Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант №785461

Лабораторная работа №2

По дисциплине

Базы Данных

Выполнил студент группы P3117:

Изаак Герман Константинович

Преподаватель:

Чупанов Аликылыч Алибекович

Санкт-Петербург 2025 г.

1. **Текст задания**

Для выполнения лабораторной работы №2 необходимо:

По варианту, выданному преподавателем, составить и выполнить запросы к [базе данных "Учебный процесс"](https://se.ifmo.ru/documents/10180/733702/%D0%91%D0%94+%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D1%8B%D0%B9+%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81.pdf/2eae3fcd-ea34-4496-924b-6ee4e889a9e5).

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

1. 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.ИД = 1.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 117219.  
   c) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 117219.  
   Вид соединения: INNER 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.ЧЛВК\_ИД, Н\_СЕССИЯ.ДАТА.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ОТЧЕСТВО = Георгиевич.  
   b) Н\_ВЕДОМОСТИ.ИД > 1426978.  
   Вид соединения: LEFT JOIN.
3. Составить запрос, который ответит на вопрос, есть ли среди студентов ФКТИУ те, кто старше 25 лет.
4. Выдать различные имена преподавателей и число людей с каждой из этих имен, ограничив список именами, встречающимися ровно 10 раз на на заочной форме обучения.  
   Для реализации использовать подзапрос.
5. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка), у которых средняя оценка больше средней оценк(е|и) в группе 3100.
6. Получить список студентов, отчисленных до первого сентября 2012 года с заочной формы обучения (специальность: 230101). В результат включить:  
   номер группы;  
   номер, фамилию, имя и отчество студента;  
   номер пункта приказа;  
   Для реализации использовать подзапрос с EXISTS.
7. Вывести список студентов, имеющих одинаковые отчества, но не совпадающие даты рождения.
8. **Реализация запросов на SQL**

--First request  
SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
FROM "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"  
INNER JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ВЕД\_ИД"  
WHERE "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = 1  
AND "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" > 117219  
AND "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = 117219;  
  
--Second request  
SELECT "Н\_ЛЮДИ"."ИД", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД", "Н\_СЕССИЯ"."ДАТА"  
FROM "Н\_ЛЮДИ"  
LEFT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
LEFT JOIN "Н\_СЕССИЯ" ON "Н\_ЛЮДИ"."ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ОТЧЕСТВО" = 'Георгиевич'  
AND "Н\_ВЕДОМОСТИ"."ИД" > 1426978;  
  
--Third request  
SELECT  
 CASE  
 WHEN *COUNT*(\*) > 0 THEN 'Да, существуют'  
 ELSE 'Нет, не существуют'  
 END AS "Есть ли студенты ФКТИУ старше 25 лет"  
FROM  
 "Н\_ЛЮДИ"  
INNER JOIN  
 "Н\_ОБУЧЕНИЯ"  
 ON "Н\_ЛЮДИ"."ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"  
INNER JOIN  
 "Н\_УЧЕНИКИ"  
 ON "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
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"."КОРОТКОЕ\_ИМЯ" = 'КТиУ'  
 AND *EXTRACT*(YEAR FROM *AGE*("Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ")) > 25;  
  
--Fourth request  
  
-- Имена преподавателей (люди не из Н\_УЧЕНИКИ)  
WITH Преподаватели AS (  
 SELECT "ИМЯ"  
 FROM "Н\_ЛЮДИ"  
 WHERE "ИД" NOT IN (SELECT "ЧЛВК\_ИД" FROM "Н\_УЧЕНИКИ")  
),  
  
-- Имена учеников на заочной форме (ровно 10 повторов)  
УченикиЗаочной AS (  
 SELECT "Н\_ЛЮДИ"."ИМЯ"  
 FROM "Н\_ЛЮДИ"  
 JOIN "Н\_УЧЕНИКИ" 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'  
 GROUP BY "Н\_ЛЮДИ"."ИМЯ"  
 HAVING *COUNT*(\*) = 10  
),  
  
-- Имена преподавателей, которые встречаются ровно 10 раз у заочников  
ПодходящиеИмена AS (  
 SELECT "ИМЯ"  
 FROM Преподаватели  
 WHERE "ИМЯ" IN (SELECT "ИМЯ" FROM УченикиЗаочной)  
)  
  
-- Итоговый подсчет всех людей с подходящими именами  
SELECT  
 "ИМЯ" AS "Имя преподавателя",  
 *COUNT*(\*) AS "Общее количество людей с этим именем"  
FROM  
 "Н\_ЛЮДИ"  
WHERE  
 "ИМЯ" IN (SELECT "ИМЯ" FROM ПодходящиеИмена)  
GROUP BY  
 "ИМЯ";  
  
--Fifth request  
WITH СредняяОценка3100 AS (  
 SELECT  
 *AVG*("ОЦЕНКА"::numeric) AS Средняя  
 FROM  
 "Н\_ВЕДОМОСТИ"  
 INNER JOIN "Н\_УЧЕНИКИ"  
 ON "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
 INNER JOIN "Н\_ГРУППЫ\_ПЛАНОВ"  
 ON "Н\_УЧЕНИКИ"."ГРУППА" = "Н\_ГРУППЫ\_ПЛАНОВ"."ГРУППА"  
 WHERE  
 "Н\_ГРУППЫ\_ПЛАНОВ"."ГРУППА" = '3100'  
 AND "Н\_ВЕДОМОСТИ"."ОЦЕНКА" ~ '^[2-5]$'  
),  
  
Студенты4100 AS (  
 SELECT  
 "Н\_УЧЕНИКИ"."ЧЛВК\_ИД",  
 "Н\_ЛЮДИ"."ФАМИЛИЯ",  
 "Н\_ЛЮДИ"."ИМЯ",  
 "Н\_ЛЮДИ"."ОТЧЕСТВО",  
 "Н\_ВЕДОМОСТИ"."ОЦЕНКА"::numeric AS Оценка  
 FROM  
 "Н\_УЧЕНИКИ"  
 INNER JOIN "Н\_ГРУППЫ\_ПЛАНОВ"  
 ON "Н\_УЧЕНИКИ"."ГРУППА" = "Н\_ГРУППЫ\_ПЛАНОВ"."ГРУППА"  
 INNER JOIN "Н\_ЛЮДИ"  
 ON "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
 LEFT JOIN "Н\_ВЕДОМОСТИ"  
 ON "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
 WHERE  
 "Н\_ГРУППЫ\_ПЛАНОВ"."ГРУППА" = '4100'  
 AND "Н\_ВЕДОМОСТИ"."ОЦЕНКА" ~ '^[2-5]$'  
)  
  
SELECT  
 Студенты4100."ЧЛВК\_ИД" AS "Номер",  
 Студенты4100."ФАМИЛИЯ" AS "Фамилия",  
 Студенты4100."ИМЯ" AS "Имя",  
 Студенты4100."ОТЧЕСТВО" AS "Отчество",  
 *AVG*(Студенты4100.Оценка) AS "Ср\_оценка"  
FROM  
 Студенты4100  
CROSS JOIN СредняяОценка3100  
GROUP BY  
 Студенты4100."ЧЛВК\_ИД",  
 Студенты4100."ФАМИЛИЯ",  
 Студенты4100."ИМЯ",  
 Студенты4100."ОТЧЕСТВО"  
HAVING  
 *AVG*(Студенты4100.Оценка) > (SELECT Средняя FROM СредняяОценка3100);  
  
--Sixth request  
SELECT  
 "Н\_УЧЕНИКИ"."ГРУППА" AS "Номер группы",  
 "Н\_ЛЮДИ"."ФАМИЛИЯ" AS "Фамилия",  
 "Н\_ЛЮДИ"."ИМЯ" AS "Имя",  
 "Н\_ЛЮДИ"."ОТЧЕСТВО" AS "Отчество",  
 "Н\_УЧЕНИКИ"."КОНЕЦ\_ПО\_ПРИКАЗУ" AS "Номер пункта приказа"  
FROM  
 "Н\_УЧЕНИКИ"  
INNER JOIN  
 "Н\_ЛЮДИ" ON "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
WHERE  
 "Н\_УЧЕНИКИ"."КОНЕЦ\_ПО\_ПРИКАЗУ" < TIMESTAMP '2012-09-01 00:00:00'  
 AND *EXISTS* (  
 SELECT 1  
 FROM  
 "Н\_ПЛАНЫ"  
 INNER JOIN  
 "Н\_ФОРМЫ\_ОБУЧЕНИЯ" ON "Н\_ПЛАНЫ"."ФО\_ИД" = "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."ИД"  
 INNER JOIN  
 "Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ" ON "Н\_ПЛАНЫ"."НАПС\_ИД" = "Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ"."ИД"  
 INNER JOIN  
 "Н\_НАПР\_СПЕЦ" ON "Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ"."НС\_ИД" = "Н\_НАПР\_СПЕЦ"."ИД"  
 WHERE  
 "Н\_ПЛАНЫ"."ИД" = "Н\_УЧЕНИКИ"."ПЛАН\_ИД"  
 AND "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Заочная'  
 AND "Н\_НАПР\_СПЕЦ"."КОД\_НАПРСПЕЦ" = '230101'  
 );  
  
--Seventh request  
SELECT  
 "Н\_ЛЮДИ"."ФАМИЛИЯ",  
 "Н\_ЛЮДИ"."ИМЯ",  
 "Н\_ЛЮДИ"."ОТЧЕСТВО",  
 "Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ"  
FROM  
 "Н\_ЛЮДИ"  
WHERE  
 *EXISTS* (  
 SELECT 1  
 FROM "Н\_ЛЮДИ" AS "Н\_ЛЮДИ\_2"  
 WHERE  
 "Н\_ЛЮДИ"."ОТЧЕСТВО" = "Н\_ЛЮДИ\_2"."ОТЧЕСТВО" -- Одинаковое отчество  
 AND "Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ" <> "Н\_ЛЮДИ\_2"."ДАТА\_РОЖДЕНИЯ" -- Разные даты  
 )  
 AND "Н\_ЛЮДИ"."ОТЧЕСТВО" IS NOT NULL  
ORDER BY  
 "Н\_ЛЮДИ"."ОТЧЕСТВО",  
 "Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ";

1. **Вывод по работе**

При выполнении лабораторной работы я познакомился с основными операторами PostgreSQL. Научился использовать различные типы соединений таблиц, использовать подзапросы, фильтровать, группировать и сортировать данные.