GROUP BY

1 GROUP BY 문법

2 GROUP BY 사용이유

3 집계함수의 종류

강사 강태우

테스트 쿼리 생성

```
DROP TABLE 학생인적사항 ;
DROP TABLE 수강생정보 ;
DROP TABLE 성적표 ;

CREATE TABLE 수강생정보 (
학생ID VARCHAR2(9) PRIMARY KEY ,
학생이름 VARCHAR2(50) NOT NULL ,
소속반 VARCHAR2(5)
);

CREATE TABLE 성적표 (
    학생ID VARCHAR2(9) ,
    과목    VARCHAR2(30) ,
    성적    NUMBER  ,
    CONSTRAINT PK_성적표 PRIMARY KEY(학생ID , 과목) ,
    CONSTRAINT FK_성적표 FOREIGN KEY(학생ID) REFERENCES 수강생정보(학생ID)
) ;

INSERT INTO 수강생정보 VALUES ('S0001' , '김현철' , 'A') ;
INSERT INTO 수강생정보 VALUES ('S0002' , '문현중' , 'A') ;
INSERT INTO 수강생정보 VALUES ('S0003' , '강문치' , 'B') ;
INSERT INTO 수강생정보 VALUES ('S0004' , '박나선' , 'B') ;
INSERT INTO 수강생정보 VALUES ('S0005' , '신태강' , 'B') ;
INSERT INTO 수강생정보 VALUES ('S0006' , '물고기' , 'C') ;
INSERT INTO 수강생정보 VALUES ('S0007' , '자라니' , 'C') ;
INSERT INTO 수강생정보 VALUES ('S0008' , '공팔두' , 'C') ;
INSERT INTO 수강생정보 VALUES ('S0009' , '최팔현' , 'C') ;

INSERT INTO 성적표 VALUES('S0001'  ,'국어' , 90);
INSERT INTO 성적표 VALUES('S0001'  ,'수학' , 85);
INSERT INTO 성적표 VALUES('S0001'  ,'영어' , 100);
INSERT INTO 성적표 VALUES('S0002'  ,'국어' , 100);
INSERT INTO 성적표 VALUES('S0002'  ,'수학' , 100);
INSERT INTO 성적표 VALUES('S0002'  ,'영어' , 20);
INSERT INTO 성적표 VALUES('S0003'  ,'국어' , 100);
INSERT INTO 성적표 VALUES('S0003'  ,'수학' , 100);
INSERT INTO 성적표 VALUES('S0003'  ,'영어' , 20);
INSERT INTO 성적표 VALUES('S0004'  ,'국어' , 85);
INSERT INTO 성적표 VALUES('S0004'  ,'수학' , 40);
INSERT INTO 성적표 VALUES('S0004'  ,'영어' , 60);
INSERT INTO 성적표 VALUES('S0005'  ,'국어' , 100);
INSERT INTO 성적표 VALUES('S0005'  ,'수학' , 100);
INSERT INTO 성적표 VALUES('S0005'  ,'영어' , 100);
INSERT INTO 성적표 VALUES ( 'S0006' , '국어' , NULL ) ;
INSERT INTO 성적표 VALUES ( 'S0006' , '수학' , NULL ) ;
INSERT INTO 성적표 VALUES ( 'S0006' , '영어' , NULL ) ;

COMMIT;
```

테스트 테이블 내용

수강생정보

학생ID	학생이름	소속반
S0001	김현철	A
S0002	문현중	A
S0003	강문치	B
S0004	박나선	B
S0005	신태강	B
S0006	물고기	C
S0007	자라니	C
S0008	공팔두	C
S0009	최팔현	C

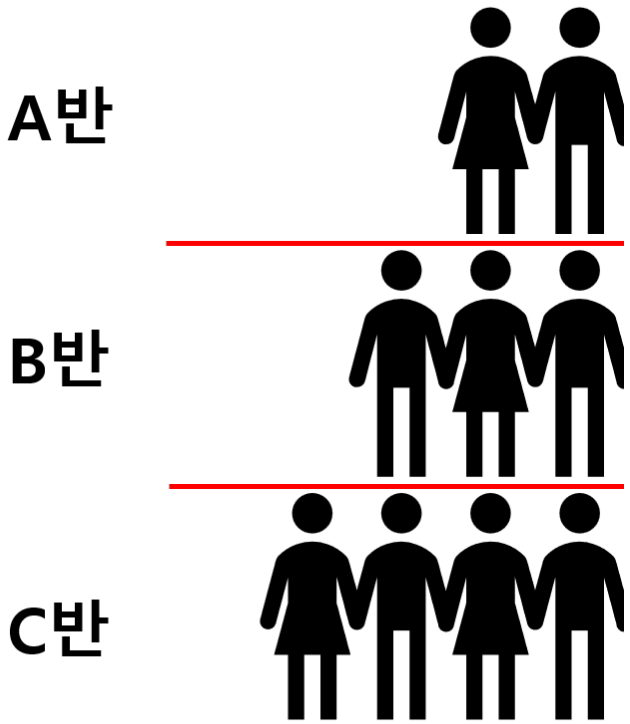
성적표

학생ID	과목	성적
S0001	국어	90
S0001	수학	85
S0001	영어	100
S0002	국어	100
S0002	수학	100
S0002	영어	20
S0003	국어	100
S0003	수학	100
S0003	영어	20
S0004	국어	85
S0004	수학	40
S0004	영어	60
S0005	국어	100
S0005	수학	100
S0005	영어	100
S0006	국어	(null)
S0006	수학	(null)
S0006	영어	(null)

1. GROUP BY 문법

반별로 몇 명이 있는지 집계해봅시다

학생ID	학생이름	소속반
S0001	김현철	A
S0002	문현중	A
S0003	강문치	B
S0004	박나선	B
S0005	신태강	B
S0006	물고기	C
S0007	자라니	C
S0008	공팔두	C
S0009	최팔현	C



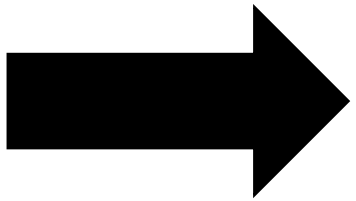
1. GROUP BY 문법

GROUP BY 문법을 이용하면 집계를 구할 수 있습니다.

```
SELECT 소속반 , COUNT(*) AS 반별인원수
  FROM 수강생정보
 GROUP BY 소속반;
```

소속반 컬럼을 기준으로 그룹화

학생ID	학생이름	소속반
S0001	김현철	A
S0002	문현중	A
S0003	강문치	B
S0004	박나선	B
S0005	신태강	B
S0006	물고기	C
S0007	자라니	C
S0008	공팔두	C
S0009	최팔현	C



소속반	반별인원수
A	2
B	3
C	4

1. GROUP BY 문법

GROUP BY는 특정 컬럼(표현식)을 기준으로

튜플(행)을 그룹화(=묶어서)하여 각각 단일행으로 표기합니다.

학생ID	학생이름	소속반
S0001	김현철	A
S0002	문현중	A
S0003	강문치	B
S0004	박나선	B
S0005	신태강	B
S0006	물고기	C
S0007	자라니	C
S0008	공팔두	C
S0009	최팔현	C

```
SELECT 소속반
  FROM 수강생정보
 GROUP BY 소속반;
```

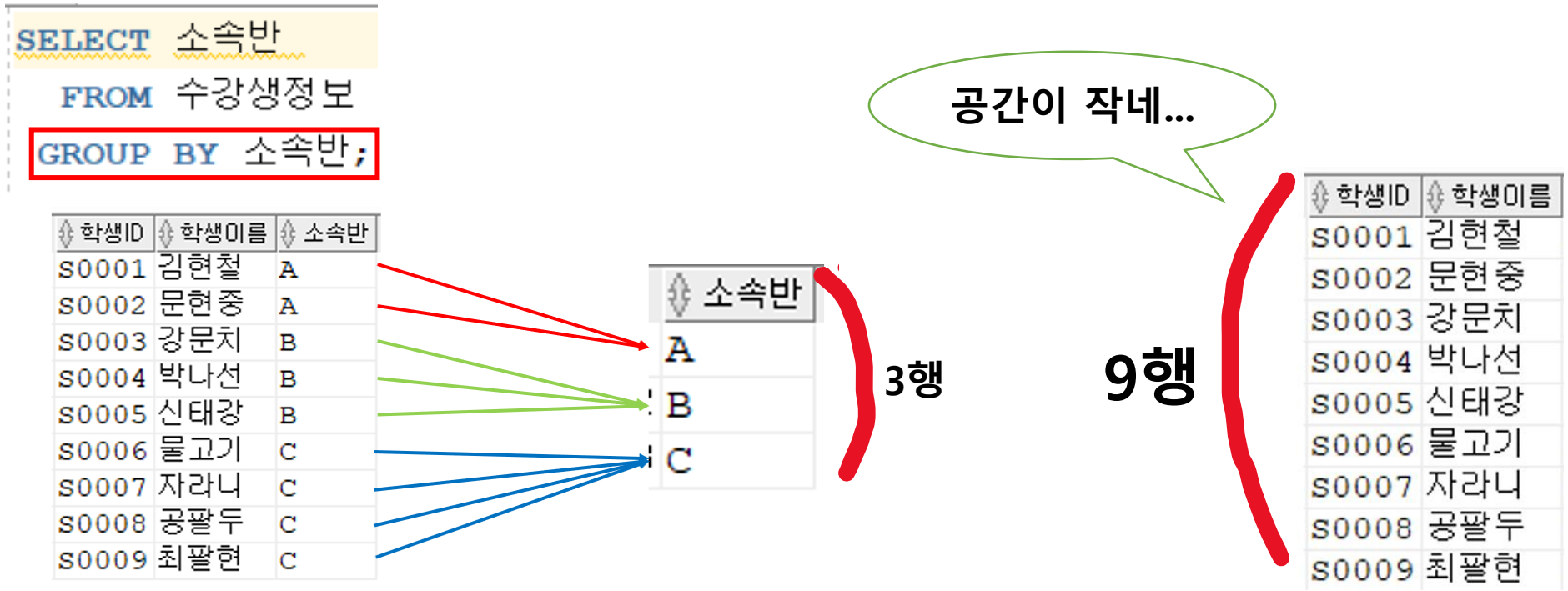
소속반 컬럼을 기준으로 그룹화

소속반
A
B
C

1. GROUP BY 문법

GROUP BY 는 실제로 출력되는 튜플(행)이 감소합니다.

따라서 입력할 수 있는 컬럼이 제한됩니다. (HAVING , ORDER BY , SELECT 에서 제한)



학생ID	학생이름	소속반
S0001	김현철	A
S0002	문현중	A
S0003	강문치	B
S0004	박나선	B
S0005	신태강	B
S0006	물고기	C
S0007	자라니	C
S0008	공팔두	C
S0009	최팔현	C

소속반
A
B
C

학생ID	학생이름
S0001	김현철
S0002	문현중
S0003	강문치
S0004	박나선
S0005	신태강
S0006	물고기
S0007	자라니
S0008	공팔두
S0009	최팔현

1. GROUP BY 문법

GROUP BY 는 실제로 출력되는 튜플(행)이 감소합니다.

따라서 입력할 수 있는 컬럼이 제한됩니다. (HAVING , ORDER BY , SELECT 에서 제한)

```
SELECT 소속반 , 학생이름
  FROM 수강생정보
 GROUP BY 소속반;
```

학생ID	학생이름	소속반
S0001	김현철	A
S0002	문현중	A
S0003	강문치	B
S0004	박나선	B
S0005	신태강	B
S0006	물고기	C
S0007	자라니	C
S0008	공팔두	C
S0009	최팔현	C

소속반	학생이름
A	김현철
B	문현중
C	강문치
	박나선
	신태강
	물고기
	자라니

ㅠㅠ 역시 안되네

ORA-00979: GROUP BY 표현식이 아닙니다.
00979, 00000 - "not a GROUP BY expression"
*Cause:
*Action:
2행, 1열에서 오류 발생

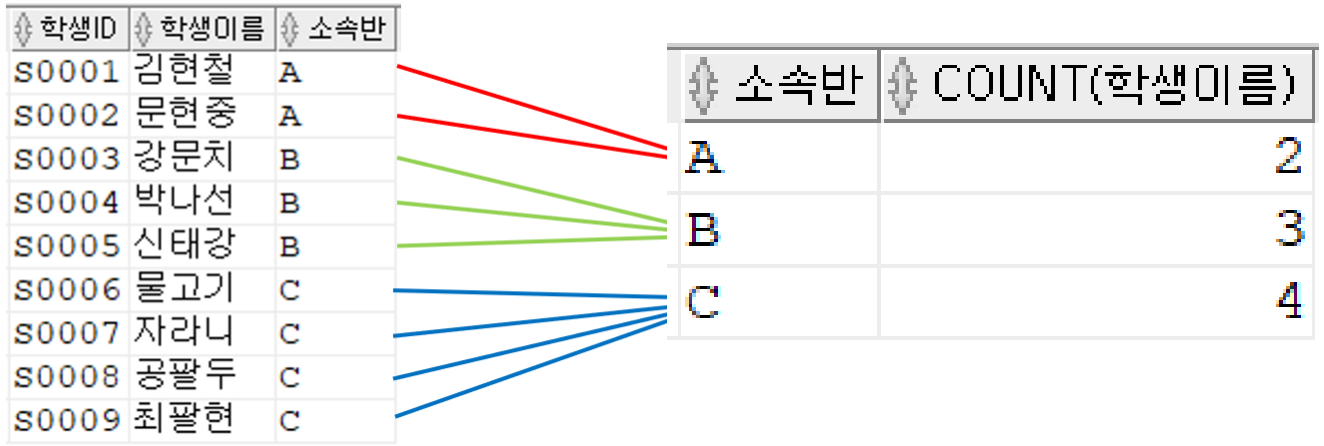
1. GROUP BY 문법

대신 집계함수로 처리한 컬럼은

HAVING , ORDER BY , SELECT 에도 입력이 가능합니다.

```
SELECT 소속반 , COUNT(학생이름)
  FROM 수강생정보
 GROUP BY 소속반 ;
```

<- COUNT() :
학생이름 컬럼에 집계함수인 COUNT() 를 적용하였습니다.
COUNT() 는 여러 행을 입력 받아 행 개수를 집계해 하나의 결과로 출력합니다.



학생ID	학생이름	소속반
S0001	김현철	A
S0002	문현중	A
S0003	강문치	B
S0004	박나선	B
S0005	신태강	B
S0006	물고기	C
S0007	자라니	C
S0008	공팔두	C
S0009	최팔현	C

소속반	COUNT(학생이름)
A	2
B	3
C	4

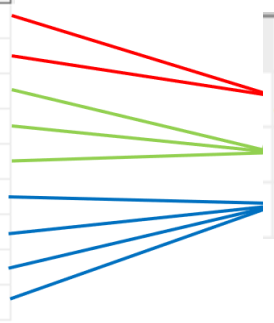
1. GROUP BY 문법

집계함수는 다중행 함수 라고도 불립니다.

여러 행을 입력 받아 행 개수를 집계해
하나의 결과로 출력하는 다중행 함수입니다.

```
SELECT 소속반 , COUNT(학생이름)
  FROM 수강생정보
 GROUP BY 소속반 ;
```

학생ID	학생이름	소속반
S0001	김현철	A
S0002	문현중	A
S0003	강문치	B
S0004	박나선	B
S0005	신태강	B
S0006	물고기	C
S0007	자라니	C
S0008	공팔두	C
S0009	최팔현	C



소속반	COUNT(학생이름)
A	2
B	3
C	4

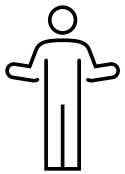
하나의 행을 입력 받아
하나의 결과를 출력하는 것은 단일행 함수입니다.

```
SELECT 직원ID , 연락처 , REPLACE(연락처 , '-' , '')
  FROM 직원연락처
 WHERE ROWNUM <= 5 ;
```

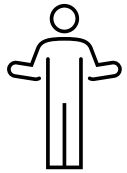
직원ID	연락처	REPLACE(연락처,'-','')
A0001	062-123-1234	0621231234
A0001	010-1231-1234	01012311234
A0002	062-254-6342	0622546342
A0002	010-2544-6342	01025446342
A0003	062-776-5231	0627765231

2. GROUP BY 사용 이유

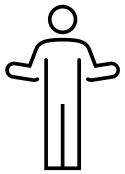
집계함수를 배워봅시다.



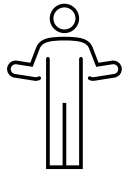
소속반 별로
몇 명의 학생이 있나요?



학생별로 과목(국어,수학,영어)
의 평균은 얼마인가요?



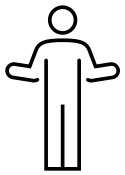
학생별로 과목(국어,수학,영어)
의 합계는 얼마인가요?



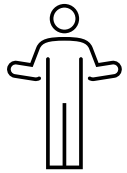
학생별로 과목(국어,수학,영어) 중
최고/최저 점수는 얼마인가요?

2. GROUP BY 사용 이유

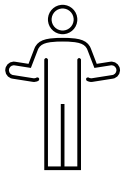
COUNT() , AVG() , SUM() , MAX() , MIN()



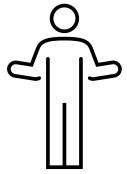
```
SELECT 소속반 , COUNT(*)
  FROM 수강생정보
 GROUP BY 소속반 ;
```



```
SELECT 학생ID , AVG(성적)
  FROM 성적표
 GROUP BY 학생ID ;
```



```
SELECT 학생ID , SUM(성적)
  FROM 성적표
 GROUP BY 학생ID ;
```



```
SELECT 학생ID , MAX(성적) , MIN(성적)
  FROM 성적표
 GROUP BY 학생ID ;
```

3. 집계함수의 종류

COUNT(expr)

- 그루핑할 컬럼 기준으로 행의 개수를 출력합니다.
- 다른 집계함수와 달리 * 을 expr에 쓸 수 있습니다.
- 모든 자료형에 이용가능 합니다.

```
SELECT 학생ID , COUNT(*)
  FROM 성적표
 GROUP BY 학생ID ;
```

학생ID	COUNT(*)
S0001	3
S0002	3
S0003	3
S0004	3
S0005	3
S0006	3

COUNT(*) 은 그룹별 NULL을 포함한 행의 개수를 출력합니다.

```
SELECT 학생ID , COUNT(성적)
  FROM 성적표
 GROUP BY 학생ID ;
```

학생ID	COUNT(성적)
S0001	3
S0002	3
S0003	3
S0004	3
S0005	3
S0006	0

COUNT(컬럼) 은 해당 컬럼에서 NULL을 제외하고 행의 개수를 출력합니다.

3. 집계함수의 종류

GROUP BY 가 없어도 집계함수를 사용할 수 있습니다

```
SELECT  COUNT(*)
   FROM 성적표 ;
```

이 경우에는 성적표 테이블에 있는 모든 데이터를 하나의 그룹으로 판단합니다.

단, SELECT에 어떤 컬럼도 사용할 수 없습니다.

다른 집계함수들도 모두 동일하게 사용가능 합니다.

3. 집계함수의 종류

MAX(expr) , MIN(expr)

- 그룹 기준으로 입력한 expr 에 대해 최대값이나 최소값을 출력합니다.
- NULL 데이터는 무시합니다. (모두 NULL 이면 NULL 출력)
- 모든 자료형에 이용가능 합니다.

```
SELECT 학생ID , MAX(성적)
  FROM 성적표
GROUP BY 학생ID ;
```

학생ID	MAX(성적)
S0001	100
S0002	100
S0003	100
S0004	85
S0005	100
S0006	(null)

MAX(expr) 은 그룹별 **최대값**을 출력합니다.
S0006의 성적은 데이터가 NULL밖에 없어
NULL이 출력되었습니다.

```
SELECT 학생ID , MIN(성적)
  FROM 성적표
GROUP BY 학생ID ;
```

학생ID	MIN(성적)
S0001	85
S0002	20
S0003	20
S0004	40
S0005	100
S0006	(null)

MIN(expr) 은 그룹별 **최소값**을 출력합니다.
S0006의 성적은 데이터가 NULL밖에 없어
NULL이 출력되었습니다.

3. 집계함수의 종류

AVG(expr)

- 그룹 기준으로 입력한 expr 에 대해 평균값을 출력합니다.
- NULL 데이터는 무시합니다. (모두 NULL 이면 NULL 출력)
- 숫자형에만 이용가능 합니다.

```
SELECT 학생ID , AVG(성적)
  FROM 성적표
 GROUP BY 학생ID ;
```

학생ID	AVG(성적)
S0001	91.6666666666666666666666666666666666666667
S0002	73.3333333333333333333333333333333333333333
S0003	73.3333333333333333333333333333333333333333
S0004	61.6666666666666666666666666666666666666667
S0005	100
S0006	(null)

각 학생별로 성적(국어,수학,영어) 의 평균값을 출력했습니다.

S0006의 성적은 데이터가 NULL밖에 없어 NULL이 출력되었습니다.

3. 집계함수의 종류

SUM(expr)

- 그룹 기준으로 입력한 expr 에 대해 합계값을 출력합니다.
- NULL 데이터는 무시합니다. (모두 NULL 이면 NULL 출력)
- 숫자형에만 이용가능 합니다.

```
SELECT 학생ID , SUM(성적)
  FROM 성적표
 GROUP BY 학생ID ;
```

학생ID	SUM(성적)
S0001	275
S0002	220
S0003	220
S0004	185
S0005	300
S0006	(null)

각 학생별로 성적(국어,수학,영어) 의 합계값을 출력했습니다.

S0006의 성적은 데이터가 NULL밖에 없어 NULL이 출력되었습니다.

실습문제를 풀어봅시다.(1/2)

1. 성적표 테이블에서 학생별로 취득한 성적의 합계를 출력해주세요.
   (힌트 : SUM)

학생ID	학생별성적합계
S0001	275
S0005	300
S0002	220
S0004	185
S0003	220
S0006	(null)

2. TB_PRD 테이블에서 PRD_TYPE (상품종류) 별로 몇 개의 상품이 있는지 오른쪽 결과와 같이 출력해주세요. [힌트 : COUNT , MAX] (출력 순서는 상관X)

PRD_TYPE	상품타입별개수	상품별최고가
컴퓨터	3	2000000
주방용품	4	80000
스마트폰	2	1500000
가전	4	1500000
욕실용품	6	50000

3. 성적표 테이블에서 학생별로 국어와 영어 성적의 평균을 출력해주세요.
   (힌트 : WHERE 절로 과목이 수학이 아닌 데이터만 출력하기)

학생ID	수학제외한평균
S0001	95
S0002	60
S0003	60
S0004	72.5
S0005	100
S0006	(null)

실습문제를 풀어봅시다.(2/2)

4 (심화) TB_MEMBER 테이블과 TB_MEMBER_TEL 테이블을 이용해서 회원별로 연락처가 몇개 있는지 알고 싶습니다. TB_MEMBER 테이블을 기준으로 모든 회원을 보여주되, 연락처가 없는 대상도 0건으로 출력되도록 해주세요. (출력되는 순서는 상관없습니다)

예) 회원 AAAAA 는 집,휴대폰,회사 총 3개의 연락처를 가지고 있습니다.
회원 BBBBB 는 집,휴대폰 총 2개의 연락처를 가지고 있습니다.
그 외 회원들은 연락처 정보가 없어 모두 0건의 연락처를 가지고 있습니다.

힌트1 : SELECT A.MEMBER_ID, COUNT(B.TEL_NO) AS 연락처개수 ....
힌트2 : ANSI 문법으로 아우터 조인을 활용하세요. 기준은 TB_MEMBER 테이블입니다.

MEMBER_ID	연락처개수
CCCCC	0
IIIII	0
FFFFF	0
EEEEE	0
DDDDD	0
HHHHH	0
AAAAA	3
BBBBB	2
GGGGG	0

답(1/2)

1. 성적표 테이블에서 학생별로 취득한 성적의 합계를 출력해주세요.(힌트 : SUM)

답)
```
SELECT 학생ID , SUM(성적) AS 학생별성적합계
  FROM 성적표
 GROUP BY 학생ID ;
```

학생ID	학생별성적합계
S0001	275
S0005	300
S0002	220
S0004	185
S0003	220
S0006	(null)

2. TB_PRD 테이블에서 PRD_TYPE (상품종류) 별로 몇 개의 상품이 있는지 오른쪽 결과와 같이 출력해주세요. [힌트 : COUNT , MAX] (출력 순서는 상관X)

답)
```
SELECT PRD_TYPE
     , COUNT(*) AS 상품타입별개수
     , MAX(PRD_PRICE) AS 상품별최고가
  FROM TB_PRD
 GROUP BY PRD_TYPE ;
```

PRD_TYPE	상품타입별개수	상품별최고가
컴퓨터	3	2000000
주방용품	4	80000
스마트폰	2	1500000
가전	4	1500000
욕실용품	6	50000

3. 성적표 테이블에서 학생별로 국어와 영어 성적의 평균을 출력해주세요. (힌트 : WHERE 절로 과목이 수학이 아닌 데이터만 출력하기)

답)
```
SELECT 학생ID , AVG(성적) AS 수학제외한평균
  FROM 성적표
 WHERE 과목 != '수학'
 GROUP BY 학생ID ;
```

학생ID	수학제외한평균
S0001	95
S0002	60
S0003	60
S0004	72.5
S0005	100
S0006	(null)

답(2/2)

4 (심화) TB_MEMBER 테이블과 TB_MEMBER_TEL 테이블을 이용해서 회원별로 연락처가 몇개 있는지 알고 싶습니다. TB_MEMBER 테이블을 기준으로 모든 회원을 보여주되, 연락처가 없는 대상도 0건으로 출력되도록 해주세요. (출력되는 순서는 상관없습니다)

예) 회원 AAAAA 는 집,휴대폰,회사 총 3개의 연락처를 가지고 있습니다.
회원 BBBBB 는 집,휴대폰 총 2개의 연락처를 가지고 있습니다.
그 외 회원들은 연락처 정보가 없어 모두 0건의 연락처를 가지고 있습니다.

힌트1 : SELECT A.MEMBER_ID, COUNT(B.TEL_NO) AS 연락처개수 ....
힌트2 : ANSI 문법으로 아우터 조인을 활용하세요. 기준은 TB_MEMBER 테이블입니다.

답)

```
SELECT A.MEMBER_ID, COUNT(B.TEL_NO) AS 연락처개수
  FROM TB_MEMBER A LEFT OUTER JOIN TB_MEMBER_TEL B
    ON ( A.MEMBER_ID = B.MEMBER_ID)
 GROUP BY A.MEMBER_ID ;
```

MEMBER_ID	연락처개수
CCCCC	0
IIIII	0
FFFFF	0
EEEEE	0
DDDDD	0
HHHHH	0
AAAAA	3
BBBBB	2
GGGGG	0

HAVING

1 HAVING 문법

2 HAVING 사용이유

3 HAVING 주의사항

강사 강태우

1. HAVING 문법

```
SELECT 학생ID , ROUND( AVG(성적) , 1) AS 평균성적
  FROM 성적표
 GROUP BY 학생ID ;
```

학생ID	평균성적
S0001	91.7
S0002	73.3
S0003	73.3
S0004	61.7
S0005	100
S0006	(null)

평균성적이 75점 이하인 학생은 몇 명인가요?

(NULL은 제외)

1. HAVING 문법

HAVING 문법은

집계가 완료된 대상을 필터링하는 문법입니다.

학생ID	평균성적
S0001	91.7
S0002	73.3
S0003	73.3
S0004	61.7
S0005	100
S0006	(null)

```
SELECT 학생ID , ROUND( AVG(성적) , 1) AS 평균성적
  FROM 성적표
 GROUP BY 학생ID
 HAVING AVG(성적) <= 75 ;
```

학생ID	평균성적
S0002	73.3
S0003	73.3
S0004	61.7

2. HAVING 사용이유

HAVING은 집계함수에 대해 조건을 줄 수 있습니다.

WHERE절에서는 집계함수를 사용할 수 없습니다.

```
SELECT 학생ID , ROUND( AVG(성적) , 1) AS 평균성적
  FROM 성적표
 GROUP BY 학생ID
 HAVING AVG(성적) <= 75 ;
```

학생ID	평균성적
S0002	73.3
S0003	73.3
S0004	61.7

```
SELECT 학생ID , ROUND( AVG(성적) , 1) AS 평균성적
  FROM 성적표
 WHERE AVG(성적) <= 75
 GROUP BY 학생ID ;
```

```
ORA-00934: 그룹 함수는 허가되지 않습니다
00934. 00000 -  "group function is not allowed here"
```

2. HAVING 사용이유

WHERE절에서는 집계함수를 사용하지 못하는 이유

- **WHERE 절은 GROUP BY 보다 먼저 실행**됩니다.

```
SELECT 학생ID , AVG(성적) AS 수학제외한평균
  FROM 성적표
 WHERE 과목 != '수학'
 GROUP BY 학생ID
 HAVING AVG(성적) <= 75;
```

학생ID	과목	성적
S0001	국어	90
S0001	수학	85
S0001	영어	100
S0002	국어	100
S0002	수학	100
S0002	영어	20
S0003	국어	100
S0003	수학	100
S0003	영어	20
S0004	국어	85
S0004	수학	40
S0004	영어	60
S0005	국어	100
S0005	수학	100
S0005	영어	100
S0006	국어	(null)
S0006	수학	(null)
S0006	영어	(null)

WHERE

과목 != '수학'

학생ID	과목	성적
S0001	국어	90
S0001	영어	100
S0002	국어	100
S0002	영어	20
S0003	국어	100
S0003	영어	20
S0004	국어	85
S0004	영어	60
S0005	국어	100
S0005	영어	100
S0006	국어	(null)
S0006	영어	(null)

GROUP BY

학생ID

GROUP BY 완료

이제 집계함수 사용이 가능합니다!

학생ID	수학제외한평균
S0001	95
S0002	60
S0003	60
S0004	72.5
S0005	100
S0006	(null)

HAVING

AVG(성적) <= 75

학생ID	수학제외한평균
S0002	60
S0003	60
S0004	72.5

3. HAVING 주의사항

WHERE -> GROUP BY -> HAVING 순서이므로 HAVING 은 GROUP BY 의 영향을 받게 됩니다.
따라서 GROUP BY 에 입력된 컬럼에 의해서 입력 가능한 컬럼의 제약을 받게 됩니다.

```
SELECT 부서ID , SUM(연봉)
  FROM 직원
 GROUP BY 부서ID
HAVING 부서ID IN ('D001' , 'D002') ;
```

```
SELECT 부서ID , SUM(연봉)
  FROM 직원
 GROUP BY 부서ID
HAVING SUM(연봉) >= 13000
```

```
SELECT 부서ID , SUM(연봉)
  FROM 직원
 GROUP BY 부서ID
HAVING 연봉 >= 6000 ;
```

ORA-00979: GROUP BY 표현식이 아닙니다.
00979, 00000 - "not a GROUP BY expression"
*Cause:
*Action:
77행, 68열에서 오류 발생

```
SELECT 부서ID , SUM(연봉) AS 연봉합계
  FROM 직원
 GROUP BY 부서ID
HAVING 연봉합계 >= 6000 ;
```

ORA-00904: "연봉합계": 부적합한 식별자
00904, 00000 - "%s: invalid identifier"
*Cause:
*Action:
77행, 76열에서 오류 발생

실습문제를 풀어봅시다.

1. 수강생정보 테이블에서 소속반 별로 인원수가 3명이상인 튜플(행)만 출력해주세요.

소속반	반별인원수
B	3
C	4

2. TB_PRD 테이블에서 PRD_TYPE(상품타입) 별로 최고가를 구한 후,
그 최고가가 100만원을 넘는 대상만을 추출해주세요.

PRD_TYPE	상품별최고가
컴퓨터	2000000
스마트폰	1500000
가전	1500000

3. TB_PRD 테이블에서 PRD_TYPE(상품타입) 별로 상품개수를 구한 후,
해당 상품개수가 4개인 대상만을 추출해주세요.

PRD_TYPE	상품별개수
주방용품	4
가전	4

답안

1. 수강생정보 테이블에서 소속반 별로 인원수가 3명이상인 튜플(행)만 출력해주세요.

답)
```
SELECT 소속반 , COUNT(*) AS 반별인원수
  FROM 수강생정보
 GROUP BY 소속반
 HAVING COUNT(*) >= 3 ;
```

소속반	반별인원수
B	3
C	4

2. TB_PRD 테이블에서 PRD_TYPE(상품타입) 별로 최고가를 구한 후,
그 최고가가 100만원을 넘는 대상만을 추출해주세요.

답)
```
SELECT PRD_TYPE
     , MAX(PRD_PRICE) AS 상품별최고가
  FROM TB_PRD
 GROUP BY PRD_TYPE
 HAVING MAX(PRD_PRICE) > 1000000;
```

PRD_TYPE	상품별최고가
컴퓨터	2000000
스마트폰	1500000
가전	1500000

3. TB_PRD 테이블에서 PRD_TYPE(상품타입) 별로 상품개수를 구한 후,
해당 상품개수가 4개인 대상만을 추출해주세요.

답)
```
SELECT PRD_TYPE , COUNT(*) AS 상품별개수
  FROM TB_PRD
 GROUP BY PRD_TYPE
 HAVING COUNT(*) = 4 ;
```

PRD_TYPE	상품별개수
주방용품	4
가전	4

ORDER BY

1 ORDER BY 문법

2 ORDER BY 원리

3 ORDER BY 사용방식

강사 강태우

1. ORDER BY 문법

ORDER BY는 특정 컬럼을 기준으로

데이터를 오름차순/내림차순 정렬합니다.

예) TB_PRD 테이블을 PRD_PRICE(상품가격) 순으로 오름차순으로 정렬하여 아래와 같이 출력해주세요.

```
SELECT  PRD_ID
      , PRD_NAME
      , PRD_TYPE
      , PRD_PRICE
  FROM TB_PRD
ORDER BY PRD_PRICE ;
```

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0015	수세미	욕실용품	5000
P0020	수건	욕실용품	5000
P0017	곰팡이제거제	욕실용품	10000
P0014	칼	주방용품	15000
P0010	조아샴푸	욕실용품	20000
P0019	린스	욕실용품	20000
P0011	주전자	주방용품	20000
P0013	냄비	주방용품	30000
P0001	헤어드라이기	가전	30000
P0018	샤워기	욕실용품	50000
P0012	전기밥솥	주방용품	80000
P0003	세탁기	가전	600000
P0007	태블릿	컴퓨터	800000
P0004	건조기	가전	800000
P0008	애플14	스마트폰	1200000
P0005	노트북	컴퓨터	1500000
P0002	에어컨	가전	1500000
P0009	갤럭시s23	스마트폰	1500000
P0006	데스크탑	컴퓨터	2000000

1. ORDER BY 문법

ORDER BY는 특정 컬럼을 기준으로

데이터를 오름차순/내림차순 정렬합니다.

예) TB_PRD 테이블을 PRD_PRICE(상품가격) 순으로 내림차순으로 정렬하여 아래와 같이 출력해주세요.

```
SELECT PRD_ID
     , PRD_NAME
     , PRD_TYPE
     , PRD_PRICE
  FROM TB_PRD
 ORDER BY PRD_PRICE DESC  ;
```

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0006	데스크탑	컴퓨터	2000000
P0009	갤럭시S23	스마트폰	1500000
P0002	에어컨	가전	1500000
P0005	노트북	컴퓨터	1500000
P0008	애플14	스마트폰	1200000
P0004	건조기	가전	800000
P0007	태블릿	컴퓨터	800000
P0003	세탁기	가전	600000
P0012	전기밥솥	주방용품	80000
P0018	샤워기	욕실용품	50000
P0001	헤어드라이기	가전	30000
P0013	냄비	주방용품	30000
P0011	주전자	주방용품	20000
P0010	조아샴푸	욕실용품	20000
P0019	린스	욕실용품	20000
P0014	칼	주방용품	15000
P0017	곰팡이제거제	욕실용품	10000
P0020	수건	욕실용품	5000
P0015	수세미	욕실용품	5000

1. ORDER BY 문법

ORDER BY는 특정 컬럼을 기준으로

데이터를 오름차순/내림차순 정렬합니다. (여러 컬럼 사용 가능)

예) TB_PRD 테이블을 PRD_TYPE(상품타입) 별로 오름차순 정렬하되 ,
같은 타입일 경우 PRD_PRICE(상품가격) 순으로 내림차순으로 정렬하여 아래와 같이 출력해주세요.

```
SELECT PRD_ID
     , PRD_NAME
     , PRD_TYPE
     , PRD_PRICE
  FROM TB_PRD
 ORDER BY PRD_TYPE , PRD_PRICE DESC  ;
```

PRD_ID	PRD_NAME	PRD_TYPE	PRD_PRICE
P0002	에어컨	가전	1500000
P0004	건조기	가전	800000
P0003	세탁기	가전	600000
P0001	헤어드라이기	가전	30000
P0009	갤럭시S23	스마트폰	1500000
P0008	애플14	스마트폰	1200000
P0018	샤워기	욕실용품	50000
P0010	조아샴푸	욕실용품	20000
P0019	린스	욕실용품	20000
P0017	곰팡이제거제	욕실용품	10000
P0020	수건	욕실용품	5000
P0015	수세미	욕실용품	5000
P0012	전기밥솥	주방용품	80000
P0013	냄비	주방용품	30000
P0011	주전자	주방용품	20000
P0014	칼	주방용품	15000
P0006	데스크탑	컴퓨터	2000000
P0005	노트북	컴퓨터	1500000
P0007	태블릿	컴퓨터	800000

2. ORDER BY 원리

주의) GROUP BY 가 사용되었을 경우 입력 가능한 컬럼에 제약이 발생합니다.

그 이유는 ORDER BY가 GROUP BY -> HAVING -> SELECT -> ORDER BY 순서로 실행되기 때문입니다.

```
SELECT PRD_TYPE
     , COUNT(*) AS 상품타입별개수
  FROM TB_PRD
 GROUP BY PRD_TYPE
 ORDER BY PRD_TYPE ;
```

O

```
SELECT PRD_TYPE
     , COUNT(*) AS 상품타입별개수
  FROM TB_PRD
 GROUP BY PRD_TYPE
 ORDER BY PRD_PRICE ;
```

X

```
ORA-00979: not a GROUP BY expression
00979, 00000 - "not a GROUP BY expression"
```

2. ORDER BY 사용 방식

컬럼이름 외에 AS 명칭이나 숫자로도 표현이 가능합니다.

```
SELECT PRD_ID
     , PRD_NAME
     , PRD_TYPE
     , PRD_PRICE
  FROM TB_PRD
ORDER BY 4 ;
```

```
SELECT PRD_ID
     , PRD_NAME
     , PRD_TYPE
     , PRD_PRICE
  FROM TB_PRD
ORDER BY PRD_PRICE ;
```

```
SELECT PRD_ID
     , PRD_NAME
     , PRD_TYPE
     , PRD_PRICE AS 상품가격
  FROM TB_PRD
ORDER BY 상품가격 ;
```

세 쿼리는 모두 동일한 정렬 순서를 의미합니다.

GROUP BY
HAVING
ORDER BY
END