*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Программная инженерия,*

*Системное и прикладное программное обеспечение*

**Лабораторная работа по №2**

по дисциплине

«Базы данных»

**Вариант №312126**

Выполнил:

Ткачев Денис Владимирович

Группа P3111

Преподаватели:

Харитонова Анастасия Евгеньевна

Оглавление

[Задание 2](#_Toc87370215)

[Ход работы 3](#_Toc87370216)

[Текст исходной программы 3](#_Toc87370217)

[Описание программы 3](#_Toc87370218)

[Трассировка 4](#_Toc87370219)

[Вариант программы с меньшим количеством команд 4](#_Toc87370220)

[Трассировка с данными числами 5](#_Toc87370221)

[Вывод 5](#_Toc87370222)

[Список литературы 5](#_Toc87370223)

## 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.ИМЯ, Н\_СЕССИЯ.ИД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ОТЧЕСТВО < Владимирович.  
   b) Н\_СЕССИЯ.ИД > 14369.  
   Вид соединения: LEFT 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.ИД, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.УЧГОД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ИД < 142095.  
   b) Н\_ВЕДОМОСТИ.ДАТА = 2022-06-08.  
   c) Н\_СЕССИЯ.ДАТА < 2002-01-04.  
   Вид соединения: RIGHT JOIN.
3. Вывести число студентов ФКТИУ, которые младше 20 лет.  
   Ответ должен содержать только одно число.
4. Найти группы, в которых в 2011 году было более 10 обучающихся студентов на кафедре вычислительной техники.  
   Для реализации использовать соединение таблиц.
5. Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст), где средний возраст больше максимального возраста в группе 1100.
6. Получить список студентов, отчисленных после первого сентября 2012 года с очной или 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 людей, не являющихся или не являвшихся студентами ФКТИУ (данные, о которых отсутствуют в таблице Н\_УЧЕНИКИ). В запросе нельзя использовать DISTINCT.

## **Реализация на PostgreSQL**

-- 1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
-- Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.  
-- Вывести атрибуты: Н\_ЛЮДИ.ИМЯ, Н\_СЕССИЯ.ИД.  
-- Фильтры (AND):  
-- a) Н\_ЛЮДИ.ОТЧЕСТВО < Владимирович.  
-- b) Н\_СЕССИЯ.ИД > 14369.  
-- Вид соединения: LEFT JOIN.  
  
SELECT "Н\_ЛЮДИ"."ИМЯ", "Н\_СЕССИЯ"."ИД" FROM "Н\_ЛЮДИ" LEFT JOIN "Н\_СЕССИЯ" ON "Н\_ЛЮДИ"."ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД" WHERE "Н\_ЛЮДИ"."ОТЧЕСТВО" < 'ОТЧЕСТВО' AND "Н\_СЕССИЯ"."ИД" > 14369;

-- 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.ИД, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.УЧГОД.  
-- Фильтры (AND):  
-- a) Н\_ЛЮДИ.ИД < 142095.  
-- b) Н\_ВЕДОМОСТИ.ДАТА = 2022-06-08.  
-- c) Н\_СЕССИЯ.ДАТА < 2002-01-04.  
-- Вид соединения: RIGHT JOIN.  
SELECT "Н\_ЛЮДИ"."ИД", "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_СЕССИЯ"."УЧГОД" FROM "Н\_ЛЮДИ" RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" RIGHT JOIN "Н\_СЕССИЯ" ON "Н\_ВЕДОМОСТИ"."СЭС\_ИД" = "Н\_СЕССИЯ"."СЭС\_ИД" -- Связь таблиц через ключ таблицы содержания\_элементов\_строк WHERE "Н\_ЛЮДИ"."ИД" < 142095 AND "Н\_СЕССИЯ"."ДАТА" < '2002-01-04' AND "Н\_ВЕДОМОСТИ"."ДАТА" = '2022-06-08';  
  
SELECT \* FROM "Н\_СЕССИЯ" WHERE "ДАТА" < '2002-01-04'; -- ноль строк, соответствующему фильтру  
  
  
-- 3. Вывести число студентов ФКТИУ, которые младше 20 лет.  
-- Ответ должен содержать только одно число.  
  
SELECT \*  
FROM "Н\_ЛЮДИ" INNER JOIN "Н\_УЧЕНИКИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" INNER JOIN "Н\_ПЛАНЫ" ON "Н\_УЧЕНИКИ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ИД" -- ПЛАН\_ИД <--> "Н\_ПЛАНЫ";.ИД "Н\_УЧЕНИКИ"."ПЛАН\_ИД", "Н\_ПЛАНЫ"."ИД","Н\_ПЛАНЫ"."ПЛАН\_ИД" INNER JOIN "Н\_ОТДЕЛЫ" ON "Н\_ПЛАНЫ"."ОТД\_ИД" = "Н\_ОТДЕЛЫ"."ИД" WHERE "Н\_ОТДЕЛЫ"."КОРОТКОЕ\_ИМЯ" = 'КТиУ' AND "Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ" > '1991-11-08'; -- SELECT age('2011-11-08'::DATE, '1991-11-08'::DATE); == 20 years  
  
-- 4. Найти группы, в которых в 2011 году было более 10 обучающихся студентов на кафедре вычислительной техники.  
-- Для реализации использовать соединение таблиц.  
SELECT "Н\_УЧЕНИКИ"."ГРУППА" FROM "Н\_УЧЕНИКИ" INNER JOIN "Н\_ПЛАНЫ" ON "Н\_УЧЕНИКИ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ИД" INNER JOIN "Н\_ОТДЕЛЫ" ON "Н\_ПЛАНЫ"."ОТД\_ИД" = "Н\_ОТДЕЛЫ"."ИД" WHERE "Н\_ОТДЕЛЫ"."ИД"=102 AND "Н\_ПЛАНЫ"."УЧЕБНЫЙ\_ГОД"= '2010/2011'-- уже на этом этапе нет людей с ВТ GROUP BY "Н\_УЧЕНИКИ"."ГРУППА" HAVING COUNT("Н\_УЧЕНИКИ"."ЧЛВК\_ИД") > 10;  
  
  
-- 5. Выведите таблицу со средним возрастом  
-- студентов во всех группах (Группа, Средний возраст),  
-- где средний возраст больше максимального возраста в группе 1100.  
  
SELECT "Н\_УЧЕНИКИ"."ГРУППА", avg(date\_part('year', age("Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ")))  
FROM "Н\_ЛЮДИ" JOIN "Н\_УЧЕНИКИ" ON "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД" GROUP BY "Н\_УЧЕНИКИ"."ГРУППА" HAVING avg(date\_part('year', age("Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ"))) < (SELECT avg(date\_part('year', age("Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ")))  
 FROM "Н\_ЛЮДИ" JOIN "Н\_УЧЕНИКИ" ON "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД" WHERE "Н\_УЧЕНИКИ"."ГРУППА" = '1100' );

-- 6. Получить список студентов, отчисленных после первого сентября 2012 года с очной или заочной формы обучения. В результат включить:  
-- номер группы;  
-- номер, фамилию, имя и отчество студента;  
-- номер пункта приказа;  
-- Для реализации использовать подзапрос с IN.  
  
SELECT "Н\_УЧЕНИКИ"."ГРУППА", "Н\_УЧЕНИКИ"."ИД", "Н\_ЛЮДИ"."ИМЯ", "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_УЧЕНИКИ"."П\_ПРКОК\_ИД" FROM "Н\_ЛЮДИ" JOIN "Н\_УЧЕНИКИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" WHERE "Н\_УЧЕНИКИ"."ПРИЗНАК" = 'отчисл' AND "Н\_УЧЕНИКИ"."СОСТОЯНИЕ" = 'утвержден' AND DATE("Н\_УЧЕНИКИ"."КОНЕЦ") > '2012-09-01' AND DATE("Н\_УЧЕНИКИ"."КОНЕЦ") != '9999-09-09' AND "Н\_УЧЕНИКИ"."ПЛАН\_ИД" IN (  
 SELECT "Н\_ПЛАНЫ"."ИД" FROM "Н\_ПЛАНЫ" JOIN "Н\_ФОРМЫ\_ОБУЧЕНИЯ" ON "Н\_ПЛАНЫ"."ФО\_ИД" = "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."ИД" WHERE "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" IN ('Очная', 'Заочная')  
 );

-- 7. Вывести список людей, не являющихся или не являвшихся студентами ФКТИУ  
-- (данные, о которых отсутствуют в таблице Н\_УЧЕНИКИ).  
-- В запросе нельзя использовать DISTINCT.  
  
  
SELECT \*  
FROM "Н\_ЛЮДИ" WHERE NOT EXISTS (  
 SELECT \*  
 FROM "Н\_УЧЕНИКИ" JOIN "Н\_ПЛАНЫ" ON "Н\_УЧЕНИКИ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ИД" JOIN "Н\_ОТДЕЛЫ" ON "Н\_ПЛАНЫ"."ОТД\_ИД" = "Н\_ОТДЕЛЫ"."ИД" AND "Н\_ОТДЕЛЫ"."КОРОТКОЕ\_ИМЯ" = 'КТиУ' WHERE "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
);

## Вывод

При выполнении лабораторной работы я познакомился с основными функциями языка SQL и диалекта PostgreSQL. Научился писать запросы, получать, агрегировать, отсеивать и сортировать полученные данные с использованием различных синтаксических конструкций языка. В результате был освоен язык DML SQL, предназначенный для работы с данными, хранящимися внутри базы данных.