**Отчет по выполнению лабораторной работы №3**

**По предмету «базы данных».**

**Тема: *«Использование объединяющих и вложенных запросов языка SQL»***

**Задание 1**

**with count\_degree AS (**

**SELECT DISTINCT degree from professors**

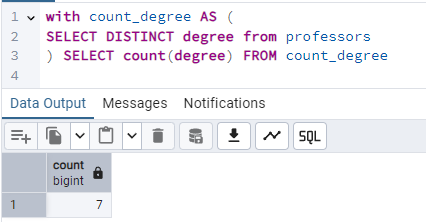
**) SELECT count(degree) FROM count\_degree**

Рассмотрите следующий запрос. Какой результат был получен после его выполнения.

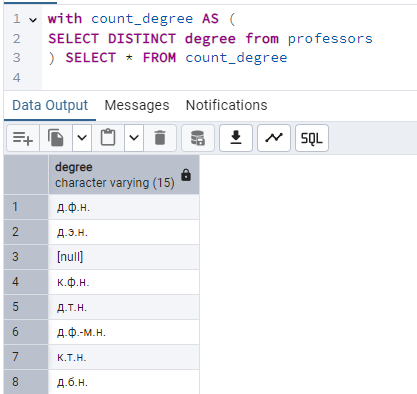
Измените запрос, поставив вместе значения degree в агрегатную функцию символ «\*».

Сравните результаты выполнения запросов и объясните его.

**Результат выполнения:**

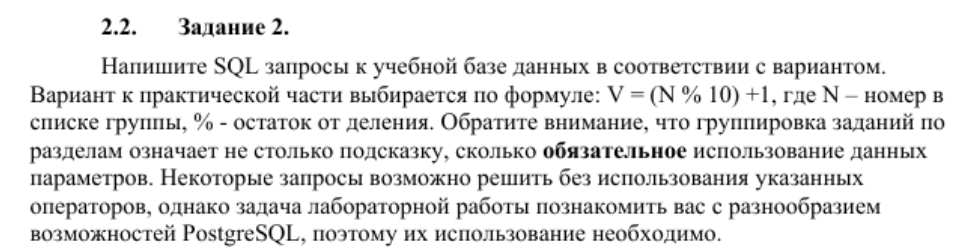
****

**Измененный запрос:**

****

После выполнения первого запроса, у нас вывелось количество уникальных образований профессоров, при этом не считалось значение null.

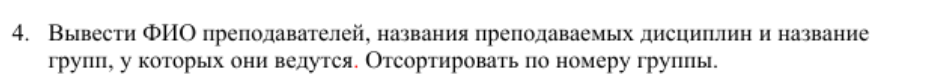
Второй запрос вывел те уникальные степни образования, которые считались в первом запросе.



Мой вариант:



**INNER JOIN**

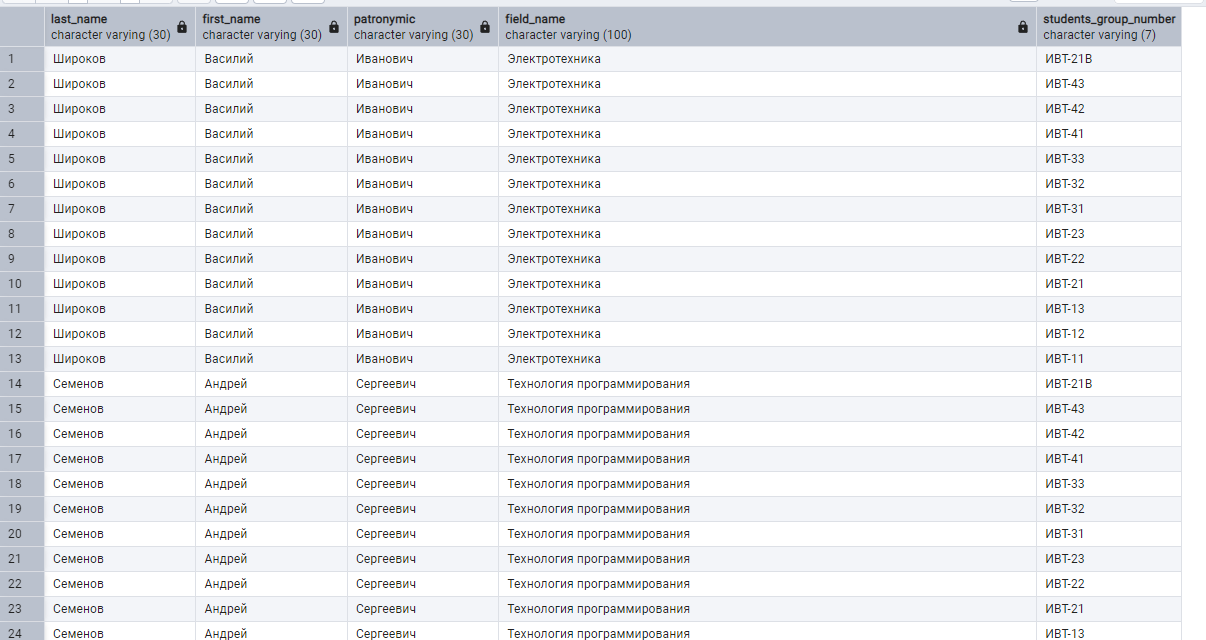


**select** p.last\_name, p.first\_name, p.patronymic, f.field\_name, s.students\_group\_number

**from** professors p **join** fields f

**on** f.professor\_id = p.professor\_id

**join** students\_groups s **on** f.structural\_unit\_id = s.structural\_unit\_id;

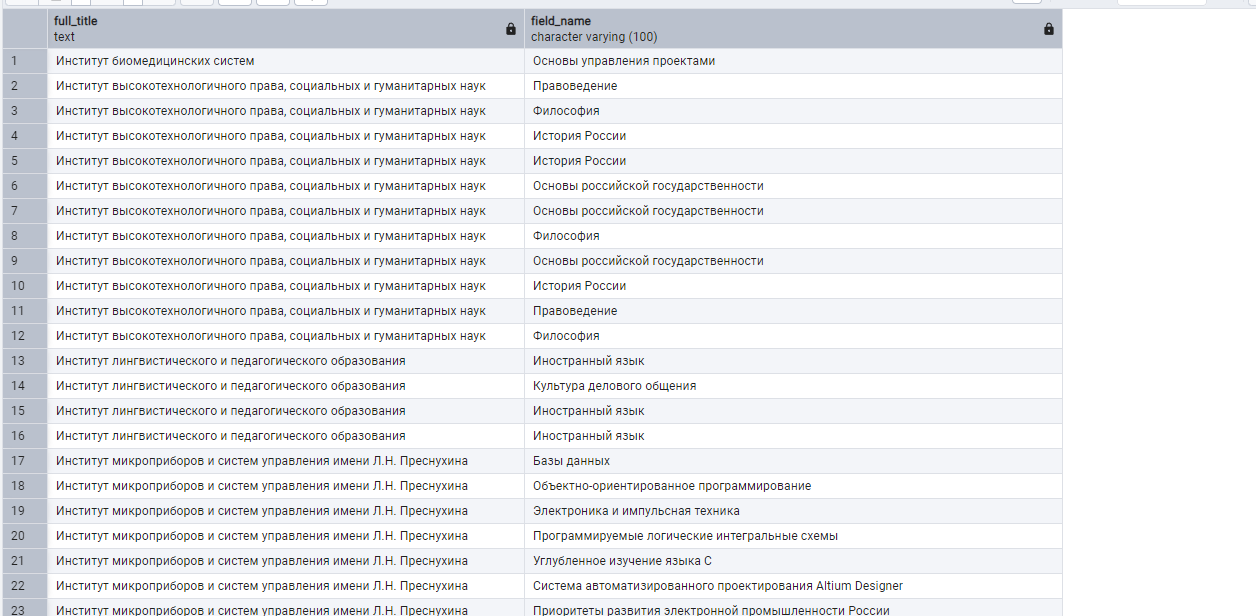




**SELECT** su.full\_title, f.field\_name **FROM** structural\_units su

**join** fields f **on** su.structural\_unit\_id = f.structural\_unit\_id

**order by** su.full\_title;





--Средняя зарплата преподавателя высшего учебного заведения по России 97 276 рублей.

**SELECT** p.last\_name, p.first\_name, p.patronymic, p.salary

**FROM** professors p

**WHERE** p.salary > '97276'::MONEY;





**select** s.last\_name, s.first\_name, s.patronymic

**from** students s join students\_groups sg

**on** s.students\_group\_number = sg.students\_group\_number

**where** sg.enrolment\_status = 'Заочная' and

**extract**(**year from age**(birthday)) >= 20

**order by** last\_name;



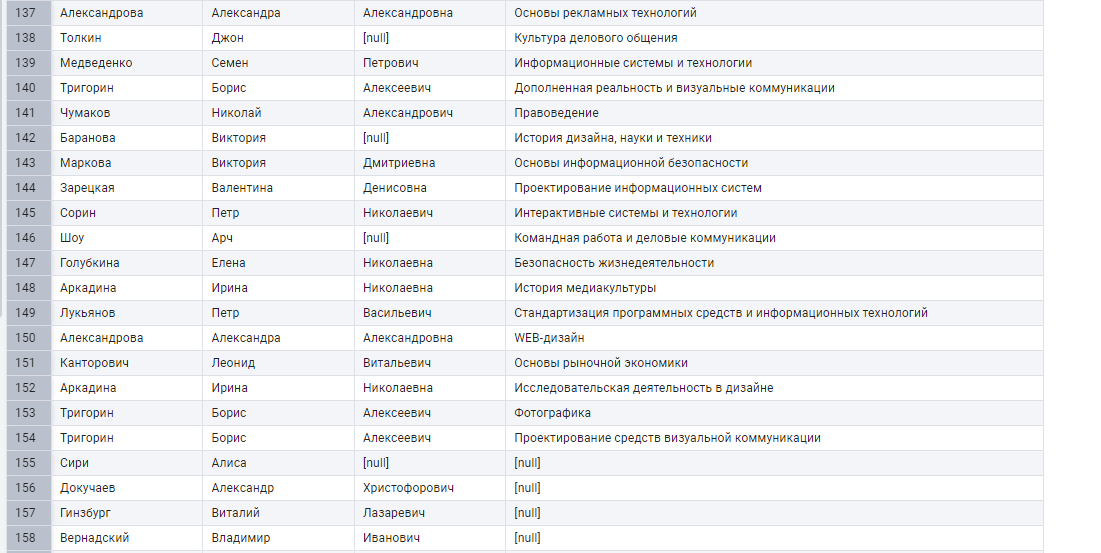
**LEFT JOIN, RIGHT JOIN, FULL JOIN**

****

**select** p.last\_name, p.first\_name, p.patronymic, f.field\_name

**from** professors p left join fields f

**on** p.professor\_id = f.professor\_id;



**UNION/EXCEPT/INTERSECT**

**select**

'максимальный оклад' as description, **max**(salary::numeric) as value **from** professors

**union all**

**select**

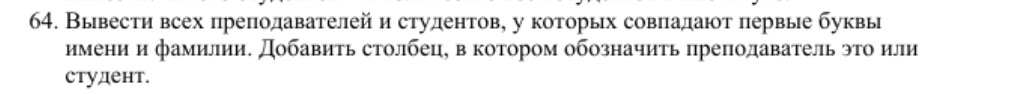
'средний оклад', **round**(**avg**(salary::numeric), 2) **from** professors

**union all**

**select**

'минимальный оклад', **min**(salary::numeric) **from** professors;





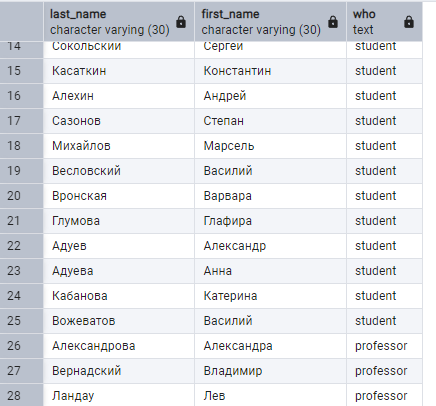
**select** last\_name, first\_name, 'student' as "who" **from** students

**where** left(last\_name, 1) = left(first\_name, 1)

**union all**

**select** last\_name, first\_name, 'professor' **from** professors

**where** left(last\_name, 1) = left(first\_name, 1) ;



**ВЛОЖЕННЫЕ ЗАПРОСЫ**

****

**select distinct on** (s.student\_id)

s.last\_name,

s.first\_name,

f.mark

**from** students s

**join** field\_comprehensions f **on** s.student\_id = f.student\_id

**where** s.students\_group\_number = 'иб-21'

**and** f.mark > (

**select avg**(fc.mark)

**from** field\_comprehensions fc

**join** students sc **on** fc.student\_id = sc.student\_id

**where** sc.students\_group\_number = 'иб-21'

);





**select** s1.last\_name, s1.first\_name, s1.students\_group\_number

**from** students s1

**where** s1.students\_group\_number = (

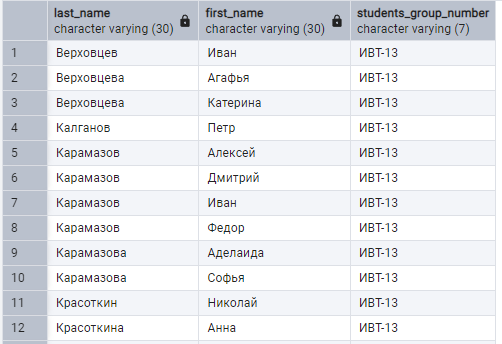
**select** students\_group\_number

**from** students s2

**group by** students\_group\_number

**order by count**(s2.student\_id) **desc**

limit 1);





--Вывести ФИО студентов и профессоров чьи фамилии одинаковые

**select** s.last\_name as "фамилия", s.first\_name as "имя студента",

s.patronymic as "отчество студента", p.first\_name as "имя преподавателя",

p.patronymic as "отчество преподавателя" **from** students s

**join** professors p **on** s.last\_name = p.last\_name



--Вывести какие группы с какими преподавателями связаны

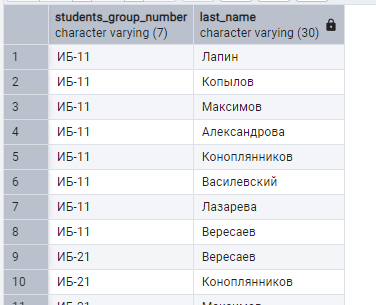
**select** sg.students\_group\_number, p.last\_name from professors p

**join** employments e **on** e.professor\_id = p.professor\_id

**right join** students\_groups sg

**on** sg.structural\_unit\_id = e.structural\_unit\_id

**order by** students\_group\_number;



--Вывести преподавателей (если есть такие), которые не относятся ни к одному предмету

**select** professor\_id **from** professors

**except**

**select** distinct professor\_id **from** fields;

