Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №2**

по дисциплине «Базы данных»

**Вариант: 9987**

Выполнил:

Молчанов Фёдор Денисович

Группа: Р3113

Принял:

Николаев Владимир Вячеславович

г. Санкт-Петербург, 2024

Оглавление

[Задание 3](#_Toc162538724)

[Выполнение задания 4](#_Toc162538725)

[Заключение 7](#_Toc162538726)

# Задание

Составить запросы на языке SQL (пункты 1-7).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.  
   Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_СЕССИЯ.УЧГОД.  
   Фильтры (AND):   
   a) Н\_ЛЮДИ.ОТЧЕСТВО > Сергеевич.  
   b) Н\_СЕССИЯ.ЧЛВК\_ИД > 151200.  
   c) Н\_СЕССИЯ.ЧЛВК\_ИД > 100622.  
   Вид соединения: RIGHT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
   Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.ИД.  
   Фильтры (AND):   
   a) Н\_ЛЮДИ.ФАМИЛИЯ > Соколов.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 153285.  
   Вид соединения: INNER JOIN.
3. Вывести число студентов ФКТИУ, которые не имеет отчества.  
   Ответ должен содержать только одно число.
4. Выдать различные отчества людей и число людей с каждой из этих отчеств, ограничив список отчествами, встречающимися более 10 раз на ФКТИУ.  
   Для реализации использовать подзапрос.
5. Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст), где средний возраст меньше минимального возраста в группе 1100.
6. Получить список студентов, отчисленных после первого сентября 2012 года с заочной формы обучения. В результат включить:  
   номер группы;  
   номер, фамилию, имя и отчество студента;  
   номер пункта приказа;  
   Для реализации использовать подзапрос с IN.
7. Вывести список студентов, имеющих одинаковые фамилии, но не совпадающие даты рождения.

# Выполнение задания

/\*

    1.Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

    Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.

    Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_СЕССИЯ.УЧГОД.

    Фильтры (AND):

    a) Н\_ЛЮДИ.ОТЧЕСТВО > Сергеевич.

    b) Н\_СЕССИЯ.ЧЛВК\_ИД > 151200.

    c) Н\_СЕССИЯ.ЧЛВК\_ИД > 100622.

    Вид соединения: RIGHT JOIN.

\*/

/\*

Максимальный ид = 151200, поэтому ничего не выводит. Вот код для проверки:

SELECT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_СЕССИЯ.УЧГОД, Н\_СЕССИЯ.ЧЛВК\_ИД

FROM Н\_СЕССИЯ

INNER JOIN Н\_ЛЮДИ ON Н\_СЕССИЯ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

ORDER BY Н\_СЕССИЯ.ЧЛВК\_ИД DESC;

\*/

SELECT DISTINCT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_СЕССИЯ.УЧГОД

FROM Н\_СЕССИЯ

RIGHT JOIN Н\_ЛЮДИ ON Н\_СЕССИЯ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

WHERE Н\_ЛЮДИ.ОТЧЕСТВО > 'Сергеевич' AND Н\_СЕССИЯ.ЧЛВК\_ИД > 151200 AND Н\_СЕССИЯ.ЧЛВК\_ИД > 100622;

/\*

    2.Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

    Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.

    Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.ИД.

    Фильтры (AND):

    a) Н\_ЛЮДИ.ФАМИЛИЯ > Соколов.

    b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 153285.

    Вид соединения: INNER JOIN.

\*/

/\*

ничего не выводит, так как максимальный ид 134262. вот код для проверки:

SELECT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.ИД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

FROM Н\_ЛЮДИ

INNER JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

INNER JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД

ORDER BY Н\_ВЕДОМОСТИ.ЧЛВК\_ИД DESC;

\*/

SELECT DISTINCT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.ИД

FROM Н\_ЛЮДИ

INNER JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

INNER JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ФАМИЛИЯ > 'Соколов' AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 153285;

/\*

    3.Вывести число студентов ФКТИУ, которые не имеет отчества.

    Ответ должен содержать только одно число.

\*/

SELECT COUNT(\*)

FROM Н\_УЧЕНИКИ

RIGHT JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

JOIN Н\_ПЛАНЫ ON Н\_УЧЕНИКИ.ПЛАН\_ИД = Н\_ПЛАНЫ.ИД

JOIN Н\_ОТДЕЛЫ ON Н\_ПЛАНЫ.ОТД\_ИД = Н\_ОТДЕЛЫ.ИД

WHERE Н\_ОТДЕЛЫ.КОРОТКОЕ\_ИМЯ = 'КТиУ' AND Н\_ЛЮДИ.ОТЧЕСТВО = '.'; -- Н\_ЛЮДИ.ОТЧЕСТВО = NULL

/\*

    4.Выдать различные отчества людей и число людей с каждой из этих отчеств,

    ограничив список отчествами, встречающимися более 10 раз на ФКТИУ.

    Для реализации использовать подзапрос.

\*/

/\*

работающий код но без подзапроса

SELECT Н\_ЛЮДИ.ОТЧЕСТВО, COUNT(Н\_ЛЮДИ.ИД) AS ЧИСЛО\_ЛЮДЕЙ

FROM Н\_ЛЮДИ

JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

JOIN Н\_ПЛАНЫ ON Н\_ПЛАНЫ.ИД = Н\_УЧЕНИКИ.ПЛАН\_ИД

JOIN Н\_ОТДЕЛЫ ON Н\_ОТДЕЛЫ.ИД = Н\_ПЛАНЫ.ОТД\_ИД

WHERE Н\_ОТДЕЛЫ.КОРОТКОЕ\_ИМЯ = 'КТиУ'

GROUP BY Н\_ЛЮДИ.ОТЧЕСТВО

HAVING COUNT(Н\_ЛЮДИ.ОТЧЕСТВО) > 10;

\*/

SELECT ОТЧЕСТВО, COUNT(ОТЧЕСТВО) AS ЧИСЛО\_ЛЮДЕЙ

FROM (

    SELECT Н\_ЛЮДИ.ОТЧЕСТВО

    FROM Н\_ЛЮДИ

    JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

    JOIN Н\_ПЛАНЫ ON Н\_ПЛАНЫ.ИД = Н\_УЧЕНИКИ.ПЛАН\_ИД

    JOIN Н\_ОТДЕЛЫ ON Н\_ОТДЕЛЫ.ИД = Н\_ПЛАНЫ.ОТД\_ИД

    WHERE Н\_ОТДЕЛЫ.КОРОТКОЕ\_ИМЯ = 'КТиУ'

) AS ПОДЗАПРОС

GROUP BY ОТЧЕСТВО

HAVING COUNT(ОТЧЕСТВО) > 10;

/\*

    5.Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст),

    где средний возраст меньше минимального возраста в группе 1100.

\*/

-- SELECT MAX(EXTRACT (YEARS FROM ДАТА\_РОЖДЕНИЯ)) FROM Н\_ЛЮДИ; выдаст 2069...

--учитывать дату смерти

--у даты смерти на null проверка не требуется, так как нет строчек с null значением (проверил)

-- в ответе нет строчек, так как минимальный возраст отрицательный. Из за того, что есть люди, которые родили позже существующего года

--поиск минимального возраста в группе 1100

SELECT MIN(CASE

            WHEN Н\_ЛЮДИ.ДАТА\_СМЕРТИ != '9999-09-09' THEN (EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_СМЕРТИ) -  EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ))

            ELSE (EXTRACT (YEAR FROM CURRENT\_DATE) - EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ))

        END

        )

FROM Н\_ЛЮДИ

JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

WHERE Н\_УЧЕНИКИ.ГРУППА = '1100';

--поиск среднего возраста людей

SELECT AVG(ВОЗРАСТ\_ЛЮДЕЙ) AS Средний\_возраст\_всех\_людей

FROM(

    SELECT (CASE

                WHEN Н\_ЛЮДИ.ДАТА\_СМЕРТИ != '9999-09-09' THEN (EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_СМЕРТИ) -  EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ))

                ELSE (EXTRACT (YEAR FROM CURRENT\_DATE) - EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ))

            END

            ) AS ВОЗРАСТ\_ЛЮДЕЙ

    FROM Н\_ЛЮДИ

) AS SUBQUERY

;

--поиск среднего возраста людей в группах

SELECT Н\_УЧЕНИКИ.ГРУППА, AVG(

    CASE

        WHEN Н\_ЛЮДИ.ДАТА\_СМЕРТИ != '9999-09-09'

            THEN EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_СМЕРТИ) - EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)

        ELSE EXTRACT (YEAR FROM CURRENT\_DATE) - EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)

    END) AS ВОЗРАСТ\_ЛЮДЕЙ

FROM Н\_УЧЕНИКИ

JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

GROUP BY Н\_УЧЕНИКИ.ГРУППА;

--решение с включением двух предыдущих запросов

SELECT Н\_УЧЕНИКИ.ГРУППА, AVG(

    CASE

        WHEN Н\_ЛЮДИ.ДАТА\_СМЕРТИ != '9999-09-09'

            THEN EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_СМЕРТИ) - EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)

        ELSE EXTRACT (YEAR FROM CURRENT\_DATE) - EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)

    END

) AS Средний\_возраст

FROM Н\_УЧЕНИКИ

JOIN Н\_ЛЮДИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

GROUP BY Н\_УЧЕНИКИ.ГРУППА

HAVING AVG(

    CASE

        WHEN Н\_ЛЮДИ.ДАТА\_СМЕРТИ != '9999-09-09'

            THEN EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_СМЕРТИ) - EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)

        ELSE EXTRACT (YEAR FROM CURRENT\_DATE) - EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)

    END

) <

(

    SELECT MIN(

        CASE

            WHEN Н\_ЛЮДИ.ДАТА\_СМЕРТИ != '9999-09-09'

                THEN EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_СМЕРТИ) -  EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)

            ELSE EXTRACT (YEAR FROM CURRENT\_DATE) - EXTRACT (YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)

        END

    )

    FROM Н\_ЛЮДИ

    JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

    WHERE Н\_УЧЕНИКИ.ГРУППА = '1100'

)

;

-- SELECT MIN(EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)) AS ВОЗРАСТ

-- FROM Н\_ЛЮДИ

-- JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

-- WHERE Н\_УЧЕНИКИ.ГРУППА = '1100';

/\*

    6.Получить список студентов, отчисленных после первого сентября 2012 года с заочной формы обучения. В результат включить:

    номер группы;

    номер, фамилию, имя и отчество студента;

    номер пункта приказа;

    Для реализации использовать подзапрос с IN.

\*/

--если их отчислили после первого сентября 2012, то разница между началом и концом будет <4

SELECT Н\_УЧЕНИКИ.ГРУППА, CONCAT(Н\_ЛЮДИ.ФАМИЛИЯ, ' ', Н\_ЛЮДИ.ИМЯ, ' ', Н\_ЛЮДИ.ОТЧЕСТВО) AS ФИО, Н\_УЧЕНИКИ.В\_СВЯЗИ\_С AS НОМЕР\_ПУНКТА\_ПРИКАЗА

FROM Н\_УЧЕНИКИ

JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

JOIN Н\_ПЛАНЫ ON Н\_ПЛАНЫ.ИД = Н\_УЧЕНИКИ.ПЛАН\_ИД

JOIN Н\_ФОРМЫ\_ОБУЧЕНИЯ ON Н\_ПЛАНЫ.ФО\_ИД = Н\_ФОРМЫ\_ОБУЧЕНИЯ.ИД

WHERE Н\_ФОРМЫ\_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ  IN ('Заочная')

    AND EXTRACT(YEAR FROM Н\_УЧЕНИКИ.НАЧАЛО) - EXTRACT(YEAR FROM Н\_УЧЕНИКИ.КОНЕЦ) < 4

    AND Н\_УЧЕНИКИ.КОНЕЦ > '2012-09-01'

;

/\*

    7.Вывести список студентов, имеющих одинаковые фамилии, но не совпадающие даты рождения.

\*/

--написать без дистинкта

SELECT CONCAT(Л1.ФАМИЛИЯ, ' ', Л1.ИМЯ, ' ', Л1.ОТЧЕСТВО) AS ФИО1, Л1.ДАТА\_РОЖДЕНИЯ, CONCAT(Л2.ФАМИЛИЯ, ' ', Л2.ИМЯ, ' ', Л2.ОТЧЕСТВО) AS ФИО2, Л2.ДАТА\_РОЖДЕНИЯ

FROM Н\_ЛЮДИ AS Л1

JOIN Н\_ЛЮДИ AS Л2 ON Л1.ФАМИЛИЯ = Л2.ФАМИЛИЯ

WHERE Л1.ФАМИЛИЯ = Л2.ФАМИЛИЯ AND Л1.ДАТА\_РОЖДЕНИЯ <> Л2.ДАТА\_РОЖДЕНИЯ AND Л1.ИД <> Л2.ИД

GROUP BY Л1.ФАМИЛИЯ, Л1.ИМЯ, Л1.ОТЧЕСТВО, Л1.ДАТА\_РОЖДЕНИЯ, Л2.ФАМИЛИЯ, Л2.ИМЯ, Л2.ОТЧЕСТВО, Л2.ДАТА\_РОЖДЕНИЯ;

Запросы представлены по ссылке <https://github.com/fefumo/itmo/tree/main/DB/lab2>

# Заключение

В ходе работы я научился выполнять запросы с JOIN в psql, выполнять подзапросы, использовать некоторые агрегатные функции (avg, min, max)