GROUP BY

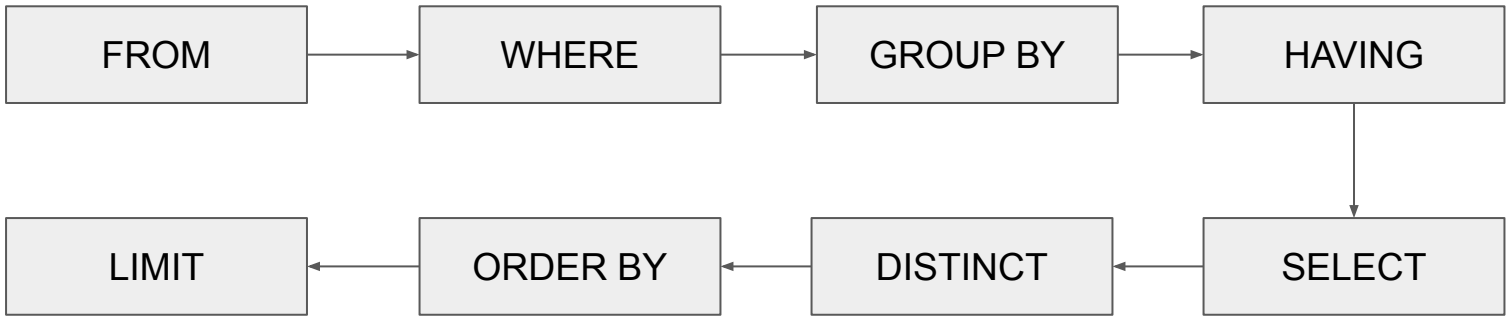
```
SELECT
        course_id ,
        count(student_id) as total_student
from enrollment
group by course_id
order by course_id
```

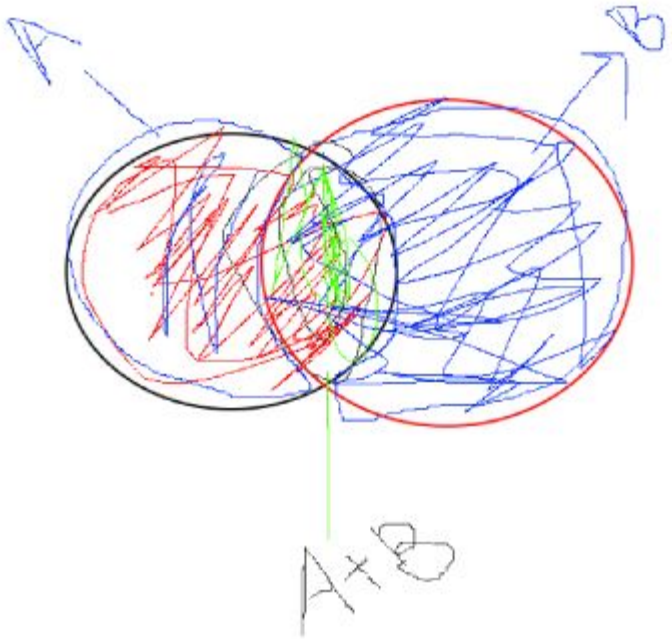
Group by with **Having**

```
-- In which course , there are at least 20 students
SELECT
	course_id as course,
	count(student_id) as total_student
from enrollment
group by course_id
HAVING count(student_id)  >= 20
order by course_id
```

```
-- In which course , there are at least 20 students
SELECT
	course_id as course,
	count(student_id) as total_student
from enrollment
WHERE course_id in (6,7,8,9,10,11,12)
group by course_id
HAVING count(student_id)  >= 20
order by total_student DESC
LIMIT 3
```

Why SQL works. (Sql query generation flow)




A
B
A+B

persons

person_id	person_name
1	A
2	B
3	C
4	D
5	E

pet

pet_id	pet_name	owner
1	Red Dog	1
2	Black Dog	NULL
3	Red Cat	NULL
4	Yellow Cow	2
5	White Rabbit	3
6	Small Cat	NULL
7	Black Cat	4
8	Purple Parrot	1
9	Mix cat	NULL
10	White Dog	2

JOIN

pet_id	pet_name	owner
1	Red Dog	1
2	Black Dog	NULL
3	Red Cat	NULL
4	Yellow Cow	2
5	White Rabbit	3
6	Small Cat	NULL
7	Black Cat	4
8	Purple Parrot	1
9	Mix cat	NULL
10	White Dog	2

person_id	person_name
1	A
2	B
3	C
4	D
5	E

JOIN

pet_id	pet_name	person_id
1	Red Dog	1
2	Black Dog	NULL
3	Red Cat	NULL
4	Yellow Cow	2
5	White Rabbit	3
6	Small Cat	NULL
7	Black Cat	4
8	Purple Parrot	1
9	Mix cat	NULL
10	White Dog	2

person_id	person_name
1	A
2	B
3	C
4	D
5	E

INNER JOIN

RULES **person.id = pet.owner**

pet_id	pet_name	owner	id	name
1	Red Dog	1	1	A
4	Yellow Cow	2	2	B
5	White Rabbit	3	3	C
7	Black Cat	4	4	D
8	Purple Parrot	1	1	A
10	White Dog	2	2	B

LEFT JOIN

Condition: person.id = pet.owner
Direction: Pet -> Person

pet_id	pet_name	owner	person_id	name
1	Red Dog	1	1	A
2	Black Dog	NULL	NULL	NULL
3	Red Cat	NULL	NULL	NULL
4	Yellow Cow	2	2	B
5	White Rabbit	3	3	C
6	Small Cat	NULL	NULL	NULL
7	Black Cat	4	4	D
8	Purple Parrot	1	1	A
9	Mix cat	NULL	NULL	NULL
10	White Dog	2	2	B

LEFT JOIN

Condition: person.id = pet.owner
Direction: Person -> Pet

person_id	person_name	id	name	**owner**
1	A	1	Red Dog	**1**
2	B	4	Yellow Cow	**2**
3	C	5	White Rabbit	**3**
4	D	7	Black Cat	**4**
5	E	NULL	NULL	**NULL**

INNER JOIN:

SELECT * from enrollment
INNER JOIN student ON student.id = enrollment.student_id

Example:

```
SELECT
	enrollment.id,
	enrollment.enrollment_date,
	course.name AS "Course Name",
	student.name AS "Student Name"
from enrollment
INNER JOIN student ON student.id = enrollment.student_id
INNER JOIN course ON course.id = enrollment.course_id
WHERE course.name = 'Python'
```

```
-- select un-enrollmented course
SELECT * from course
where id not in (select Distinct course_id from enrollment)

select * from course
left join enrollment on course.id = enrollment.course_id
Where enrollment.id is null
```

Which courses are taught by a specific instructor?

```
SELECT c.id AS course_id, c.name AS course_name
FROM course c
INNER JOIN course_instructor ci ON c.id = ci.course_id
WHERE ci.instructor_id = <specific_instructor_id>;
```

Which students are enrolled in a particular course?

```
SELECT s.id AS student_id, s.name AS student_name
FROM student s
INNER JOIN enrollment e ON s.id = e.student_id
WHERE e.course_id = <specific_course_id>;
```

What are the details of all courses along with the names of their instructors (if any)?

```
SELECT c.id AS course_id, c.name AS course_name, i.name AS instructor_name
FROM course c
LEFT JOIN course_instructor ci ON c.id = ci.course_id
LEFT JOIN instructor i ON ci.instructor_id = i.id;
```

Which instructors are associated with courses they teach?

```
SELECT i.id AS instructor_id, i.name AS instructor_name, c.name AS course_name
FROM instructor i
INNER JOIN course_instructor ci ON i.id = ci.instructor_id
INNER JOIN course c ON ci.course_id = c.id;
```

List all students and their enrollment details (if any).

```
SELECT s.id AS student_id, s.name AS student_name, e.enrollment_date
FROM student s
LEFT JOIN enrollment e ON s.id = e.student_id;
```