Database

(데이터베이스)

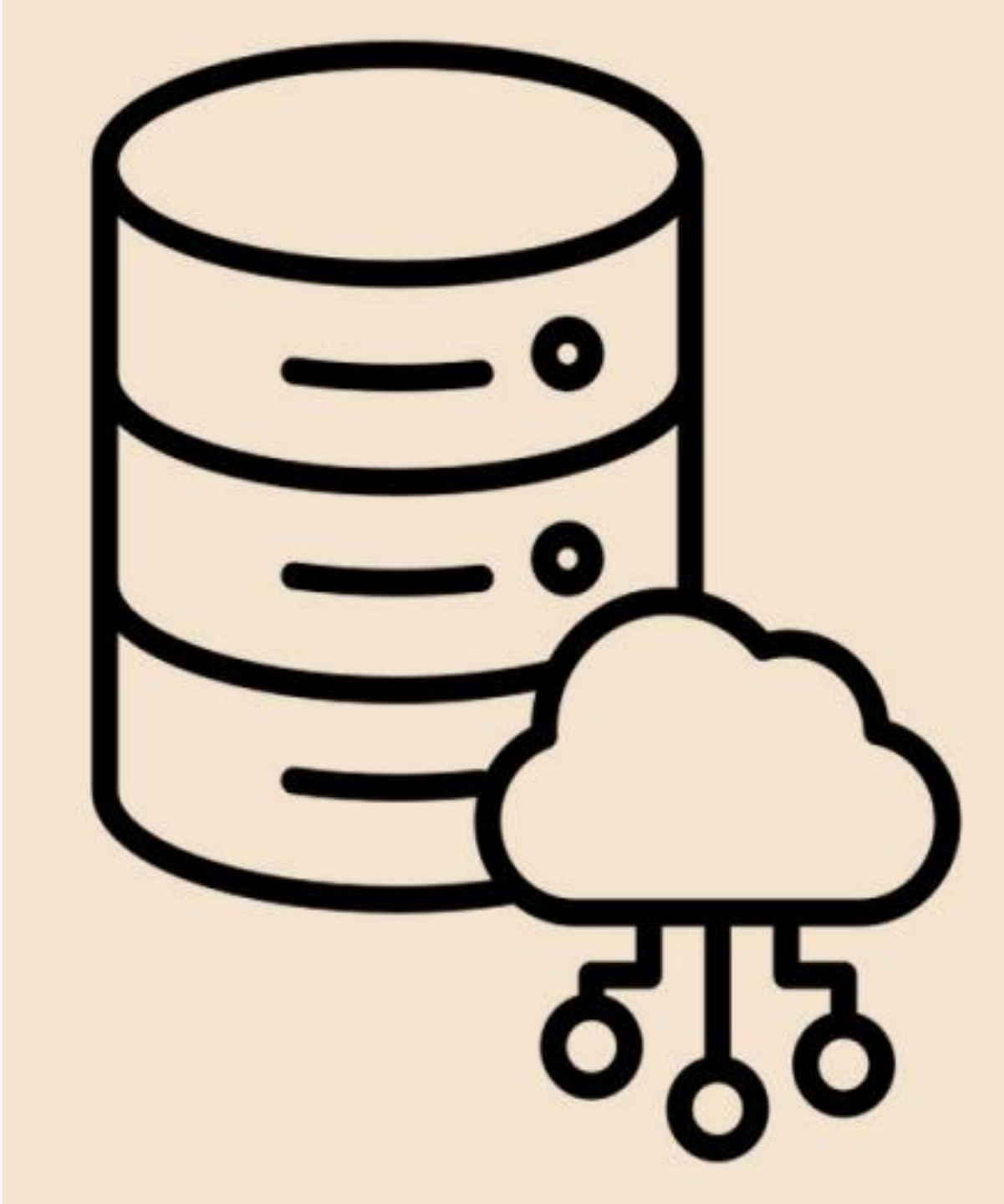
김민수

GROUP BY

1 GROUP BY를 쓰는 이유

2 GROUP BY 문법

3 집계함수의 종류



GROUP BY를 쓰는 이유

SQL 문법 실행 순서

5	SELECT	출력하고 싶은 컬럼만 작성하기
1	FROM	데이터를 가져올 테이블 입력
2	WHERE	원하는 튜플만 가져오도록 필터링(조건문)
3	GROUP BY	특정 컬럼을 기준으로 그룹화
4	HAVING	그룹화 상태의 데이터를 필터링
6	ORDER BY	특정 컬럼으로 정렬하기

GROUP BY를 쓰는 이유

GROUP BY : 특정 컬럼을 기준으로 그룹화

사람 TABLE

이름	분류	연봉
민수	선생님	?
톰 크루즈	외국 연예인	?
경남	선생님	?
장범준	한국 연예인	?
헤이즈	한국 연예인	?
아이유	한국 연예인	?
수민	선생님	?
톰 홀랜드	외국 연예인	?
안유진	한국 연예인	?
잭블랙	외국 연예인	?
로다주	외국 연예인	?
지영	선생님	?

한국 연예인들의
평균 연봉은 얼마일까?

"분류" 컬럼을 기준으로
그룹화 해보자!!
→ 계산하기 편리하겠네

분류	평균 연봉
한국 연예인	?원
외국 연예인	?원
선생님	?원

GROUP BY를 쓰는 이유

집계를 편하게 하기 위해 GROUP BY 문법을 이용

GROUP BY는 특정 조건에 어울리는 튜플끼리 묶어서 그룹화

예) 한국 연예인, 외국 연예인, 선생님의(직업별) 평균 연봉을 구해주세요

예) 인사팀, 기획팀, 홍보팀의(팀별) 각 팀 인원이 몇 명인지 구해주세요

GROUP BY 문법

예) 반별로 몇 명이 있는지 집계해 주세요

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

Q1. 무엇을 기준으로 그룹화 할까?

Q2. 필요한 컬럼은 무엇일까?

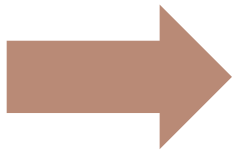
Q3. 결과는 어떻게 출력되어야 할까?

GROUP BY 문법

GROUP BY 문법을 사용하여 튜플 그룹화

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

소속반 컬럼을 기준으로 그룹화



소속반	COUNT(소속반)
A	2
B	3
C	4

아하! GROUP BY를 사용하면 공통 튜플로 묶여버리는구나!

GROUP BY 문법

GROUP BY 로 그룹화 한 결과는 출력되는 튜플(행)이 감소

따라서 그룹화되지 않은 다른 컬럼과 나란히 사용 불가능!!!

(HAVING , ORDER BY , SELECT 에서 제한)

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

	소속반
1	A
2	B
3	C

+

학생이름
김현철
문현중
강문치
박나선
신태강
물고기
자라니
공팔두
최팔현

=

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

GROUP BY 문법

대신 집계함수로 처리한 컬럼은 HAVING , ORDER BY , SELECT 에도 입력 가능

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

<- COUNT() : ()안에 있는 컬럼의 개수를 세어주는 함수

COUNT() 는 여러 행의 개수를 세어서 한 개의 결과를 도출

여기서는 소속 반 각각의 수를 세었기 때문에 3개의 결과 출력!

소속반	COUNT(학생이름)
A	2
B	3
C	4

집계함수의 종류

GROUP BY에 필요한 집계함수

소속반 별로
몇 명의 학생이 있나요?

학생들의 과목별
평균은 몇 점인가요?

학생들의 과목별
합계 점수는 몇 점인가요?

학생들의 과목별 최고/최저
점수는 몇 점인가요?

집계함수의 종류

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;
```

집계함수의 종류

COUNT(col)

- 그룹화한 컬럼 기준으로 행의 개수를 출력
- 다른 집계함수와 달리 col 자리에 * 사용 가능
- 모든 자료형에 이용가능

```
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 출력

```
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 제외한 행의 개수를 출력

집계함수의 종류

MAX(col) , MIN(col)

- 그룹화된 컬럼을 기준으로 col의 최대 및 최소값을 출력
- NULL 데이터는 무시 (대신, 모두 NULL 이면 NULL 출력)
- 모든 자료형에 이용가능

```
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(col) 은 그룹별 최대값을 출력

S0006의 성적은 데이터가 NULL밖에 없으므로 NULL 출력

```
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(col) 은 그룹별 최소값을 출력

S0006의 성적은 데이터가 NULL밖에 없으므로 NULL 출력

집계함수의 종류

AVG(col)

- 그룹 기준으로 입력한 col 에 대해 평균값을 출력
- NULL 데이터는 무시 (대신, 모두 NULL 이면 NULL 출력)
- 숫자형에만 이용가능

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

학생ID	AVG(성적)
S0001	91.666666666666666666666666666666666666667
S0002	73.333333333333333333333333333333333333333
S0003	73.333333333333333333333333333333333333333
S0004	61.666666666666666666666666666666666666667
S0005	100
S0006	(null)

각 학생별로 성적(국어,수학,영어) 의 평균값 출력

S0006의 성적은 데이터가 NULL밖에 없으므로 NULL이 출력

집계함수의 종류

SUM(col)

- 그룹 기준으로 입력한 col 에 대해 합계값을 출력
- NULL 데이터는 무시 (대신, 모두 NULL 이면 NULL 출력)
- 숫자형에만 이용가능

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

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

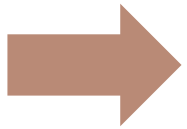
각 학생별로 성적(국어,수학,영어) 의 합계 값 출력

S0006의 성적은 데이터가 NULL밖에 없으므로 NULL이 출력

집계함수의 종류

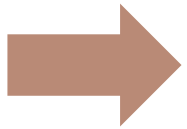
GROUP BY 가 없어도 집계함수 사용 가능

```
SELECT COUNT(*)
FROM 성적표 ;
```



COUNT(*)
18

```
SELECT MAX(성적)
FROM 성적표 ;
```



MAX(성적)
100

이 경우에는 성적표 테이블에 있는 모든 튜플을 하나의 그룹으로 판단

대신 이 때, SELECT에는 집계되지 않은 컬럼은 집계된 함수와 사용 불가능

집계함수의 종류

실습 문제

1. 성적표 테이블에서 학생별로 평균점수를 출력해주세요.
   이때 소수점 1자리 까지만 출력되도록 ROUND 함수도 활용해보세요.
   [힌트 : ROUND (평균을뽑은값 , 1)]

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

2. 직원 테이블에서 모든 직원 중에 최고연봉과 최저연봉을 출력해주세요.

최고연봉	최저연봉
9300	2600

3. 수강생정보 테이블에서 각 소속된 반별로 몇 명이 있는지 출력해주세요.

소속반	반별인원수
A	2
B	3
C	4

집계함수의 종류

실습 문제

4. 성적표 테이블에서 학생별로 국어와 영어 성적의 평균을 출력해주세요.
(힌트 : 과목이 수학인 데이터는 제외하기)

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

5. 직원 테이블에서 부서별로 연봉의 합계를 출력해주세요.

부서ID	부서별연봉합계
D001	10300
D002	9200
D003	10100
D004	16500
D005	14300

집계함수의 종류

실습 문제

6. 직원 테이블과 직원 연락처 테이블을 이용해서 직원별로 연락처정보가 몇개 있는지 출력해주세요.
직원 테이블을 기준으로 A0001 ~ A0011 의 모든 직원을 보여주되 , 연락처가 없는 대상도 0건으로 출력되도록 해주세요. (단 , 조인시 오라클방식의 조인을 이용해보세요)

예) 직원 A0001 은 집전화, 휴대폰 둘다 가지고 있으므로 2개의 연락처 정보가 있습니다.
직원 A0006 은 휴대폰 만 있으므로 1개의 연락처 정보가 있습니다.
직원 A0009 은 연락처 정보가 없어서 0건을 표시하고 싶습니다.

힌트1 : SELECT A.직원ID , COUNT(B.연락처) AS 연락처개수 …
힌트2 : 아우터조인 활용

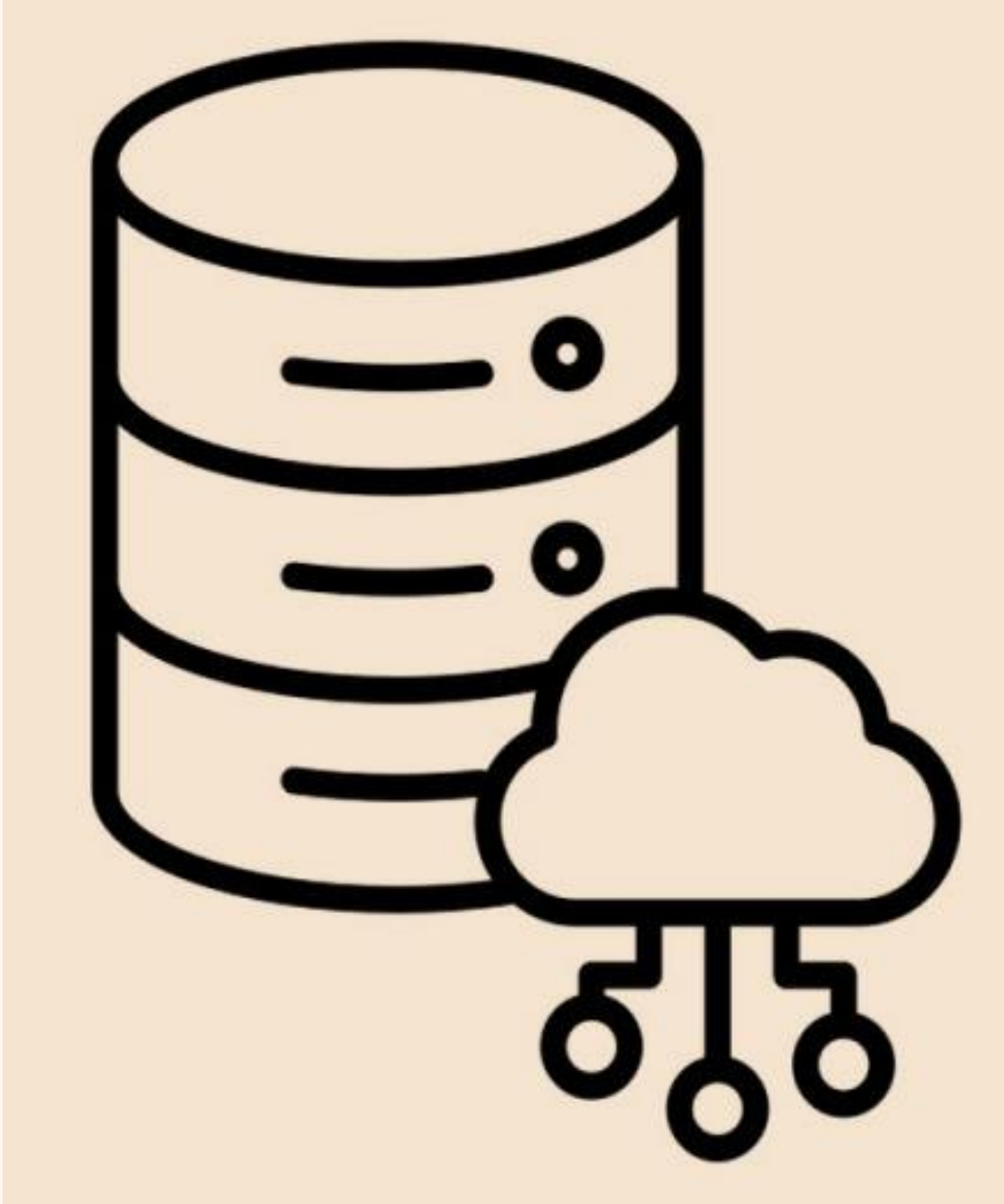
직원ID	연락처개수
A0001	2
A0002	2
A0003	2
A0004	2
A0005	2
A0006	1
A0007	1
A0008	1
A0011	0
A0009	0
A0010	0

HAVING

1 HAVING을 쓰는 이유

2 HAVING 문법

3 HAVING 사용 시 주의사항



HAVING을 쓰는 이유

SQL 문법 실행 순서

5	SELECT	출력하고 싶은 컬럼만 작성하기
1	FROM	데이터를 가져올 테이블 입력
2	WHERE	원하는 튜플만 가져오도록 필터링(조건문)
3	GROUP BY	특정 컬럼을 기준으로 그룹화
4	HAVING	그룹화 상태의 데이터를 필터링
6	ORDER BY	특정 컬럼으로 정렬하기

HAVING을 쓰는 이유

HAVING : 그룹화 상태의 데이터를 필터링(조건)

사람 TABLE

이름	분류	연봉
민수	선생님	?
톰 크루즈	외국 연예인	?
경남	선생님	?
장범준	한국 연예인	?
헤이즈	한국 연예인	?
아이유	한국 연예인	?
수민	선생님	?
톰 홀랜드	외국 연예인	?
안유진	한국 연예인	?
잭블랙	외국 연예인	?
로다주	외국 연예인	?
지영	선생님	?

이번엔 세 직업 중 평균 연봉이 3000만원 이상인 직업이 궁금해요!

평균 연봉 컬럼에 조건을 추가해보자!

분류	평균 연봉
한국 연예인	?원
외국 연예인	?원
선생님	?원

HAVING을 쓰는 이유

Question? 선생님! 저는 WHERE문 배웠으니까 그걸 사용할게요!

→ **불가능!**

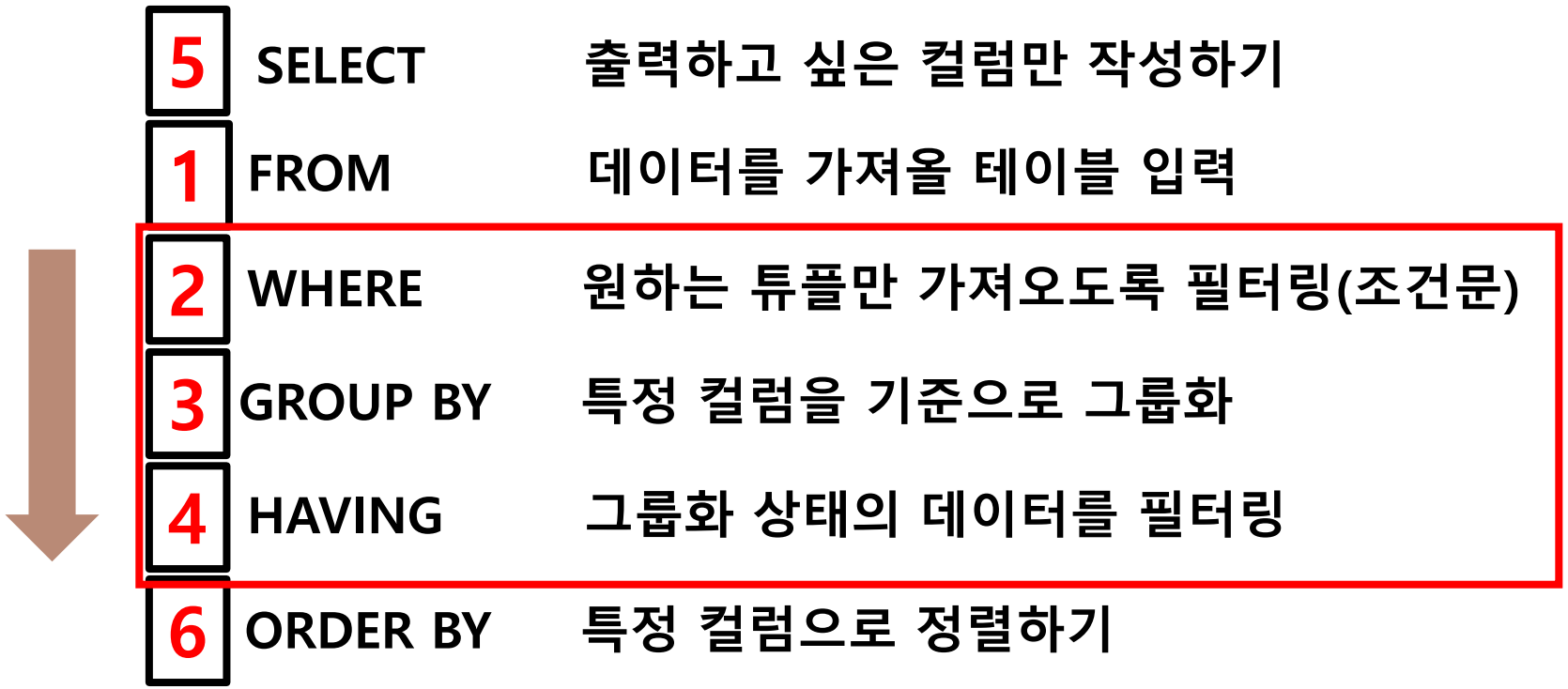
이유 1. HAVING 내부에서 집계함수 사용 가능 (WHERE문에선 X)

```
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

HAVING을 쓰는 이유

이유 2. WHERE문은 GROUP BY 보다 먼저 실행되기 때문!



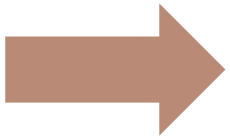
HAVING문법

예) 평균성적이 75점 이하인 학생 ID와 평균 성적을 계산해보자!

(NULL은 제외)

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

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



학생ID	평균성적
S0002	73.3
S0003	73.3
S0004	61.7

HAVING 사용 시 주의사항

WHERE -> GROUP BY -> HAVING 순서이므로 HAVING 은 GROUP BY 의 영향을 받음!
따라서 GROUP BY 에 입력된 컬럼에 의해서 입력 가능한 컬럼의 제약 발생

```
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 ;
```

GOOD

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

컬럼에 적용되는 일반조건이므로
HAVING -> WHERE

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

BAD

연봉합계 → SUM(연봉)

HAVING 실습

실습 문제

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

소속반	인원수
B	3
C	4

2. 직원 테이블에서 부서별 최고연봉이 7500인 튜플(행)만 출력해주세요.

부서ID	최고연봉
D001	7500
D003	7500

3. 성적표 테이블에서 학생별 평균성적을 구하되,
   평균값이 NULL이 아닌 값만 출력해주세요.
   (힌트 : HAVING 은 WHERE 절과 똑같이 NULL조건 사용가능)

학생ID	평균성적
S0001	91.7
S0002	73.3
S0003	73.3
S0004	61.7
S0005	100

HAVING 실습

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

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

2. **직원 테이블에서 부서별 최고연봉이 7500인 튜플(행)만 출력해주세요.**

```
SELECT 부서ID , MAX(연봉) AS 최고연봉
FROM 직원
GROUP BY 부서ID
HAVING MAX(연봉) = 7500 ;
```

3. **성적표 테이블에서 학생별 평균성적을 구하되, 평균값이 NULL이 아닌 값만 출력해주세요.**
   (힌트 : HAVING 은 WHERE 절과 똑같이 NULL조건 사용가능)

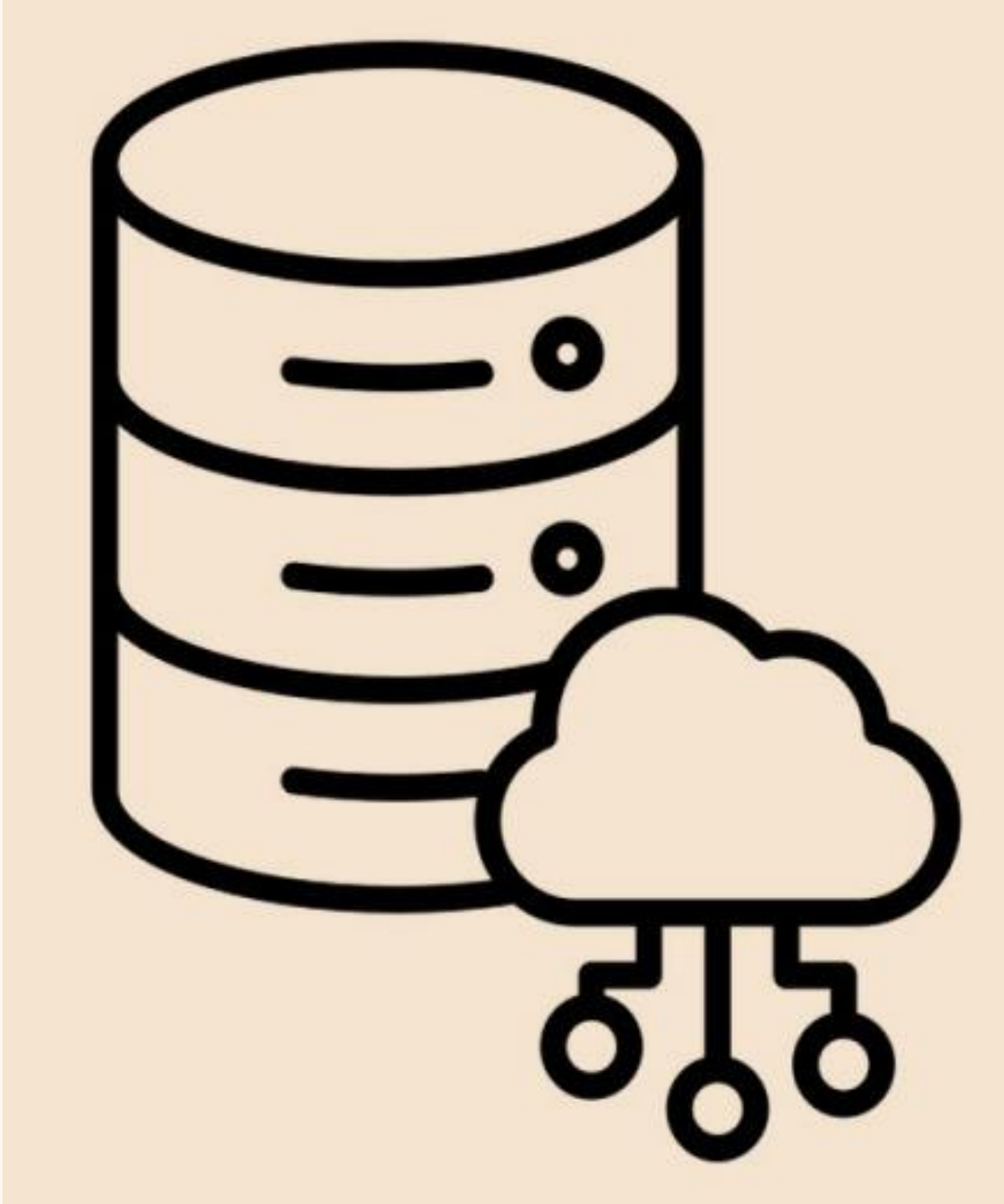
```
SELECT 학생ID , ROUND ( AVG(성적) , 1 ) AS 평균성적
FROM 성적표
GROUP BY 학생ID
HAVING AVG(성적) IS NOT NULL ;
```

ORDER BY

1 ORDER BY 문법

2 ORDER BY 원리

3 ORDER BY 사용방식



ORDER BY 문법

SQL 문법 실행 순서

5	SELECT	출력하고 싶은 컬럼만 작성하기
1	FROM	데이터를 가져올 테이블 입력
2	WHERE	원하는 튜플만 가져오도록 필터링(조건문)
3	GROUP BY	특정 컬럼을 기준으로 그룹화
4	HAVING	그룹화 상태의 데이터를 필터링
6	ORDER BY	특정 컬럼으로 정렬하기

ORDER BY 문법

ORDER BY : 특정 컬럼을 기준으로 데이터를 오름차순/내림차순 정렬

예) 직원 테이블의 모든 데이터를 출력하되 이름을 기준으로 오름차순 정렬을 해서 출력해주세요.

```
SELECT *
FROM 직원
ORDER BY 이름 ;
```

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	부서ID
A0002	hello123!	강홍수	남	28	21/09/12	950223-1562867	3000	D002
A0004	123123!!	김선미	여	(null)	20/03/11	930722-2766443	4500	D004
A0001	12345	김철수	남	25	22/03/21	991212-1566123	2800	D001
A0010	coffeegood!	김홍민	남	52	13/11/23	710513-1572876	9300	D005
A0007	pwd123	메이슨	남	40	16/08/19	830629-1676551	6200	D002
A0005	test123	문현철	남	34	(null)	891231-1786155	5000	D005
A0006	774433	송대주	남	44	15/07/16	790903-1566127	7500	D001
A0008	anjffhgkwl123	송진아	여	47	15/07/16	761212-2508143	7500	D003
A0011	newman	신입	남	(null)	23/01/03	940123-1332219	3000	D004
A0009	test123	이서연	여	50	13/11/23	730317-259616	9000	D004
A0003	nono132	이현정	여	(null)	22/11/06	000112-4566123	2600	D003

ORDER BY 문법

ORDER BY : 특정 컬럼을 기준으로 데이터를 오름차순/내림차순 정렬

예) 직원 테이블의 모든 데이터를 출력하되 연봉 기준으로 내림차순 정렬을 해서 출력해주세요.

```
SELECT *
FROM 직원
ORDER BY 연봉 DESC ;
```

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	부서ID
A0010	coffeegood!	김홍민	남	52	13/11/23	710513-1572876	9300	D005
A0009	test123	이서연	여	50	13/11/23	730317-259616	9000	D004
A0008	anjffhgkwl123	송진아	여	47	15/07/16	761212-2508143	7500	D003
A0006	774433	송대주	남	44	15/07/16	790903-1566127	7500	D001
A0007	pwd123	메이슨	남	40	16/08/19	830629-1676551	6200	D002
A0005	test123	문현철	남	34	(null)	891231-1786155	5000	(null)
A0004	123123!!	김선미	여	(null)	20/03/11	930722-2766443	4500	D004
A0002	hello123!	강홍수	남	28	21/09/12	950223-1562867	3000	D002
A0001	12345	김철수	남	25	22/03/21	991212-1566123	2800	D001
A0003	nono132	이현정	여	(null)	22/11/06	000112-4566123	2600	D003

ORDER BY 원리

ORDER BY에 여러 컬럼을 사용하면 컬럼 단위 정렬 가능

예) 먼저 직원 테이블 데이터를 부서ID 기준으로 오름차순 정렬하고,
동일한 부서ID 에 대해서는 이름을 기준으로 내림차순 정렬을 해주세요.

```
SELECT *
FROM 직원
ORDER BY 부서ID ,이름 DESC ;
```

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	부서ID
A0006	774433	송대주	남	44	15/07/16	790903-1566127	7500	D001
A0001	12345	김철수	남	25	22/03/21	991212-1566123	2800	D001
A0007	pwd123	메이슨	남	40	16/08/19	830629-1676551	6200	D002
A0002	hello123!	강홍수	남	28	21/09/12	950223-1562867	3000	D002
A0003	nono132	이현정	여	(null)	22/11/06	000112-4566123	2600	D003
A0008	anjffhgkwl123	송진아	여	47	15/07/16	761212-2508143	7500	D003
A0009	test123	이서연	여	50	13/11/23	730317-259616	9000	D004
A0011	newman	신입	남	(null)	23/01/03	940123-1332219	3000	D004
A0004	123123!!	김선미	여	(null)	20/03/11	930722-2766443	4500	D004
A0005	test123	문현철	남	34	(null)	891231-1786155	5000	D005
A0010	coffeegood!	김홍민	남	52	13/11/23	710513-1572876	9300	D005

ORDER BY 원리

<TIP!!> SELECT 에 입력되지 않은 컬럼으로도 정렬 가능 !!

(단, GROUP BY 가 명시된 경우에는 GROUP BY 에 의해 한정된 컬럼만 사용 가능)

```
SELECT 이름, 나이 , 부서ID
FROM 직원
ORDER BY 연봉 ;
```

[참고] FROM 에서 데이터를 가져올 때는 "튜플"을 기준으로 함

그 중에서 출력하려는 컬럼을 SELECT에 적는 것일 뿐! 이미 테이블 전체를 가져온 상태이므로,

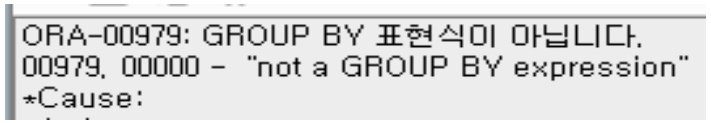
SELECT에 입력되지 않은 컬럼도 ORDER BY 에 입력 가능!

ORDER BY 원리

BUT !!

GROUP BY 에 특정 컬럼으로 그룹화가 되면 사용 가능한 컬럼에 제약 발생

```
SELECT 부서ID, SUM(연봉)
FROM 직원
GROUP BY 부서ID
ORDER BY 연봉 ;
```



SQL문이 GROUP BY -> HAVING -> SELECT -> ORDER BY 순서로 실행되기 때문

ORDER BY 사용방식

컬럼이름 외에 AS 별칭이나 숫자로도 표현 가능

```
SELECT 직원ID , 이름 ,
       연봉 AS 직원들의연봉,
       연봉 * 0.1 AS 보너스
FROM 직원
ORDER BY 3;
```

```
SELECT 직원ID , 이름 ,
       연봉 AS 직원들의연봉,
       연봉 * 0.1 AS 보너스
FROM 직원
ORDER BY 연봉;
```

```
SELECT 직원ID , 이름 ,
       연봉 AS 직원들의연봉,
       연봉 * 0.1 AS 보너스
FROM 직원
ORDER BY 직원들의연봉;
```

세 쿼리는 모두 동일하게(연봉으로) 정렬한 결과 출력