

## 그룹 함수

그룹 함수



## 그룹 함수

## 1. 그룹 함수의 개요

## 1.1 그룹 함수란?

- ☐ 그룹 함수는 단일 행 함수와는 달리 행들의 집합에 대해 연산한 후 단 하나의 결과를 반환

DEPTNO	SAL
10	5000
20	2975
30	2850
10	2450
20	3000
20	3000
30	1600
30	1250
30	1250
30	1500
30	950
10	1300
20	800
20	1100

“EMP 테이블에서  
sal의 최대값”

MAX (SAL)
5000

## 1.2 그룹 함수의 종류

함 수	의 미
<b>SUM</b> ( [DISTINCT   <u>ALL</u> ] <i>n</i> )	<i>n</i> 의 합계이며 NULL 값을 무시
<b>AVG</b> ( [DISTINCT   <u>ALL</u> ] <i>n</i> )	<i>n</i> 의 평균이며 NULL 값을 무시
<b>COUNT</b> ( * )	레코드 개수를 출력하며, *를 사용하면 중복 행 및 NULL 값을 가진 행을 포함하여 선택한 행을 모두 계산
<b>COUNT</b> ( [DISTINCT   <u>ALL</u> ] <i>expr</i> )	레코드 개수를 출력하며, <i>expr</i> 는 NULL을 제외한 값을 계산
<b>MAX</b> ( [DISTINCT   <u>ALL</u> ] <i>expr</i> )	<i>expr</i> 의 최대값이며 NULL 값을 무시
<b>MIN</b> ( [DISTINCT   <u>ALL</u> ] <i>expr</i> )	<i>expr</i> 의 최소값이며 NULL 값을 무시

### 1.3 그룹 함수의 특징

- ❑ **DISTINCT**를 지정하면 함수는 중복되지 않는 값만 검토하고, **ALL**을 지정하면 중복 값을 포함한 모든 값을 검토 (**기본값은 ALL**)
- ❑ **COUNT (\*)**를 제외한 모든 그룹 함수는 널 (**NULL**) 값을 무시
- ❑ **GROUP BY**절을 사용하면 **Oracle Server**가 결과 집합을 암시적으로 오름차순으로 정렬하는데 기본 순서를 무시하고 내림차순으로 정렬하려면 **ORDER BY**절에 **DESC**를 사용
- ❑ 구문

```
SELECT      [column,] group_function(column), ...  
FROM        table  
[WHERE      condition]  
[GROUP BY   column]  
[ORDER BY   column];
```

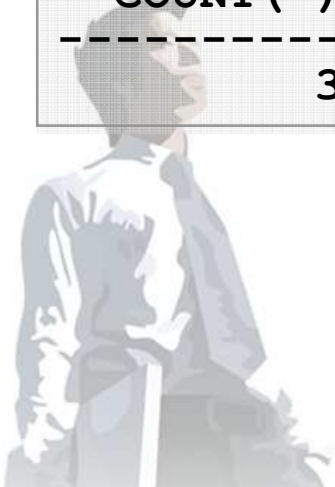
## 2. 그룹 함수의 활용

### 2.1 COUNT

- ☐ COUNT(\*)는 열에 NULL 값 및 중복 행 값을 포함한 행 수 반환
- ☐ COUNT(expr)은 expr로 식별되는 열에서 NULL 값을 제외한 행 수 반환
- ☐ COUNT(DISTINCT expr)은 NULL 값 및 중복 행 값을 제외한 행 수 반환

```
SQL> SELECT COUNT(*)  
2 FROM emp  
3 WHERE deptno = 10;
```

COUNT (*)
3



## 2.1 COUNT (계속)

```
SQL> SELECT COUNT (*)  
2 FROM emp ;
```

```
COUNT (*)  
-----  
14
```

```
SQL> SELECT COUNT (comm)  
2 FROM emp ;
```

```
COUNT (*)  
-----  
4
```

## 2.1 COUNT (계속)

```
SQL> SELECT COUNT(deptno)
      2 FROM emp;
```

```
COUNT (*)
-----
          14
```

```
SQL> SELECT COUNT(DISTINCT deptno)
      2 FROM emp;
```

```
COUNT (*)
-----
          3
```

## 2.2 AVG 및 SUM

- ☐ AVG 함수는 평균 값을 출력하고, SUM 함수는 합계 값을 출력
- ☐ AVG와 SUM 함수는 숫자 데이터 유형에서만 사용 가능
- ☐ NULL 값 무시

```
SQL> SELECT SUM(sal), AVG(sal)
2 FROM emp
3 WHERE job = 'SALESMAN';
```

SUM(SAL)	AVG(SAL)
5600	1400





## 2.3 MAX 및 MIN

- ☐ MAX 함수는 최대 값을 출력하고, MIN 함수는 최소 값 출력
- ☐ MAX 함수와 MIN 함수는 모든 데이터 유형에서 사용 가능
- ☐ NULL 값 무시

```
SQL> SELECT  MIN(hiredate) , MAX(hiredate)
           2  FROM      emp;
```

```
MIN(HIRE  MAX(HIRE
-----  -
80/12/17  83/01/12
```



## 2.4 그룹 함수와 NULL 값

- ☐ COUNT (\*)를 제외한 그룹 함수는 해당 열의 NULL 값 무시
- ☐ NVL 함수는 그룹 함수가 NULL 값을 포함하도록 강제로 지정

```
SQL> SELECT  AVG(comm) , AVG(NVL(comm, 0))
          2  FROM    emp;
```

```
AVG (COMM)  AVG (NVL (COMM, 0))
-----
          550          157.142857
```

```
SQL> SELECT  COUNT(comm) , COUNT(NVL(comm, 0))
          2  FROM    emp;
```

```
COUNT (COMM)  COUNT (NVL (COMM, 0))
-----
          4          14
```

## 데이터 그룹화 (GROUP BY)

데이터 그룹화 (GROUP BY)



## 데이터 그룹화 (GROUP BY)

## 1. GROUP BY 절

## 1.1 개요

- ☐ 테이블 정보를 더 작은 그룹으로 나누어야 할 경우 **GROUP BY** 절을 사용

DEPTNO	SAL		DEPTNO	AVG (SAL)
-----	-----		-----	-----
10	1300	2916.66667	10	2916.66667
10	2450			
10	5000			
20	800	2175	20	2175
20	3000			
20	3000			
20	1100			
20	2975			
30	950	1566.66667	30	1566.66667
30	1500			
30	1250			
30	1250			
30	2850			
30	1600			

“EMP  
테이블에서  
각 부서에  
대한  
평균 급여”

## 데이터 그룹화 (GROUP BY)

## 1.2 GROUP BY 절 사용시 고려사항

- SELECT 절에 있는 열 목록에서 그룹 함수에 없는 열을 GROUP BY 절에 반드시 지정
- GROUP BY 절에 있는 열 목록은 SELECT 절에 지정하지 않아도 됨
- WHERE 절을 사용하면 그룹별로 구분하기 전에 행을 제한할 수 있음
- GROUP BY 절에는 열 별칭을 사용할 수 없음
- 질의 결과는 GROUP BY 목록에 포함된 열의 오름차순으로 정렬

```
SELECT      [column,] group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   column]
[ORDER BY   column];
```

## 데이터 그룹화 (GROUP BY)

## 1.3 GROUP BY 절 사용

☞ SELECT 절에서 그룹 함수 안에 없는 열들은 GROUP BY 절에 포함

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 GROUP BY deptno;
```

DEPTNO	AVG (SAL)
10	2916.66667
20	2175
30	1566.66667

## 데이터 그룹화 (GROUP BY)

## 1.3 GROUP BY 절 사용 (계속)

- ☞ GROUP BY 절에 있는 열 목록은 SELECT 절에 지정하지 않아도 됨

```
SQL> SELECT  AVG(sal)
      2  FROM    emp
      3  GROUP BY deptno;
```

```
      AVG(SAL)
-----
2916.66667
      2175
1566.66667
```



## 데이터 그룹화 (GROUP BY)

## 1.3 GROUP BY 절 사용 (계속)

- ☐ GROUP BY 절에 하나 이상의 열을 사용하여 그룹과 서브 그룹에 대한 통계 결과 확인

```
SQL> SELECT deptno, job, SUM(sal)
2 FROM emp
3 GROUP BY deptno, job;
```

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	5600



## 데이터 그룹화 (GROUP BY)

## 1.4 GROUP BY 절 오류

- ☞ 그룹 함수가 아닌 **SELECT** 절의 어떤 열이나 표현식은 **GROUP BY** 절에 반드시 있어야 함

```
SQL> SELECT deptno, COUNT(ename)
      2 FROM emp;
```

```
SELECT deptno, COUNT(ename)
```

\*

1행에 오류:

ORA-00937: 단일 그룹의 그룹 함수가 아닙니다



## 데이터 그룹화 (GROUP BY)

## 1.4 GROUP BY 절 오류 (계속)

- ☐ WHERE 절을 사용하여 그룹 제한 불가능
- ☐ 그룹을 제한하기 위해서 HAVING 절 사용

```
SQL> SELECT      deptno, AVG(sal)
2  FROM          emp
3  WHERE          AVG(sal) > 2000
4  GROUP BY      deptno;
```

```
WHERE  AVG(sal) > 2000
      *
```

3행에 오류:

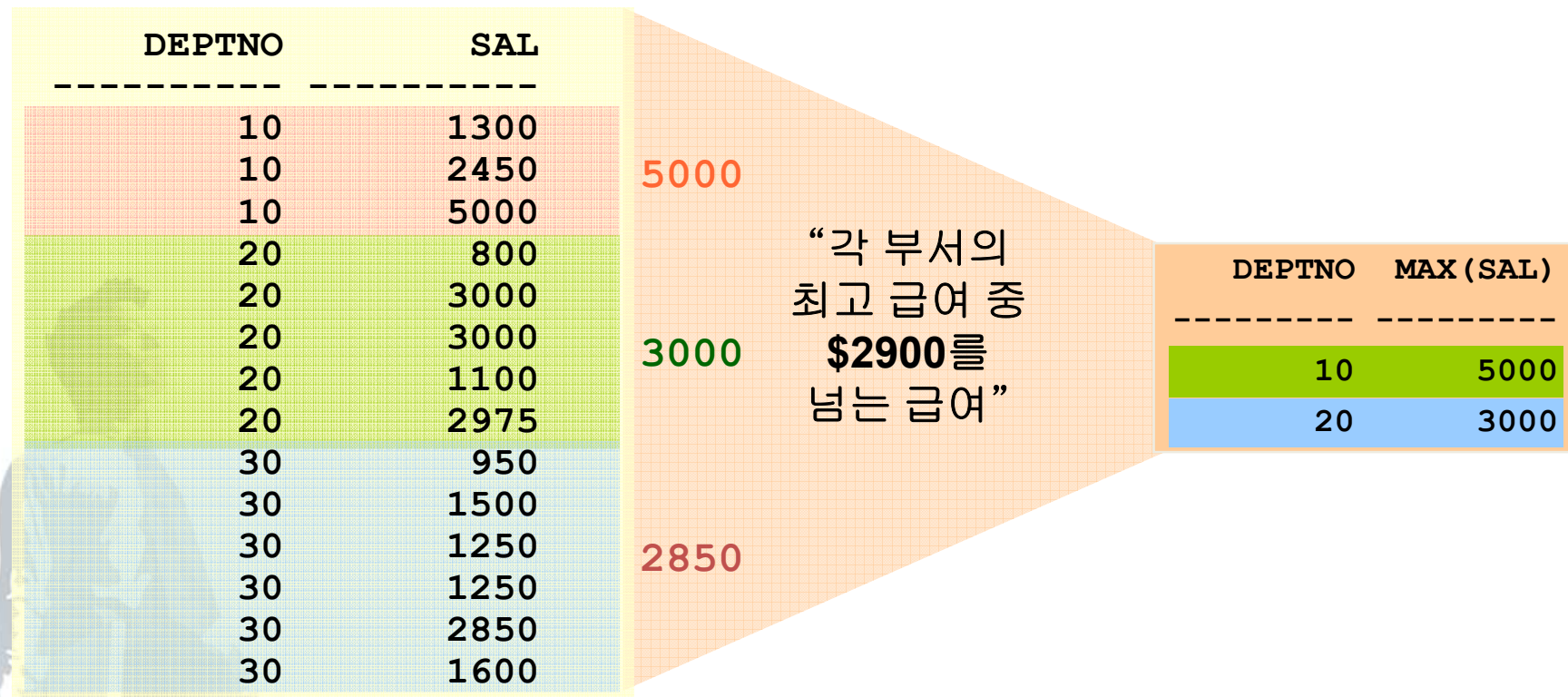
ORA-00934: 그룹 함수는 허가되지 않습니다

## 데이터 그룹화 (GROUP BY)

## 2. HAVING 절

## 2.1 개요

- ☐ WHERE 절을 사용하여 선택할 행을 제한하는 것과 동일한 방식으로 HAVING 절을 사용하여 그룹을 제한



## 데이터 그룹화 (GROUP BY)

## 2.2 HAVING 절 사용시 고려사항

- ☐ HAVING 절을 사용하여 표시할 그룹을 제한
- ☐ 그룹 정보를 바탕으로 그룹을 한층 더 제한
- ☐ 수행 단계
  - 행 그룹화 → 그룹 함수 적용 → HAVING 절과 일치하는 그룹을 표시

```
SELECT      column, group_function
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   column];
```

## 데이터 그룹화 (GROUP BY)

## 2.3 HAVING 절 사용

- ☞ 그룹 함수의 결과를 기반으로 행을 제한할 경우 GROUP BY 절 및 HAVING 절이 모두 있어야 함

```
SQL> SELECT deptno, MAX(sal)
2 FROM emp
3 GROUP BY deptno
4 HAVING max(sal) > 2900;
```

DEPTNO	MAX (SAL)
10	5000
20	3000

## 데이터 그룹화 (GROUP BY)

## 2.3 HAVING 절 사용 (계속)

```
SQL> SELECT      job, SUM(sal)
2  FROM          emp
3  WHERE          job <> 'SALESMAN'
4  GROUP BY      job
5  HAVING         SUM(sal) > 3000
6  ORDER BY      SUM(sal);
```

JOB	SUM(SAL)
CLERK	4150
PRESIDENT	5000
ANALYST	6000
MANAGER	8275

## 데이터 그룹화 (GROUP BY)

**3. 그룹 함수 중첩**

- ☐ 그룹 함수도 중첩이 필요한 경우 중첩 가능
- ☐ 그룹 함수는 두 번까지 중첩 가능

```
SQL> SELECT MAX (AVG (sal))  
2 FROM emp  
3 GROUP BY deptno;
```

```
MAX (AVG (SAL) )  
-----  
2916.66667
```

