

# 그룹 함수

# 그룹 함수

## ❖ 그룹 함수의 필요성

```
SELECT department_id, ROUND(salary, 3)
FROM employees;
```

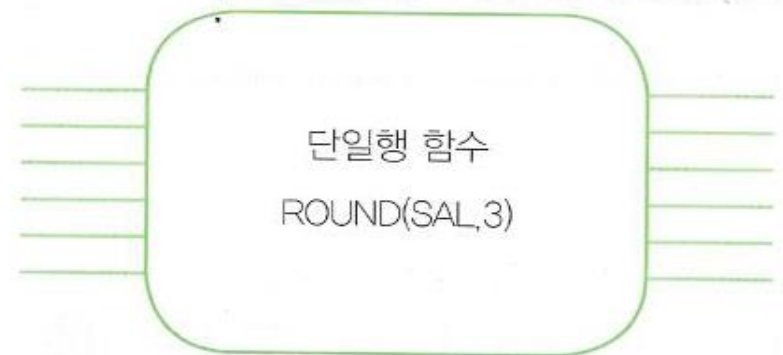
→ 단일행 함수

### ○ 사원의 총 급여 계산

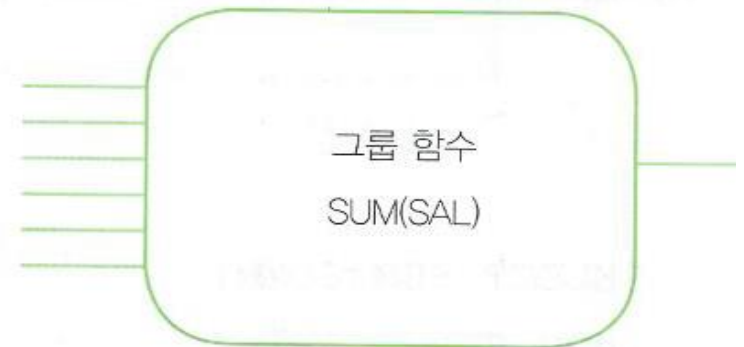
```
SELECT SUM(salary)
FROM employees;
```

→ 691416

14개의 행에 대해서 단일행 함수의 결과는 14개의 행으로 구해진다.



14개의 행에 대해서 그룹 함수의 결과는 1개의 행으로 구해진다.



# 그룹 함수

---

## ❖ 그룹 함수

구분	설명
SUM	그룹의 누적 합계
AVG	그룹의 평균
COUNT	그룹의 총 개수
MAX	그룹의 최대값
MIN	그룹의 최소값
STDDEV	그룹의 표준편차
VARIANCE	그룹의 분산

# 그룹 함수

---

## ❖ SUM 함수

```
SELECT SUM(salary)
FROM employees;
```

→ 691416

## ❖ AVG 함수

```
SELECT AVG(salary)
FROM employees;
```

## ❖ MAX, MIN 함수

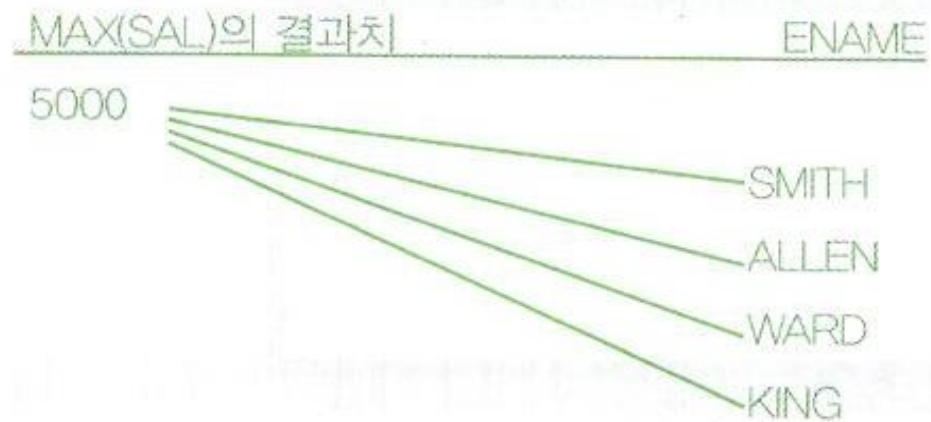
```
SELECT MAX(salary), MIN(salary)
FROM employees;
```

# 그룹 함수

## ❖ 그룹 함수와 단순 컬럼

```
SELECT last_name, MAX(salary)  
FROM employees;
```

→ 에러 발생



# 그룹 함수

---

## ❖ COUNT 함수

- 로우의 개수를 반환

```
SELECT COUNT(commission_pct)
FROM employees;
```

→ 35

```
SELECT COUNT(*), COUNT(commission_pct)
FROM employees;
```

→ 107      35

# 그룹 함수

---

## ❖ COUNT 함수

```
SELECT COUNT(job_id)  
FROM employees;
```

→ 107

```
SELECT COUNT(DISTINCT job_id)  
FROM employees;
```

→ 19

# 그룹 함수

---

## ❖ 실습

- 가장 최근에 입사한 사원의 입사일과 입사한지 가자 오래된 사원의 입사일을 출력하는 쿼리문을 작성하세요
- 80번 부서 소속의 사원중에서 커미션을 받는 사원의 수를 구하시오



# GROUP BY 절

---

## ❖ GROUP BY 절

- 어떤 컬럼을 기준으로 그룹함수를 적용해 줄지를 기술해 주어야 할 경우 GROUP BY 절 추가하고 해당 컬럼을 기술
- 형식

```
SELECT 컬럼명, 그룹함수  
FROM 테이블명  
WHERE 조건(연산자)  
GROUP BY 컬럼명
```

```
SELECT department_id  
FROM employees  
GROUP BY department_id;
```

# GROUP BY 절

---

## ❖ GROUP BY 절

- 부서별 급여 평균 구하기

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

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

# GROUP BY 절

---

## ❖ GROUP BY 절

- SELECT 컬럼명은 GROUP BY절에서 명시한 컬럼과 관련 없는 경우 에러

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

# GROUP BY 절

---

## ❖ GROUP BY 절

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

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

# GROUP BY 절

---

## ❖ GROUP BY 절

```
SELECT department_id, COUNT(*) , COUNT(commission_pct)
FROM employees
GROUP BY department_id;
```

→ 부서로 그룹지어 총 부서 직원수와 커미션을 받는 직원 수 출력

70	1	0
110	2	0
50	45	0
80	34	34

# HAVING 조건

---

## ❖ HAVING 조건

- SELECT절에 조건을 제한할 때는 WHERE절을 사용하며, 그룹의 결과를 제한할 때는 HAVING절을 사용
- 부서별로 그룹을 지은 후(GROUP BY), 그룹 지어진 부서별 평균 급여가 5000 이상인 (HAVING) 부서 번호와 부서별 평균 급여를 출력

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
HAVING AVG(salary) > 5000;
```

# HAVING 조건

---

## ❖ HAVING 조건

- 부서의 최대값과 최소값을 구하되 최대 급여가 6000 이상인 부서만 출력

```
SELECT department_id, MAX(salary), MIN(salary)
FROM employees
GROUP BY department_id
HAVING MAX(salary) > 6000;
```