**Лабораторные работы по курсу**

**Базы данных**

**Отчет по лабораторной работе № 3**

**Работу выполнил**

**Хомыженко Николай Сергеевич**

**Москва, 202****4**

**Задание 1**

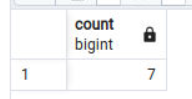
Рассмотрите следующий запрос. Какой результат был получен после его выполнения.

Измените запрос, поставив вместе значения degree в агрегатную функцию символ «\*». Сравните результаты выполнения запросов и объясните его.

**with count\_degree AS (**

**SELECT DISTINCT degree from professor**

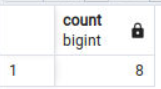
**) SELECT count(degree) FROM count\_degree;**



**with count\_degree AS (**

**SELECT DISTINCT degree from professor**

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



В первом случае мы получаем на 1 меньше, поскольку count(\*) считает так же строку с NULL значениями.

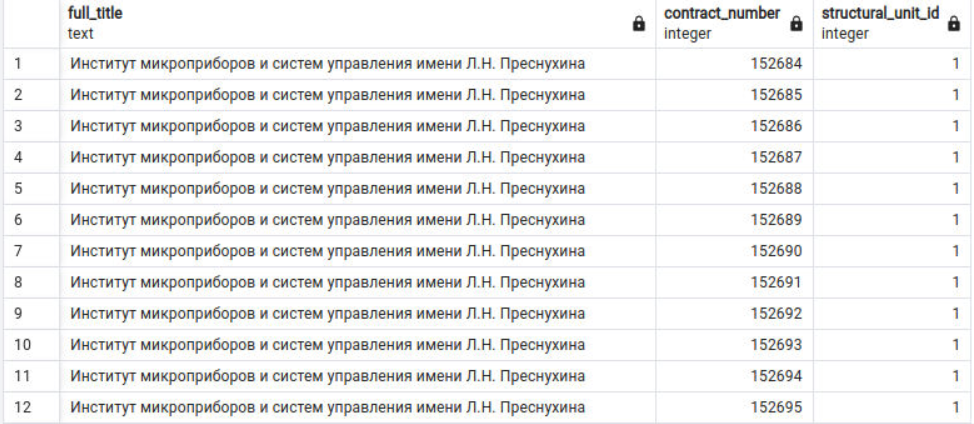
**Задание 2**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 15 | 25 | 35 | 45 | 55 | 65 | 75 | 85 | 95 | 105 |

1. Вывести номер трудового договора и название структурного подразделения, к которому он относится

**select full\_title, contract\_number, structural\_unit\_id from structural\_units**

**inner join employments using(structural\_unit\_id);**



15. Вывести ФИО и ставку всех трудоустроенных преподавателей.

**select last\_name, first\_name, patronymic, salary, wage\_rate from professors**

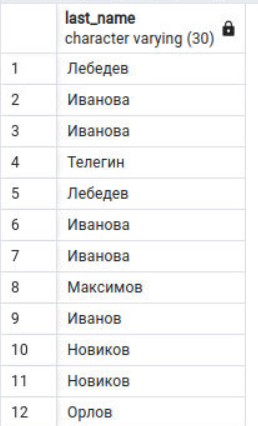
**inner join employments using(professor\_id);**



25. Вывести фамилии преподавателей и фамилии их однофамильцев среди студентов (если есть, иначе - null)

**select last\_name from professors**

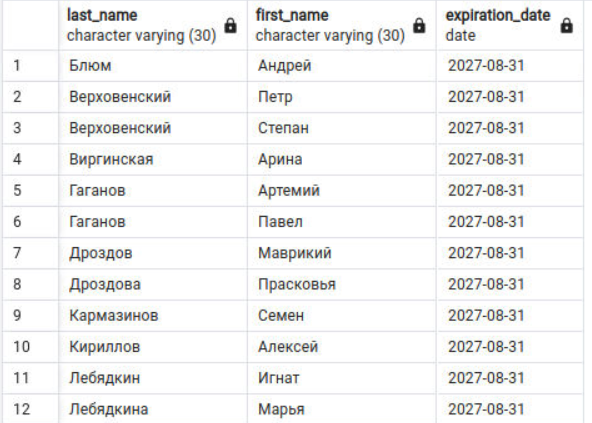
**inner join students using(last\_name);**



35. Вывести ФИО студентов-должников и дату, до которой действителен его студенческий

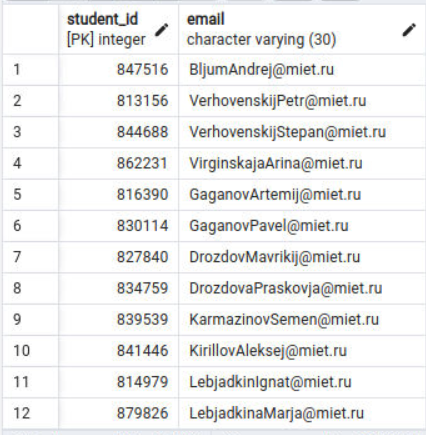
**select last\_name, first\_name, expiration\_date from students**

**inner join student\_ids using(student\_id);**



45. Вывести логины всех пользователей и почтовые адреса всех студентов

**select student\_id, email from students;**



55. Вывести ФИО преподавателей, у которых нет однофамильцев среди студентов

**select last\_name from professors**

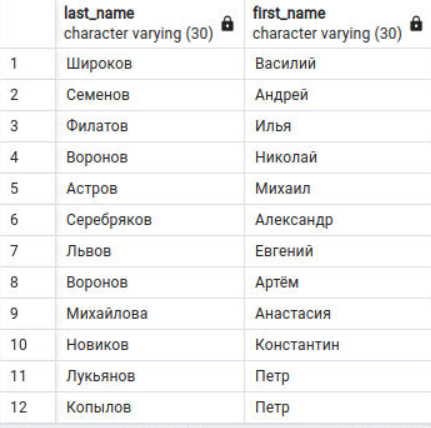
**except select last\_name from students;**



65. Вывести ФИ и оклад всех преподавателей, кроме ассистентов. Сортировать по возрастанию окладов

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

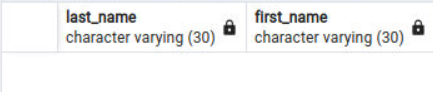
**where lower(academic\_title) != 'ассистент';**



75. Вывести номера всех нетрудоустроенных преподавателей

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

**where experience = 0;**



85. Вывести ФИ студентов и их среднюю оценку по дисциплине Базы Данных, если она больше 3

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

**AVG(fc.mark) as rate**

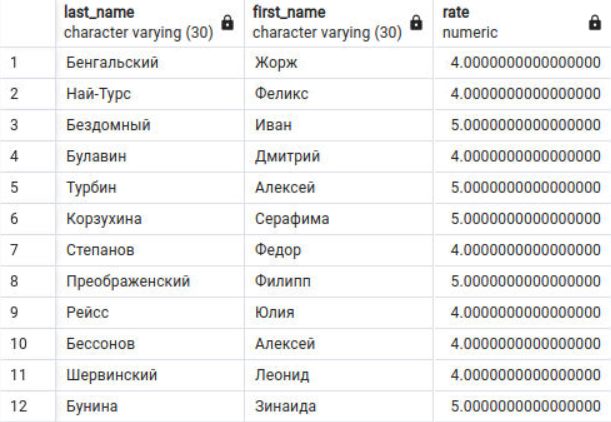
**from students s**

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

**where fc.field = '56c6b8de-f196-4b7a-b21c-db8f9eca588a'**

**group by s.last\_name, s.first\_name**

**having AVG(fc.mark) > 3;**



95. Вывести ФИ всех студентов, чей средний балл выше среднего балла в ИТД-21

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

**AVG(fc.mark) as rate**

**from students s**

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

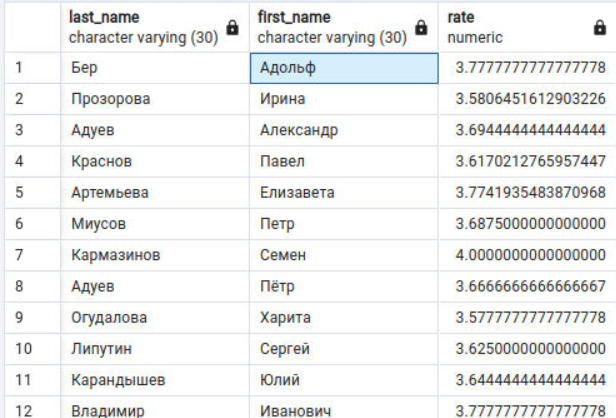
**group by s.last\_name, s.first\_name**

**having AVG(fc.mark) > (select avg(fc.mark)**

**from students s**

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

**where students\_group\_number = 'ИТД-21');**



105. Вывести ФИ преподавателя с минимальным стажем по дисциплине   Базы Данных

**select pr.last\_name, pr.first\_name, pr.experience from professors pr**

**join fields f on f.professor\_id = pr.professor\_id**

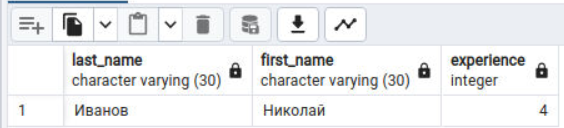
**where f.field\_name = 'Базы данных'**

**group by pr.professor\_id, pr.last\_name, pr.first\_name**

**having pr.experience = (select min(experience) from professors p**

**join fields f2 on f2.professor\_id = p.professor\_id**

**where f2.field\_name = 'Базы данных')**



**Задание 3.**

Самостоятельно разработайте 3 **осмысленных** запроса к базе данных, используя приведенные в данной лабораторной работе материалы. Вариант выбирается в соответствии с номером по списку. Из созданных запросов создайте представления.

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант: | **5,6** | | |
| Номер запроса: | **1** | **2** | **3** |
| INNER JOIN | + |  |  |
| LEFT JOIN |  | + |  |
| INTERSECT |  |  | + |