

Aggregating Data Using Group Functions

그룹 함수란?

그룹 함수의 종류

그룹 함수 구문

AVG와 SUM 함수

MIN과 MAX 함수

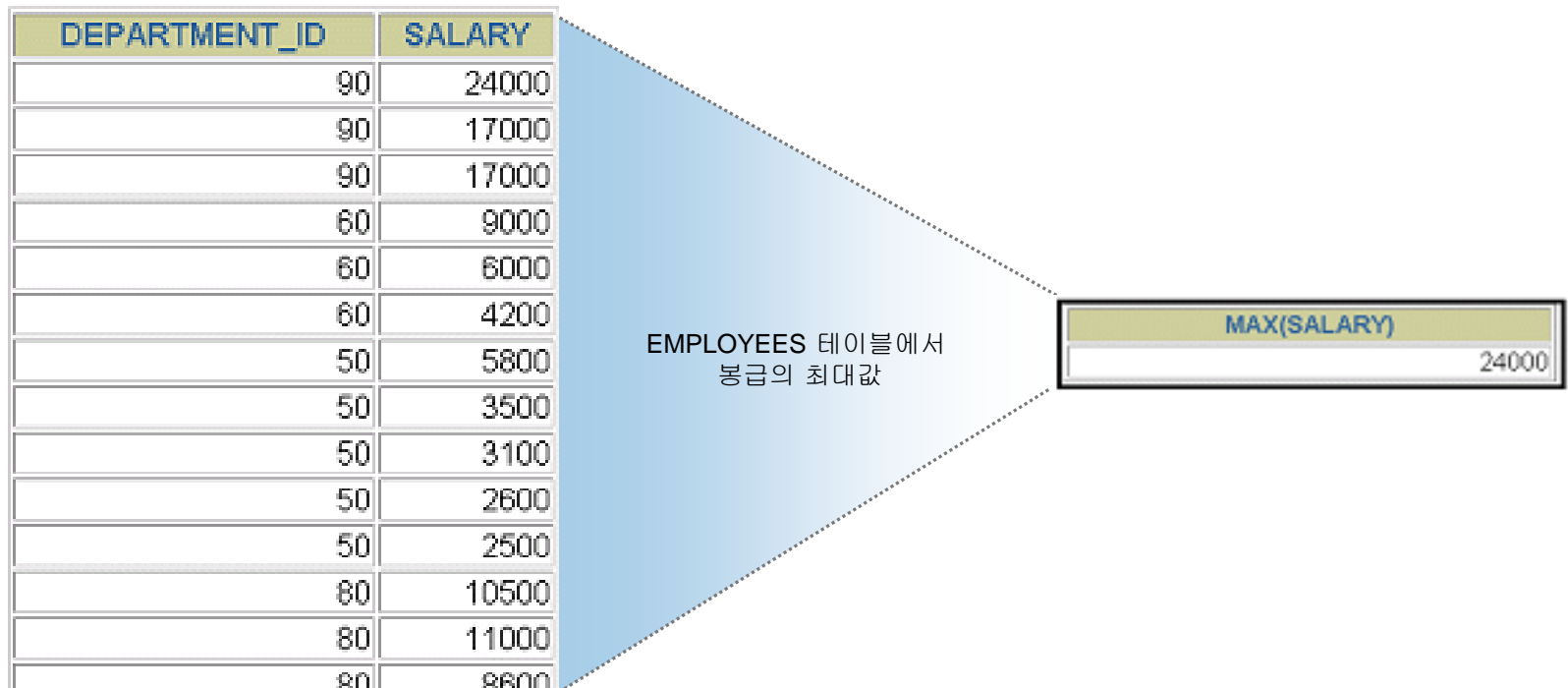
COUNT 함수

GROUP 함수

HAVING 절

그룹 함수란?

- 그룹 함수는 그룹 당 하나의 결과가 주어지도록 행의 집합에 대해 연산한다.



그룹 함수의 종류

Function	Description
AVG([DISTINCT ALL] n)	n의 평균값, null값은 무시
COUNT({* [DISTINCT ALL] expr})	row의 수. *: null 을 포함한 전체 행 수 expr: null 값 무시
MAX([DISTINCT ALL] expr)	최대 값, null값은 무시
MIN(DISTINCT ALL] expr)	최소 값, null값은 무시
STDDEV([DISTINCT ALL] n)	n의 표준 편차, null값은 무시
SUM([DISTINCT ALL] n)	n의 합계, null값은 무시
VARIANCE([DISTINCT ALL] n)	n의 분산, null값은 무시

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

AVG 및 SUM 함수 사용

- 숫자 데이터에 대하여 AVG와 SUM을 사용할 수 있다.

```
SELECT AVG(salary), MAX(salary), MIN(salary), SUM(salary)
FROM employees
WHERE job_id LIKE '%REP%';
```

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8150	11000	6000	32600

MIN 및 MAX 함수 사용

- 모든 데이터 형식에 대하여 MIN과 MAX를 사용할 수 있다.

```
SELECT MIN(hire_date), MAX(hire_date)
FROM employees;
```

MIN(HIRE	MAX(HIRE
87/06/17	00/01/29

COUNT 함수 사용

- COUNT(*)는 테이블에 있는 행의 수를 반환한다.

```
SELECT COUNT(*)  
FROM employees  
WHERE department_id = 50;
```

COUNT(*)
5

- COUNT(expr)은 expr에 대해 널이 아닌 값을 가진 행수를 반환합니다.
- EMPLOYEES 테이블에서 널 값을 제외한 부서 값의 수를 표시합니다.

```
SELECT COUNT(commission_pct)  
FROM employees  
WHERE department_id = 80;
```

COUNT(COMMISSION_PCT)
3

DISTINCT, ALL 키워드 사용

- 그룹 함수는 기본적으로 ALL 옵션이 적용된다.
- COUNT(DISTINCT expr)은 Null이 아닌 별개의 행의 수를 반환한다.

```
SELECT COUNT(ALL salary), COUNT(DISTINCT salary)
FROM employees;
```

COUNT(ALLSALARY)	COUNT(DISTINCTSALARY)
107	57

그룹 함수 및 널 값

- 그룹 함수는 컬럼에 있는 null 값을 무시한다.

```
SELECT AVG(commission_pct), AVG(NVL(commission_pct, 0))  
FROM employees;
```

AVG(COMMISSION_PCT)	AVG(NVL(COMMISSION_PCT,0))
.222857142857142857142857142857143	.072897196261682242990654205607476635514

데이터 그룹 생성 : GROUP BY 절 구분

- GROUP BY 절의 사용으로 테이블에 있는 행들은 더 작은 그룹으로 나뉘어진다.

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

GROUP BY 절 사용

- 그룹 함수가 아닌 SELECT 리스트의 모든 컬럼은 GROUP BY 절 안에 있어야 한다.

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id;
```

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

여러 열에 GROUP BY 절 사용

```
SELECT department_id, job_id, AVG(salary)
FROM employees
GROUP BY department_id, job_id;
```

DEPARTMENT_ID	JOB_ID	AVG(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
30	PU_CLERK	2780
30	PU_MAN	11000
40	HR_REP	6500
50	SH_CLERK	3215
50	ST_CLERK	2785
50	ST_MAN	7280

그룹 함수를 사용한 잘못된 질의

- SELECT 목록의 열 또는 표현식 중 그룹 함수가 아닌 것은 GROUP BY 절에 포함시켜야 합니다.

```
SELECT department_id, COUNT(last_name)
FROM employees;
```

```
SELECT department_id, COUNT(last_name) *
1행에 오류:
ORA-00937: 단일 그룹의 그룹 함수가 아닙니다
```

GROUP BY절에 열이 없습니다.

그룹 함수를 사용한 잘못된 질의

- 그룹을 제한하기 위하여 WHERE 절을 사용할 수 없다.
- 그룹을 제한하기 위하여 HAVING 절을 사용할 수 있다.
- WHERE 절에 그룹 함수를 사용할 수 없다.

```
SELECT department_id, AVG(salary)
FROM employees
WHERE AVG(salary) > 8000
GROUP BY department_id;
```

WHERE AVG(salary) > 8000 *

3행에 오류:

ORA-00934: 그룹 함수는 허가되지 않습니다

WHERE절을 사용하여 그룹을 제한할 수 없습니다.

그룹 결과 제외 : HAVING 절

- 그룹을 제한하기 위하여 HAVING 절의 사용 :
 1. 행들 분류
 2. 그룹 함수 적용
 3. HAVING 절과 일치하는 그룹들 표시

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

HAVING 절 사용

```
SELECT department_id, MAX(salary)
FROM employees
GROUP BY department_id
HAVING MAX(salary) > 10000;
```

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

HAVING 절 사용

```
SELECT job_id, SUM(salary) PAYROLL
FROM employees
WHERE job_id NOT LIKE '%REP%'
GROUP BY job_id
HAVING SUM(salary) > 13000
ORDER BY SUM(salary);
```

JOB_ID	PAYROLL
IT_PROG	19200
AD_PRES	24000
AD_VP	34000

Group 함수 중첩

- 부서 별 평균 봉급의 최대값 표시

```
SELECT MAX(AVG(salary))  
FROM employees  
GROUP BY department_id;
```

MAX(AVG(SALARY))	
	19333.3333

실습 (아래 테이블은 단순 예시임)

- 모든 사원의 급여 최고액, 최저액, 총액 및 평균액을 표시하십시오. 열 레이블을 각각 **Maximum**, **Minimum**, **Sum** 및 **Average**로 지정하고 결과를 정수로 반올림 하십시오.

Maximum	Minimum	Sum	Average
24000	2500	175500	8775

- 1번을 수정하여 각 업무 유형별로 표시하십시오.

JOB_ID	Maximum	Minimum	Sum	Average
AC_ACCOUNT	8300	8300	8300	8300
AC_MGR	12000	12000	12000	12000
AD_ASST	4400	4400	4400	4400
AD_PRES	24000	24000	24000	24000
AD_VP	17000	17000	17000	17000

- 업무가 동일한 사원 수를 표시하는 질의를 작성하십시오.

JOB_ID	COUNT(*)
AC_ACCOUNT	1
AC_MGR	1
AD_ASST	1

실습 (아래 테이블은 단순 예시임)

4. 관리자는 나열하지 말고 관리자 수를 확인하십시오. 열 레이블은 **Number of Managers**로 지정하십시오.

Number of Managers
8

5. 최고 급여와 최저 급여의 차액을 표시하는 질의를 작성하고 열 레이블을 **DIFFERENCE**로 지정하십시오.

DIFFERENCE
21500

6. 관리자 번호 및 해당 관리자에 속한 사원의 최저 급여를 표시하십시오. 관리자를 알 수 없는 사원 및 최저 급여가 \$6,000 미만인 그룹은 제외시키고 결과를 급여에 대한 내림차순으로 정렬하십시오.

MANAGER_ID	MIN(SALARY)
102	9000
105	8200

실습 (아래 테이블은 단순 예시임)

7. 각 부서에 대해 부서 이름, 위치, 인원 수, 부서 내 모든 사원의 평균 급여를 표시하는 질의를 작성하고, 열 레이블을 각각 **Name**, **Location**, **Number of People** 및 **Salary** 로 지정하십시오. 평균 급여는 소수점 둘째 자리로 반올림하십시오.

Name	Location	Number of People	Salary
Accounting	1700	2	10150
Administration	1700	1	4400
Executive	1700	3	10222.22

8. 총 인원 수 및 2005, 2006, 2007, 2008년에 입사한 인원 수를 표시하는 질의를 작성하고 적합한 열 머리글을 작성하십시오.

TOTAL	1995	1996	1997	1998
20	1	2	2	3

9. 업무를 표시한 다음 해당 업무에 대해 급여 총액과 부서 별 급여 (20, 50, 80, 90)의 총액을 각각 표시하는 형렬 질의를 작성하고 각 열에 적합한 머리글을 지정하십시오.

Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
AC_ACCOUNT					8300
AC_MGR					12000
AD_ASST					4400
AD PRES				24000	24000
AD_VP				34000	34000
IT PROG					10200