자, 함수를 사용한 데이터(필수)])

• 결과 값을 원하는 열로 묶어 출력하는 GROUP BY절

DEPT 열 값별로 급여의 평균값을 구하려면 각 부서 평균 값을 구하기 위해 SELECT문을 하나하나 제작해야 함.

예) SELECT AVG(SAL) FROM EMP WHERE DEPTNO = 10;

여러 데이터에서 의미 있는 하나의 결과를 특정 열 값별로 묶어서 출력할 때 데이터를 '그룹화'한다고 표현함. SELECT문에서는 GROUP BY절을 작성하여 데이터를 그룹화할 수 있는데 다음과 같이 순서에 맞게 작성하며 그룹으로 묶을 기준 열을 지정함.

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)]
ORDER BY	[정렬하려는 열 지정]

GROUP BY : 특정 열 또는 데이터를 기준으로 데이터를 그룹으로 묶음.

GROUP BY절을 사용할 때 유의점 : 다중행 함수를 사용하지 않은 일반 열은 GROUP BY절에 명시하지

않으면 select절에서 사용할 수 없다는 것.

- GROUP BY절에 조건을 줄 때 사용하는 HAVING절

HAVING절은 SELECT문에 GROUP BY절이 존재할 때만 사용할 수 있음. GROUP BY절을 통해 그룹화된 결과 값의 범위를 제한하는데 사용함.

예제) 각 부서의 직책별 평균 급여를 구하되 그 평균 급여가 2000이상인 그룹만 출력하시오.

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)]
HAVING	[출력 그룹을 제한하는 조건식]
ORDER BY	[정렬하려는 열 지정]

HAVING : GROUP BY절을 사용해 그룹화된 결과 중 출력 그룹을 선별하는 조건식을 지정함.

HAVING절을 사용할 때 유의점 : 조건식을 지정한다는 점에서 HAVING절이 WHERE절과 비슷하다고 생각할 수 있음. HAVING절도 WHERE절처럼 지정한 조건식이 참인 결과만 출력한다는 점에서 비슷한 부분이 있음. 하지만 WHERE절은 출력 대상 행을 제한하고, HAVING절은 그룹화된 대상을 출력에서 제한하므로 쓰임새는 전혀 다르다는 것. 만약 출력 결과를 제한하기 위해 HAVING을 사용하지 않고 조건식을 WHERE절에 명시하면 SELECT문이 실행되지 않고 오류가 발생함.

예제) HAVING절을 사용하여 EMP 테이블의 부서별 직책의 평균 급여가 500이상인 직원들의 부서 번호, 직책, 부서별 직책의 평균 급여가 출력되도록 하시오.

- 그룹화와 관련된 여러 함수

- ROLLUP, CUBE, GROUPING SETS 함수

GROUP BY절에 지정할 수 있는 특수 함수. ROLLUP함수와 CUBE함수는 그룹화 데이터의 합계를 출력할 때 유용하게 사용할 수 있음.

SELECT	[조회할 열1 이름], [열2 이름], ..., [열n 이름]
FROM	[조회할 테이블 이름]
WHERE	[조회할 행을 선별하는 조건식]
GROUP BY ROLLUP	[그룹화 열 지정(여러 개 지정 가능)];

SELECT	[조회할 열1 이름], [열2 이름], ..., [열n 이름]
FROM	[조회할 테이블 이름]
WHERE	[조회할 행을 선별하는 조건식]
GROUP BY CUBE	[그룹화 열 지정(여러 개 지정 가능)];

ROLLUP(A, B, C)

1. A 그룹별 B 그룹별 C 그룹에 해당하는 결과 출력
2. A 그룹별 B 그룹에 해당하는 결과 출력
3. A 그룹에 해당하는 결과 출력
4. 전체 데이터 결과 출력

CUBE(A, B, C)

1. A 그룹별 B 그룹별 C 그룹에 해당하는 결과 출력
2. A 그룹별 B 그룹에 해당하는 결과 출력
3. B 그룹별 C 그룹에 해당하는 결과 출력
4. A 그룹별 C 그룹에 해당하는 결과 출력
5. A 그룹 결과
6. B 그룹 결과
7. C 그룹 결과
8. 전체 데이터 결과

예제1) DEPTNO를 먼저 그룹화한 후 ROLLUP 함수에 JOB 지정하기

예제2) JOB을 먼저 그룹화한 후 ROLLUP 함수에 DEPTNO 지정하기

```
SELECT      [조회할 열1 이름], [열2 이름], ..., [열n 이름]
FROM        [조회할 테이블 이름]
WHERE       [조회할 행을 선별하는 조건식]
GROUP BY GROUPING SETS [그룹화 열 지정(여러 개 지정 가능)];
```

GROUPING SETS : 같은 수준의 그룹화 열이 여러 개일 때 각 열별 그룹화를 통해 결과 값을 출력하는데 사용. 여러 그룹화 대상 열의 결과 값을 각각 같은 수준으로 출력함.

• 그룹화 함수

데이터 자체의 가공이나 특별한 연산 기능을 수행하지는 않지만 그룹화 데이터의 식별이 쉽고 가독성을 높이기 위한 목적으로 사용함.

- GROUPING 함수

ROLLUP 또는 CUBE 함수를 사용한 GROUP BY절에 그룹화 대상으로 지정한 열이 그룹화된 상태로 결과가 집계되었는지 확인하는데 사용함. GROUP BY절에 명시된 열 중 하나를 지정할 수 있음.

```
SELECT      [조회할 열1 이름], [열2 이름], ..., [열n 이름]
            GROUPING [GROUP BY절에 ROLLUP 또는 CUBE에 명시된 그룹화 할 열 이름]
FROM        [조회할 테이블 이름]
WHERE       [조회할 행을 선별하는 조건식]
GROUP BY ROLLUP 또는 CUBE [그룹화 할 열];
```

GROUPING : 현재 결과가 그룹화 대상 열의 그룹화가 이루어진 상태의 집계인지 여부를 출력함.

- GROUPING_ID 함수

GROUPING 함수와 마찬가지로 ROLLUP 또는 CUBE 함수로 연산할 때 특정 열이 그룹화 되었는지를 출력하는 함수. 그룹화 여부를 검사할 열을 하나씩 지정하는 GROUPING 함수와 달리 GROUPING_ID 함수는 한 번에 여러 열을 지정할 수 있음.

```
SELECT      [조회할 열1 이름], [열2 이름], ..., [열n 이름]
            GROUPING_ID [그룹화 여부를 확인할 열(여러 개 지정 가능)]
FROM        [조회할 테이블 이름]
WHERE       [조회할 행을 선별하는 조건식]
GROUP BY ROLLUP 또는 CUBE [그룹화 할 열];
```

GROUPING_ID : GROUPING 함수처럼 특정 열의 그룹화 여부를 출력할 수 있으며, 검사할 열을 여러 개 지정할 수 있음.

GROUPING_ID 함수를 사용한 결과는 그룹화 비트 벡터 값으로 나타냄. GROUPING_ID(a, b)와 같이 열을 두 개 지정한다면 출력 결과는 다음과 같음.

그룹화 된 열	그룹화 비트 벡터	최종 결과
a, b	0 0	0
a	0 1	1
b	1 0	2
없음	1 1	3

각 열의 그룹화 유무에 따라 0과 1이 결과 값으로 나오는 것은 GROUPING과 같음. 하지만 GROUPING_ID 함수는 한 번에 여러 개 열을 지정할 수 있으므로 지정한 열의 순서에 따라 0, 1 값이 하나씩 출력됨. 이렇게 0과 1로 구성된 그룹화 비트 벡터 값을 2진수로 보고 10진수로 바꾼 값이 최종 결과로 출력됨.

• LISTAGG 함수

오라클 11g 버전부터 사용할 수 있는 함수. 그룹에 속해 있는 데이터를 가로로 나열할 때 사용함.

SELECT	[조회할 열1 이름], [열2 이름], ..., [열n 이름]
	LISTAGG([나열할 열(필수)], [각 데이터를 구분하는 구분자(선택)])
	WITHIN GROUP(ORDER BY 나열할 열의 정렬 기준 열(선택))
FROM	[조회할 테이블 이름]
WHERE	[조회할 행을 선별하는 조건식];

키워드	필수 요소	선택 요소	설명
LISTAGG ~ WITHIN GROUP	나열할 열	각 데이터를 구분하는 구분자, 지정하지 않을 경우 NULL이 기본값이 됨.	그룹화 데이터를 하나의 열에 가로로 나열하여 출력하는데 사용함.

• PIVOT, UNPIVOT 함수

오라클 11g 버전부터 사용할 수 있는 함수. PIVOT 함수는 기존 테이블 행을 열로 바꾸고 UNPIVOT 함수는 기존 테이블 열을 행으로 바꿔서 출력함.

실습1) EMP 테이블을 이용하여 부서번호(DEPTNO), 평균 급여(AVG_SAL), 최고 급여(MAX_SAL), 최저 급여(MIN_SAL), 직원 수(CNT)를 출력합니다. 단 평균 급여를 출력할 때 소수점을 제외하고 각 부서번호별로 출력하세요.

```
SELECT DEPTNO,
       TRUNC(AVG(SAL)) AS AVG_SAL,
       MAX(SAL) AS MAX_SAL,
       MIN(SAL) AS MIN_SAL,
       COUNT(*) AS CNT
FROM EMP
GROUP BY DEPTNO;
```

실습2) 같은 직책(JOB)에 종사하는 직원이 3명 이상인 직책과 인원수를 출력하세요.

```
SELECT JOB, COUNT(*)
FROM EMP
GROUP BY JOB
HAVING COUNT(*) >= 3;
```

실습3) 직원들의 입사 연도(HIRE_YEAR)를 기준으로 부서별로 몇 명이 입사했는지 출력하세요.

```
SELECT TO_CHAR(HIREDATE, 'YYYY') AS HIRE_YEAR,
       DEPTNO,
       COUNT(*) AS CNT
FROM EMP
GROUP BY TO_CHAR(HIREDATE, 'YYYY'), DEPTNO;
```

실습4) 추가수당(COMM)을 받는 직원수와 받지 않는 직원수를 출력하세요.

```
SELECT NVL2(COMM, 'O', 'X') AS EXIST_COMM,
       COUNT(*) AS CNT
FROM EMP
GROUP BY NVL2(COMM, 'O', 'X');
```

실습5) 각 부서의 입사 연도별 인원 수, 최고 급여, 급여 합, 평균 급여를 출력하고 각 부서별 소계와 총계를 출력하세요.

```
SELECT DEPTNO,  
       TO_CHAR(HIREDATE, 'YYYY') AS HIRE_YEAR,  
       COUNT(*) AS CNT,  
       MAX(SAL) AS MAX_SAL,  
       SUM(SAL) AS SUM_SAL,  
       AVG(SAL) AS AVG_SAL  
FROM EMP  
GROUP BY ROLLUP(DEPTNO, TO_CHAR(HIREDATE, 'YYYY'));
```