**Лабораторные работы по курсу**

**Базы данных**

**Отчет по лабораторной работе № 2**

**Работу выполнил**

**Хомыженко Николай Сергеевич**

**Москва, 202****4**

**Задание 1**

*Исследование типов данных*

Предположим, что в магазине новый конструктор стоит 999 рублей и 99 копеек. Студент С. решил приобрести для дальнейшей перепродажи 100000 таких товаров. Для расчета общей суммы, которую необходимо заплатить был создан следующий скрипт на языке PL/pgSQL. Более подробно о нём будет рассказано в одной из следующих лабораторных работ. Обратите внимание, что значение суммы имеет тип *real*.

**DO**

**$$**

**DECLARE**

**summ real :=0.0;**

**BEGIN**

**FOR i IN 1..100000 LOOP**

**summ := summ + 999.99;**

**END LOOP;**

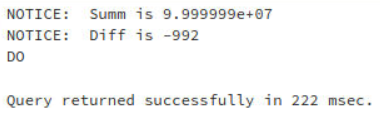
**RAISE NOTICE ‘Summ is ’, summ;**

**RAISE NOTICE ‘Diff is ’, 99999000.00 - summ;**

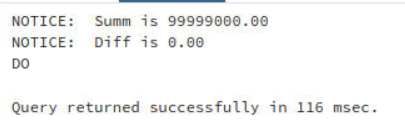
**END**

**$$ language plpgsql;**

Результат выполнения ниже:



Переплата составила 992 рубля., если поменять тип с real на numeric, то разность станет равна 0:



Чтобы поменять тип на money, надо скастовать цифры к типу money:

**DO**

**$$**

**DECLARE**

**summ money := 0.0;**

**BEGIN**

**FOR i IN 1..100000 LOOP**

**summ := summ + 999.99::money;**

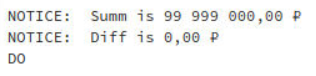
**END LOOP;**

**RAISE NOTICE ‘Summ is ’ summ;**

**RAISE NOTICE ‘Diff is ’, 99999000.00::money - summ;**

**END**

**$$ language plpgsql;**



**Задание 2**

*Написание запросов на языке SQL*

Напишите SQL запросы к учебной базе данных в соответствии с вариантом. Запросы брать из сборник запросов к учебной базе данных, расположенного ниже

|  |  |
| --- | --- |
| № варианта | № запросов |
| 5 | 5, 15, 25, 35, 45, 55, 65, 75 |

**5.** Выведите студентов, у которых дата рождения совпадает с вашей (месяц и день)

**select \* from students**

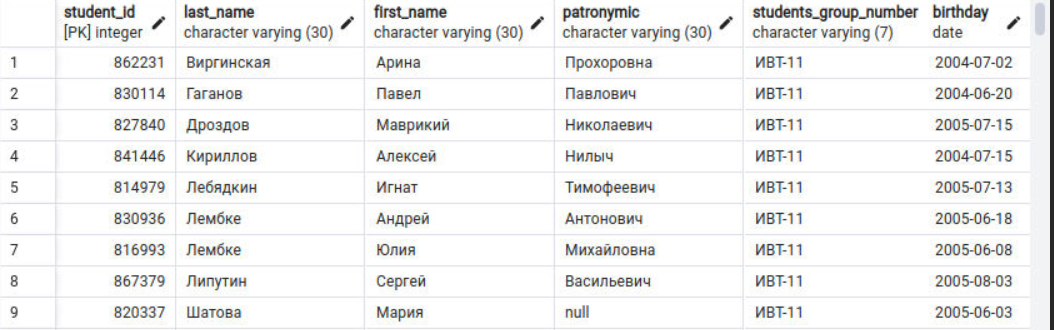
**where extract(month from birthday) = 8 and extract(day from birthday) = 8;**



15. Вывести всех студентов, родившихся под летними знаками зодиака

**select \* from students**

**where extract(month from birthday) in (6, 7, 8);**



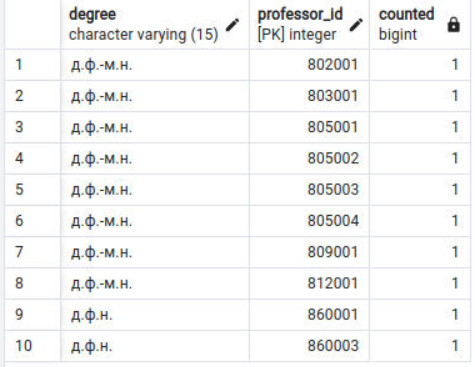
1. Посчитать число д.т.н., д.ф.н., отсортировать по Id, оставить только тех, кого больше 2

**SELECT degree, professor\_id COUNT(\*) AS counted FROM professors**

**WHERE degree N (‘д.т.н.’, ‘д.ф.-м.н.’)**

**GROUP BY degree, professor\_id**

**ORDER BY professor\_id;**



1. Вывести сколько 5 у каждого студента, отсортировать по количеству. Оставить только тех, у кого 5 больше 10

**SELECT s.first\_name, s.last\_name, s.patronymic, COUNT(fc.grade) AS count\_of\_fives**

**FROM students s**

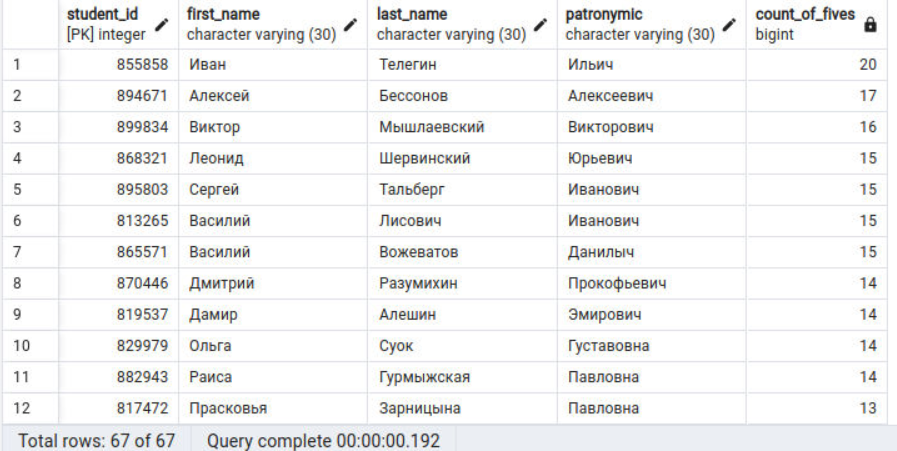
**JOIN field\_comprehensions fc ON s.student\_id = fc.student\_id**

**WHERE fc.grade = 5**

**GROUP BY s.first\_name, s.last\_name, s.patronymic**

**HAVING COUNT(fc.grade) > 10**

**ORDER BY count\_of\_fives DESC;**

****

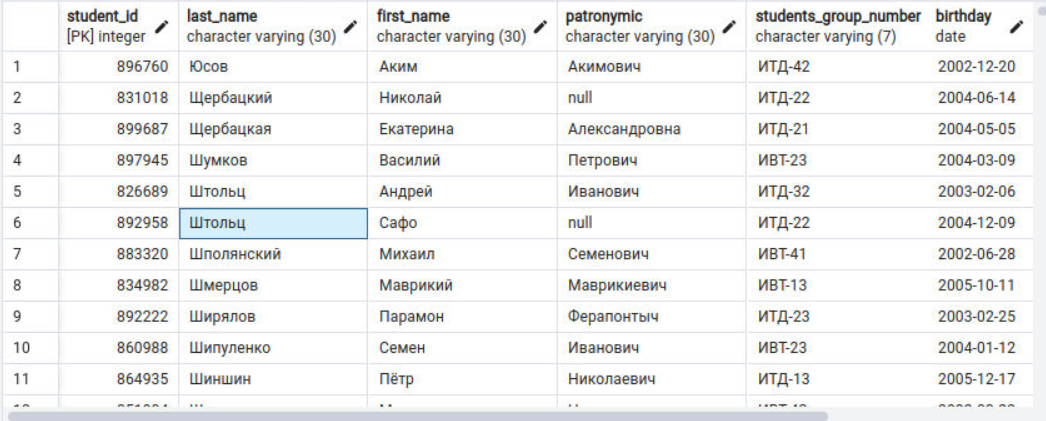
45. Вывести студентов с доменом почты .ru, отсортировать по фамилиям в обратном порядке

**SELECT \***

**FROM students**

**WHERE LOWER(RIGHT(email, 3)) = '.ru'**

**ORDER BY last\_name DESC;**

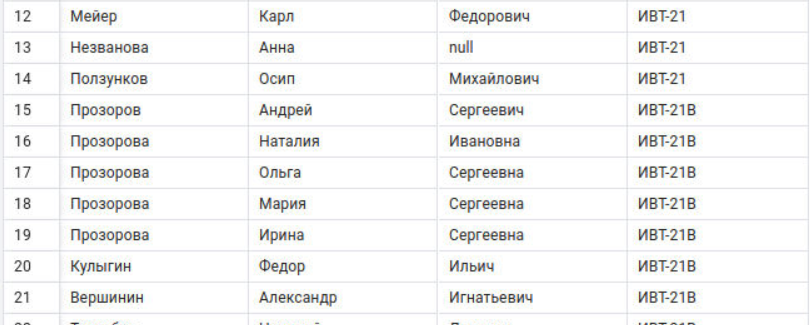
****

1. Выведите всех студентов, обучающихся на 2 курсе (ФИО, номер группы)

**SELECT first\_name, last\_name, patronymic, group\_number FROM students**

**WHERE CAST(SUBSTRING(course FROM POSITION('-' IN course) + 1 FOR 2) AS INTEGER) BETWEEN 21 AND 25;**

****

****

1. Запрос дает информацию о студентах направления «ИТД», обучающихся на 3 курсе, и возраст которых более 21 года, группируем по дате рождения и группе, т.е. в случае если одногруппники родились в один день запись будет в одной строке, и сортируем по самым старшим. (возможность получить информацию о возрастном диапазоне студентов)

**SELECT**

**birthday,**

**first\_name,**

**last\_name,**

**patronymic,**

**students\_group\_number**

**FROM**

**students**

**WHERE**

**LEFT(students\_group\_number, 5) = ‘ИТД-3’ AND**

**AND EXTRACT(YEAR FROM age(birtday)) > 21**

**GROUP BY**

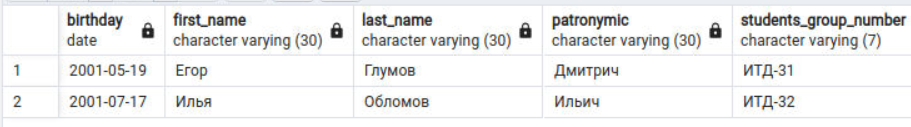
**birthday,**

**first\_name,**

**last\_name,**

**patronymic,**

**students\_group\_number;**

****

1. Найти среднюю оценку каждого студента, чей год рождения - 2002, по всем дисциплинам как "Средняя успеваемость", отсортировать, по средней оценке, оставить только тех, у кого оценка меньше 4

**SELECT s.student\_id, s.last\_name, s.first\_name, s.patronymic, s.birthday**

**AVG(fc.mark) AS average\_grade**

**FROM**

**students s**

**JOIN**

**field\_comprehensions fc ON s.student\_id = fc.student\_id**

**WHERE**

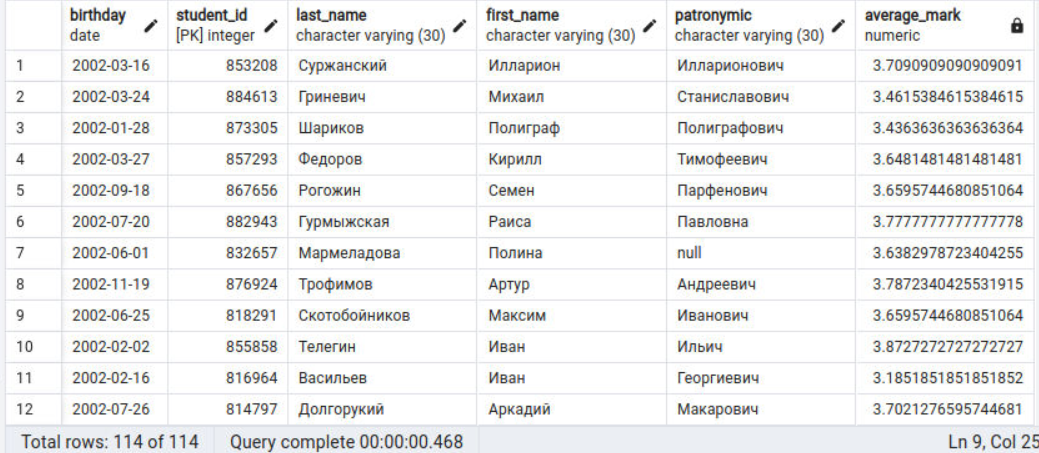
**EXTRACT(YEAR FROM s.birthday) = 2002**

**GROUP BY**

**s.birthday, s.student\_id, s.last\_name, s.first\_name**

**HAVING**

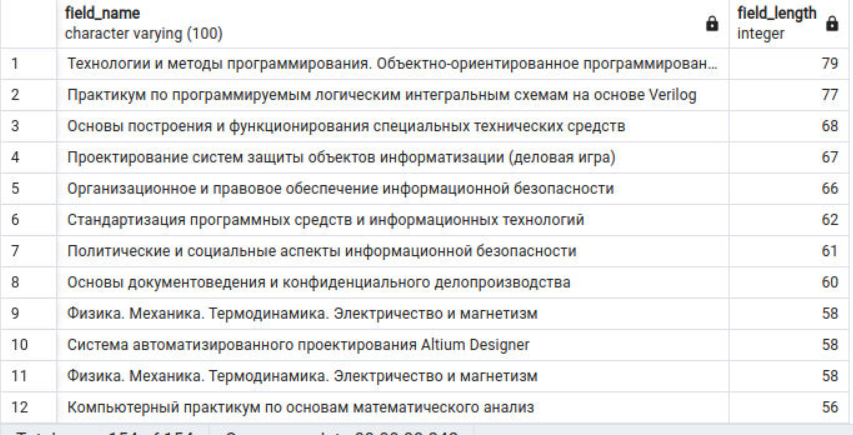
**AVG(fc.mark) < 4;**

****

85. Для оптимизации БД - нужно посчитать количество символов, используемых для названия учебных подразделений. Нужно сделать запрос, отображающий это в порядке убывания, для удобства

**SELECT field\_name, LENGTH(field\_name) AS field\_length FROM fields**

**ORDER BY field\_length DESC;**

****

**Задание 3**

Самостоятельно разработайте **7 осмысленных** запросов к базе данных, используя приведенные в данной лабораторной работе материалы.

1. Список отличников:

**SELECT s.student\_id, s.last\_name, s.first\_name, s.patronymic, s.birthday**

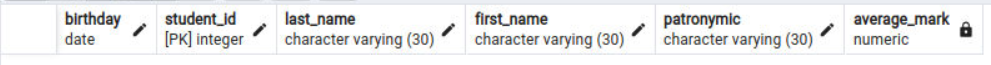
**AVG(fc.mark) AS average\_grade FROM students s**

**JOIN field\_comprehensions fc ON s.student\_id = fc.student\_id**

**GROUP BY s.birthday, s.student\_id, s.last\_name, s.first\_name**

**HAVING AVG(fc.mark) = 5;**

Жизненная ситуация- таких нет



1. Студенты Программной инженерии:

**SELECT \* FROM students**

**WHERE LEFT(students\_group\_number, 3) = ‘ПИН’;**



1. Количество студентов в группах

**SELECT students\_group\_number, COUNT(\*) as students\_count FROM students**

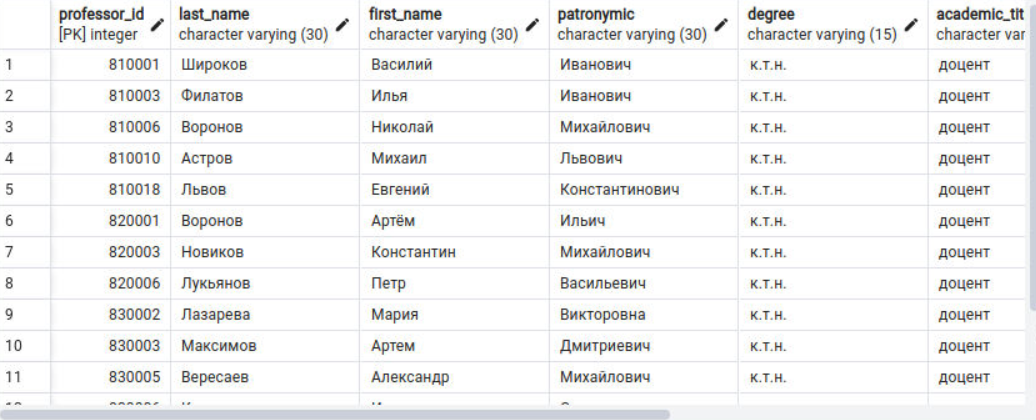
**GROUP BY students\_group\_number;**

****

1. Список доцентов:

**SELECT \* FROM professors**

**WHERE LOWER(academic\_title) = ‘доцент’;**

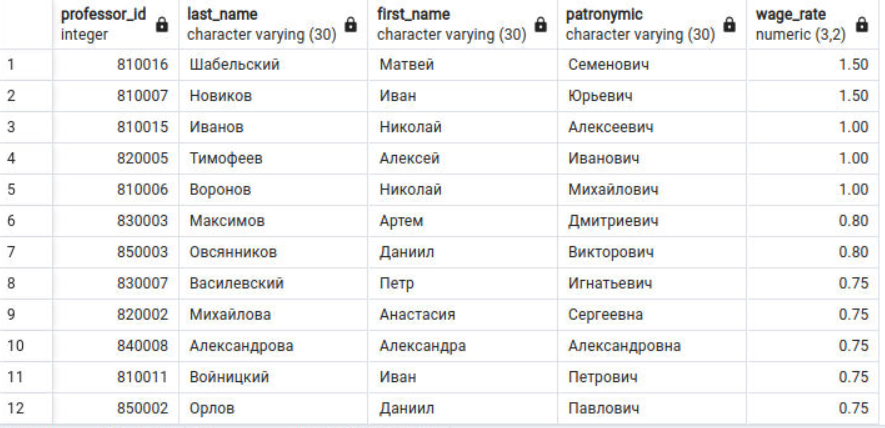
****

1. Список самых высокооплачиваемых преподавателей

**select pr.professor\_id, pr.last\_name, pr.first\_name, pr.patronymic, e.wage\_rate from professors pr**

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

**order by e.wage\_rate desc;**



1. Средний балл студентов из ИВТ:

**select s.students\_group\_number, s.student\_id, s.last\_name, s.first\_name, s.patronymic,**

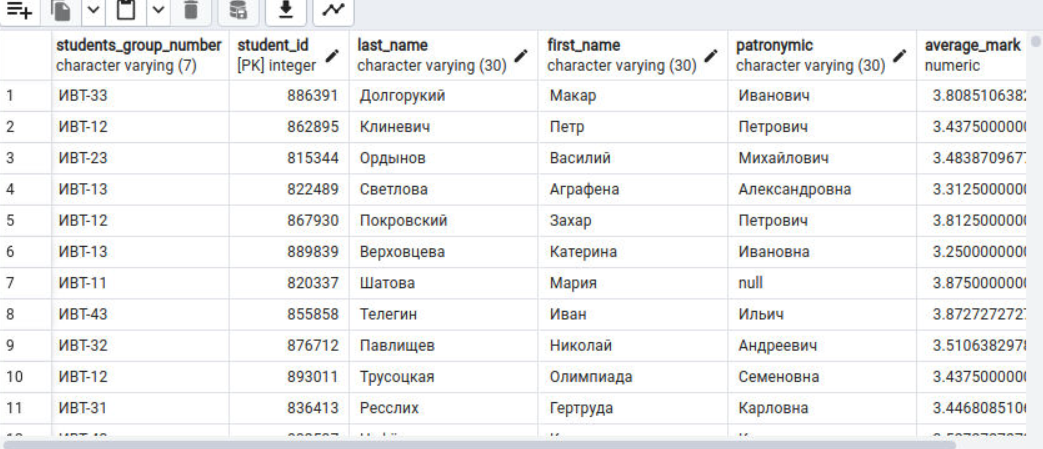
**avg(fc.mark) as average\_mark**

**from students s**

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

**where left(s.students\_group\_number, 3) = 'ИВТ'**

**group by s.birthday, s.student\_id, s.first\_name, s.last\_name;**



1. Вставка отчества-заглушки тем, у кого нет отчества

**--update students**

**--set patronymic = 'Отсутствует'**

**--where patronymic = 'null';**

**SELECT \* FROM students**

**where patronymic = 'Отсутствует';**

