GROUP BY & HAVING

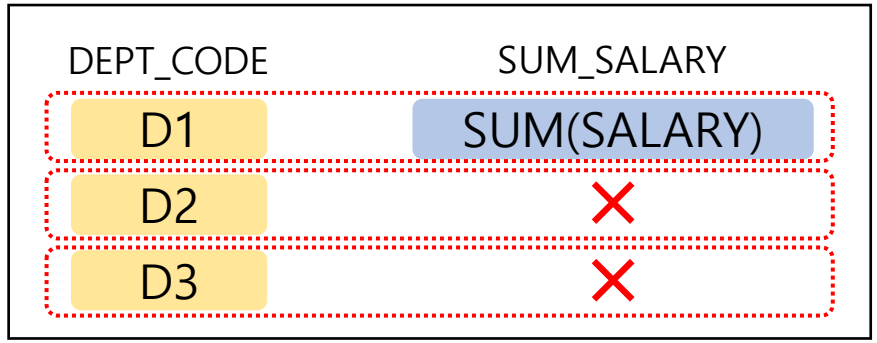
▶ GROUP BY

그룹 함수는 단 한 개의 결과 값만 산출하기 때문에 그룹이 여러 개일 경우 오류 발생

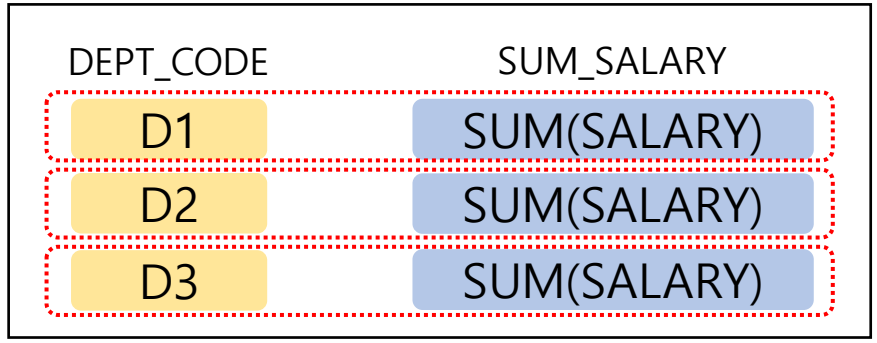
여러 개의 결과 값을 산출하기 위해 그룹 함수가 적용될 그룹의 기준을 GROUP BY절에 기술하여 사용

```
SELECT DEPT_CODE,
       SUM(SALARY)
FROM EMPLOYEE;
```

** 에러 발생




```
SELECT DEPT_CODE,
       SUM(SALARY)
FROM EMPLOYEE
GROUP BY DEPT_CODE;
```



▶ GROUP BY

✓ 예시

- EMPLOYEE테이블에서 부서코드, 그룹 별 급여의 합계, 그룹 별 급여의 평균(정수처리), 인원 수를 조회하고 부서 코드 순으로 정렬

```
SELECT DEPT_CODE 부서코드,
       SUM(SALARY) 합계,
       FLOOR(AVG(SALARY)) 평균,
       COUNT(*) 인원수
FROM EMPLOYEE
GROUP BY DEPT_CODE
ORDER BY DEPT_CODE ASC;
```

부서코드	합계	평균	인원수
D1	7820000	2606666	3
D2	6520000	2173333	3
D5	17660000	2943333	6
D6	10100000	3366666	3
D8	6986240	2328746	3
D9	15800000	5266666	3
(null)	5210000	2605000	2

- EMPLOYEE테이블에서 부서코드와 보너스 받는 사원 수 조회하고 부서코드 순으로 정렬

```
SELECT DEPT_CODE 부서코드, COUNT(BONUS) 인원수
FROM EMPLOYEE
GROUP BY DEPT_CODE
ORDER BY DEPT_CODE ASC;
```

부서코드	인원수
D1	2
D2	0
D5	2
D6	1
D8	2
D9	1
(null)	1

▶ GROUP BY

✓ 예시

- EMPLOYEE테이블에서 성별과 성별 별 급여 평균(정수처리), 급여 합계, 인원 수 조회하고 인원수로 내림차순 정렬

```
SELECT DECODE(SUBSTR(EMP_NO, 8, 1), 1, '남', 2, '여') 성별,
       FLOOR(AVG(SALARY)) 평균,
       SUM(SALARY) 합계,
       COUNT(*) 인원수
FROM EMPLOYEE
GROUP BY DECODE(SUBSTR(EMP_NO, 8, 1), 1, '남', 2, '여')
ORDER BY 인원수 DESC;
```

성별	평균	합계	인원수
남	3117333	46760000	15
여	2917030	23336240	8

▶ GROUP BY

✓ 예시

- EMPLOYEE테이블에서 부서 코드 별로 같은 직급인 사원의 급여 합계를 조회하고
  부서 코드 순으로 정렬

```
SELECT DEPT_CODE, JOB_CODE, SUM(SALARY)
FROM EMPLOYEE
GROUP BY DEPT_CODE, JOB_CODE
ORDER BY DEPT_CODE;
```

DEPT_CODE	JOB_CODE	SUM(SALARY)
D1	J6	6440000
D1	J7	1380000
D2	J4	6520000
D5	J3	3500000
D5	J5	8460000
D5	J7	5700000
D6	J3	6200000
D6	J4	3900000
D8	J6	6986240
D9	J1	8000000
D9	J2	7800000
(null)	J6	2320000
(null)	J7	2890000

* 여러 컬럼을 그룹으로 묶을 수 있음.

▶ HAVING

그룹 함수로 값을 구해올 그룹에 대해 조건을 설정할 때 HAVING절에 기술

(WHERE절은 SELECT에 대한 조건)

✓ 예시

- 부서 코드와 급여 3000000 이상인 직원의 그룹별 평균 조회

```
SELECT DEPT_CODE, FLOOR(AVG(SALARY)) 평균
FROM EMPLOYEE
WHERE SALARY >= 3000000
GROUP BY DEPT_CODE
ORDER BY 1;
```

DEPT_...	평균
D1	3660000
D5	3653333
D6	3650000
D9	7000000

- 부서 코드와 급여 평균이 3000000 이상인 그룹 조회

```
SELECT DEPT_CODE, FLOOR(AVG(SALARY)) 평균
FROM EMPLOYEE
GROUP BY DEPT_CODE
HAVING FLOOR(AVG(SALARY)) >= 3000000
ORDER BY DEPT_CODE;
```

DEPT_CODE	평균
D6	3366666
D9	5266666

▶ ROLLUP과 CUBE

그룹 별 산출한 결과 값의 집계를 계산하는 함수

✓ 예시

```
SELECT JOB_CODE, SUM(SALARY)
FROM EMPLOYEE
GROUP BY ROLLUP(JOB_CODE)
ORDER BY 1;
```

JOB_CODE	SUM(SALARY)
J1	8000000
J2	7800000
J3	9700000
J4	10420000
J5	8460000
J6	15746240
J7	9970000
(null)	70096240

```
SELECT JOB_CODE, SUM(SALARY)
FROM EMPLOYEE
GROUP BY CUBE(JOB_CODE)
ORDER BY 1;
```

▶ ROLLUP과 CUBE

✓ ROLLUP

인자로 전달받은 그룹 중 가장 먼저 지정한 그룹별로 추가적 집계 결과 반환

✓ 예시

EMPLOYEE 테이블에서 각 부서 마다 직급 별 급여합,
부서 별 급여 합, 전체 직원 급여 총합 조회

```
SELECT DEPT_CODE, JOB_CODE, SUM(SALARY)
FROM EMPLOYEE
GROUP BY ROLLUP(DEPT_CODE, JOB_CODE)
ORDER BY 1;
```

DEPT_CODE	JOB_CODE	SUM(SALARY)
D1	J6	6440000
D1	J7	1380000
D1	(null)	7820000
D2	J4	6520000
D2	(null)	6520000
D5	J3	3500000
D5	J5	8460000
D5	J7	5700000
D5	(null)	17660000
D6	J3	6200000
D6	J4	3900000
D6	(null)	10100000
D8	J6	6986240
D8	(null)	6986240
D9	J1	8000000
D9	J2	7800000
D9	(null)	15800000
(null)	J6	2320000
(null)	J7	2890000
(null)	(null)	5210000
(null)	(null)	70096240

ROLLUP과 CUBE

✓ CUBE

인자로 지정된 그룹들로 가능한 모든 조합 별로 집계한 결과 반환

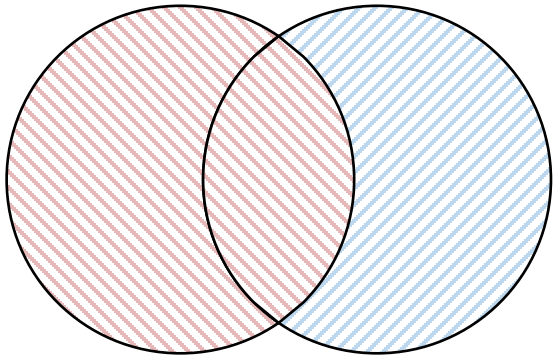
✓ 예시

```
SELECT DEPT_CODE, JOB_CODE, SUM(SALARY)
FROM EMPLOYEE
GROUP BY CUBE(DEPT_CODE, JOB_CODE)
ORDER BY 1;
```

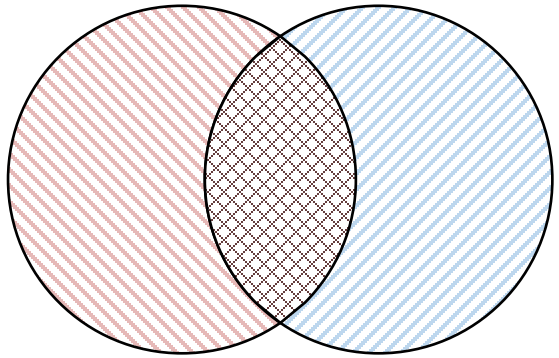
DEPT_CODE	JOB_CODE	SUM(SALARY)
D1	J6	6440000
D1	J7	1380000
D1	(null)	7820000
D2	J4	6520000
D2	(null)	6520000
D5	J3	3500000
D5	J5	8460000
D5	J7	5700000
D5	(null)	17660000
D6	J3	6200000
D6	J4	3900000
D6	(null)	10100000
D8	J6	6986240
D8	(null)	6986240
D9	J1	8000000
D9	J2	7800000
D9	(null)	15800000
(null)	J1	8000000
(null)	J2	7800000
(null)	J3	9700000
(null)	J4	10420000
(null)	J5	8460000
(null)	J6	2320000
(null)	J6	15746240
(null)	J7	2890000
(null)	J7	9970000
(null)	(null)	5210000
(null)	(null)	70096240

▶ 집합 연산자

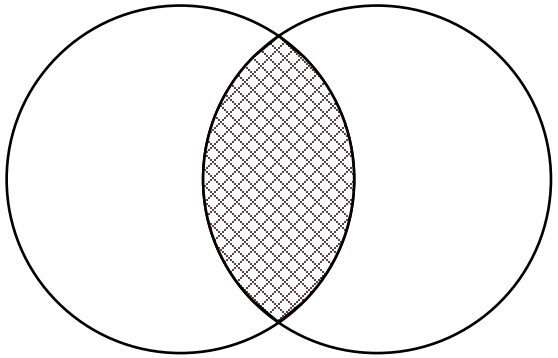
여러 개의 SELECT 결과물을 하나의 쿼리로 만드는 연산자



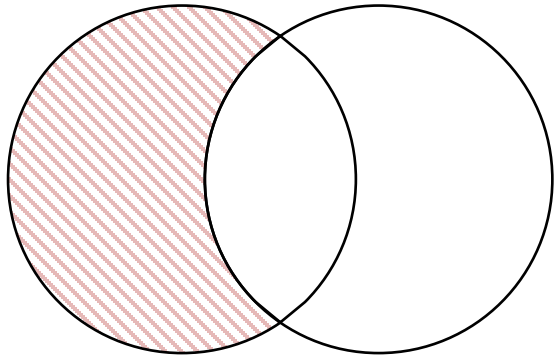
UNION



UNION ALL



INTERSECT



MINUS

▶ 집합 연산자

✓ UNION

여러 개의 쿼리 결과를 합치는 연산자로 중복된 영역은 제외하여 합침

✓ 예시

```
SELECT EMP_ID, EMP_NAME, DEPT_CODE, SALARY
FROM EMPLOYEE
WHERE DEPT_CODE = 'D5'

UNION

SELECT EMP_ID, EMP_NAME, DEPT_CODE, SALARY
FROM EMPLOYEE
WHERE SALARY > 3000000;
```

EMP_ID	EMP_NAME	DEPT_CODE	SALARY
200	선동일	D9	8000000
201	송종기	D9	6000000
203	송은희	D6	3900000
204	유재식	D6	3400000
206	박나라	D5	3700000
207	하이유	D5	2200000
208	김해술	D5	2500000
209	심봉선	D5	3500000
210	윤은해	D5	2000000
215	대북혼	D5	3760000
217	전지연	D1	3660000

▶ 집합 연산자

✓ INTERSECT

여러 개의 SELECT 결과에서 공통된 부분만 결과로 추출(교집합)

✓ 예시

```
SELECT EMP_ID, EMP_NAME, DEPT_CODE, SALARY
FROM EMPLOYEE
WHERE DEPT_CODE = 'D5'

INTERSECT

SELECT EMP_ID, EMP_NAME, DEPT_CODE, SALARY
FROM EMPLOYEE
WHERE SALARY > 3000000;
```

EMP_ID	EMP_NAME	DEPT_CODE	SALARY
206	박나라	D5	3700000
209	심봉선	D5	3500000
215	대북혼	D5	3760000

▶ 집합 연산자

✓ UNION ALL

여러 개의 쿼리 결과를 합치는 연산자로 중복된 영역 모두 포함하여 합침

✓ 예시

```
SELECT EMP_ID, EMP_NAME, DEPT_CODE, SALARY
FROM EMPLOYEE
WHERE DEPT_CODE = 'D5'

UNION ALL

SELECT EMP_ID, EMP_NAME, DEPT_CODE, SALARY
FROM EMPLOYEE
WHERE SALARY > 3000000;
```

E...	EMP_NAME	DEPT_CODE	SALARY
206	박나라	D5	3700000
207	하이유	D5	2200000
208	김해술	D5	2500000
209	심봉선	D5	3500000
210	윤은해	D5	2000000
215	대북혼	D5	3760000
200	선동일	D9	8000000
201	송종기	D9	6000000
203	송은희	D6	3900000
204	유재식	D6	3400000
206	박나라	D5	3700000
209	심봉선	D5	3500000
215	대북혼	D5	3760000
217	전지연	D1	3660000

▶ 집합 연산자

✓ MINUS

선행 SELECT 결과에서 다음 SELECT 결과와 겹치는 부분을 제외한 나머지 부분 추출(차집합)

✓ 예시

```
SELECT EMP_ID, EMP_NAME, DEPT_CODE, SALARY
FROM EMPLOYEE
WHERE DEPT_CODE = 'D5'

MINUS

SELECT EMP_ID, EMP_NAME, DEPT_CODE, SALARY
FROM EMPLOYEE
WHERE SALARY > 3000000;
```

EMP_ID	EMP_NAME	DEPT_CODE	SALARY
207	하이유	D5	2200000
208	김해술	D5	2500000
210	윤은해	D5	2000000