Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.01 - Информатика и вычислительная техника, Компьютерные системы и технологии*

*Дисциплина «Базы данных»*

**Отчет**

**По лабораторной работе №2**

**Вариант №1313**

Студент:

Миронов Иван Николаевич

Группа:

Р3132

Преподаватель:

Афанасьев Дмитрий Борисович

Санкт-Петербург, 2024 год

# **Оглавление**

[Оглавление 2](#_Toc162892150)

[Реализация запросов на PostgreSQL 3](#_Toc162892151)

[Вывод 5](#_Toc162892152)

# Реализация запросов на PostgreSQL

*-- Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
-- Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
-- Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.  
-- Фильтры (AND):  
-- a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Перезачет.  
-- b) Н\_ВЕДОМОСТИ.ИД = 1250981.  
-- Вид соединения: RIGHT JOIN.*SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
FROM "Н\_ВЕДОМОСТИ"  
 RIGHT JOIN "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" ON "Н\_ВЕДОМОСТИ"."ВЕД\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД"  
WHERE "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" < 'Перезачет'  
 AND "Н\_ВЕДОМОСТИ"."ИД" = 1250981;  
  
*-- Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
-- Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
-- Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ГРУППА.  
-- Фильтры: (AND)  
-- a) Н\_ЛЮДИ.ИД = 142095.  
-- b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 112514.  
-- c) Н\_УЧЕНИКИ.ГРУППА = 4100.  
-- Вид соединения: RIGHT JOIN.*SELECT "Н\_ЛЮДИ"."ИД", "Н\_ОБУЧЕНИЯ"."НЗК", "Н\_УЧЕНИКИ"."ГРУППА"  
FROM "Н\_ОБУЧЕНИЯ"  
 RIGHT JOIN "Н\_ЛЮДИ" ON "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
 RIGHT JOIN "Н\_УЧЕНИКИ" ON "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ИД" = 142095  
 AND "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = 112514  
 AND "Н\_УЧЕНИКИ"."ГРУППА" = '4100';  
  
*-- Вывести число дней без учета повторений.  
-- При составлении запроса нельзя использовать DISTINCT.*SELECT *COUNT*("ДАТА")  
FROM (SELECT ("Н\_ПЛАНЫ"."ДАТА\_УТВЕРЖДЕНИЯ"::timestamp::date) AS "ДАТА"  
 FROM "Н\_ПЛАНЫ"  
 GROUP BY "Н\_ПЛАНЫ"."ДАТА\_УТВЕРЖДЕНИЯ"::timestamp::date)  
 AS "Число дней без учета повторений";  
  
*-- Найти группы, в которых в 2011 году было ровно 5 обучающихся студентов на ФКТИУ.  
-- Для реализации использовать подзапрос.*SELECT DISTINCT "ГРУППЫ\_2011"."ГРУППА"  
FROM (SELECT "Н\_УЧЕНИКИ"."ГРУППА", *COUNT*("Н\_УЧЕНИКИ"."ИД") AS "КОЛИЧЕСТВО"  
 FROM "Н\_УЧЕНИКИ"  
 WHERE "Н\_УЧЕНИКИ"."НАЧАЛО" < '2011-01-01'::date  
 AND "Н\_УЧЕНИКИ"."КОНЕЦ" > '2011-12-30'::date  
 GROUP BY "Н\_УЧЕНИКИ"."ГРУППА") AS "ГРУППЫ\_2011"  
 RIGHT JOIN "Н\_ГРУППЫ\_ПЛАНОВ" ON "Н\_ГРУППЫ\_ПЛАНОВ"."ГРУППА" = "ГРУППЫ\_2011"."ГРУППА"  
 RIGHT JOIN "Н\_ПЛАНЫ" ON "Н\_ГРУППЫ\_ПЛАНОВ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ИД"  
WHERE "ГРУППЫ\_2011"."КОЛИЧЕСТВО" = 5  
 AND "Н\_ПЛАНЫ"."ОТД\_ИД" = (SELECT "Н\_ОТДЕЛЫ"."ИД"  
 FROM "Н\_ОТДЕЛЫ"  
 WHERE "Н\_ОТДЕЛЫ"."КОРОТКОЕ\_ИМЯ" = 'КТиУ');  
  
  
  
  
*-- Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка), у которых средняя оценка не больше средней оценк(е|и) в группе 3100.*SELECT "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД",  
 *AVG*(CASE  
 WHEN ("ОЦЕНКА" NOT IN ('зачет', 'осв', 'незач', 'неявка'))  
 THEN *CAST*("ОЦЕНКА" AS INTEGER) END) AS "СР\_ОЦЕНКА",  
 "Н\_ЛЮДИ"."ФАМИЛИЯ",  
 "Н\_ЛЮДИ"."ИМЯ",  
 "Н\_ЛЮДИ"."ОТЧЕСТВО"  
FROM "Н\_ВЕДОМОСТИ"  
 JOIN "Н\_УЧЕНИКИ" ON "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
 JOIN "Н\_ЛЮДИ" ON "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
WHERE "ГРУППА" = '4100'  
GROUP BY "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД", "ФАМИЛИЯ", "ИМЯ", "ОТЧЕСТВО"  
HAVING *AVG*(CASE  
 WHEN ("ОЦЕНКА" NOT IN ('зачет', 'осв', 'незач', 'неявка'))  
 THEN *CAST*("ОЦЕНКА" AS INTEGER)  
 END) <= (SELECT *AVG*("ОЦЕНКИ\_3100"."СРЕДНИЕ")  
 FROM (SELECT *AVG*(CASE  
 WHEN ("ОЦЕНКА" NOT IN ('зачет', 'осв', 'незач', 'неявка'))  
 THEN *CAST*("ОЦЕНКА" AS INTEGER) END) AS "СРЕДНИЕ"  
 FROM "Н\_УЧЕНИКИ"  
 RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
 WHERE "Н\_УЧЕНИКИ"."ГРУППА" = '3100'  
 GROUP BY "Н\_УЧЕНИКИ"."ЧЛВК\_ИД") AS "ОЦЕНКИ\_3100");  
  
*-- Получить список студентов, отчисленных ровно 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"."ФАМИЛИЯ",  
 "Н\_ЛЮДИ"."ИМЯ",  
 "Н\_ЛЮДИ"."ОТЧЕСТВО",  
 "Н\_УЧЕНИКИ"."П\_ПРКОК\_ИД"  
FROM "Н\_УЧЕНИКИ"  
 JOIN "Н\_ЛЮДИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
 JOIN "Н\_ПЛАНЫ" ON "Н\_УЧЕНИКИ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ИД"  
 JOIN "Н\_ФОРМЫ\_ОБУЧЕНИЯ" ON "Н\_ПЛАНЫ"."ФО\_ИД" = "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."ИД"  
WHERE ("Н\_ФОРМЫ\_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Заочная' OR "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Очная')  
 AND "Н\_УЧЕНИКИ"."ИД" in (SELECT "Н\_УЧЕНИКИ"."ИД"  
 FROM "Н\_УЧЕНИКИ"  
 WHERE "Н\_УЧЕНИКИ"."ПРИЗНАК" = 'отчисл'  
 AND "Н\_УЧЕНИКИ"."СОСТОЯНИЕ" = 'утвержден'  
 AND *DATE*("Н\_УЧЕНИКИ"."КОНЕЦ") = '2012-09-01');  
  
*-- Вывести список людей, не являющихся или не являвшихся студентами ФКТИУ (данные, о которых отсутствуют в таблице Н\_УЧЕНИКИ).  
-- В запросе нельзя использовать DISTINCT*SELECT "Н\_ЛЮДИ"."ИД",  
 "Н\_ЛЮДИ"."ФАМИЛИЯ",  
 "Н\_ЛЮДИ"."ИМЯ",  
 "Н\_ЛЮДИ"."ОТЧЕСТВО"  
FROM "Н\_ЛЮДИ"  
WHERE "Н\_ЛЮДИ"."ИД" NOT IN (SELECT "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
 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"  
 WHERE "Н\_ОТДЕЛЫ"."КОРОТКОЕ\_ИМЯ" = 'КТиУ');

# Вывод

При выполнении лабораторной работы я познакомился с основными функциями 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.