УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.01 Информатика и вычислительная техника

Дисциплина «Информационный системы и базы данных»

**Лабораторная работа №2**

*Вариант 3106*

Студент

*Баянов Р. Д.*

*P3134*

Преподаватель

*Перцев Т.*

Санкт-Петербург, 2023 г.

Оглавление

[**Задание** 3](#_Toc131095624)

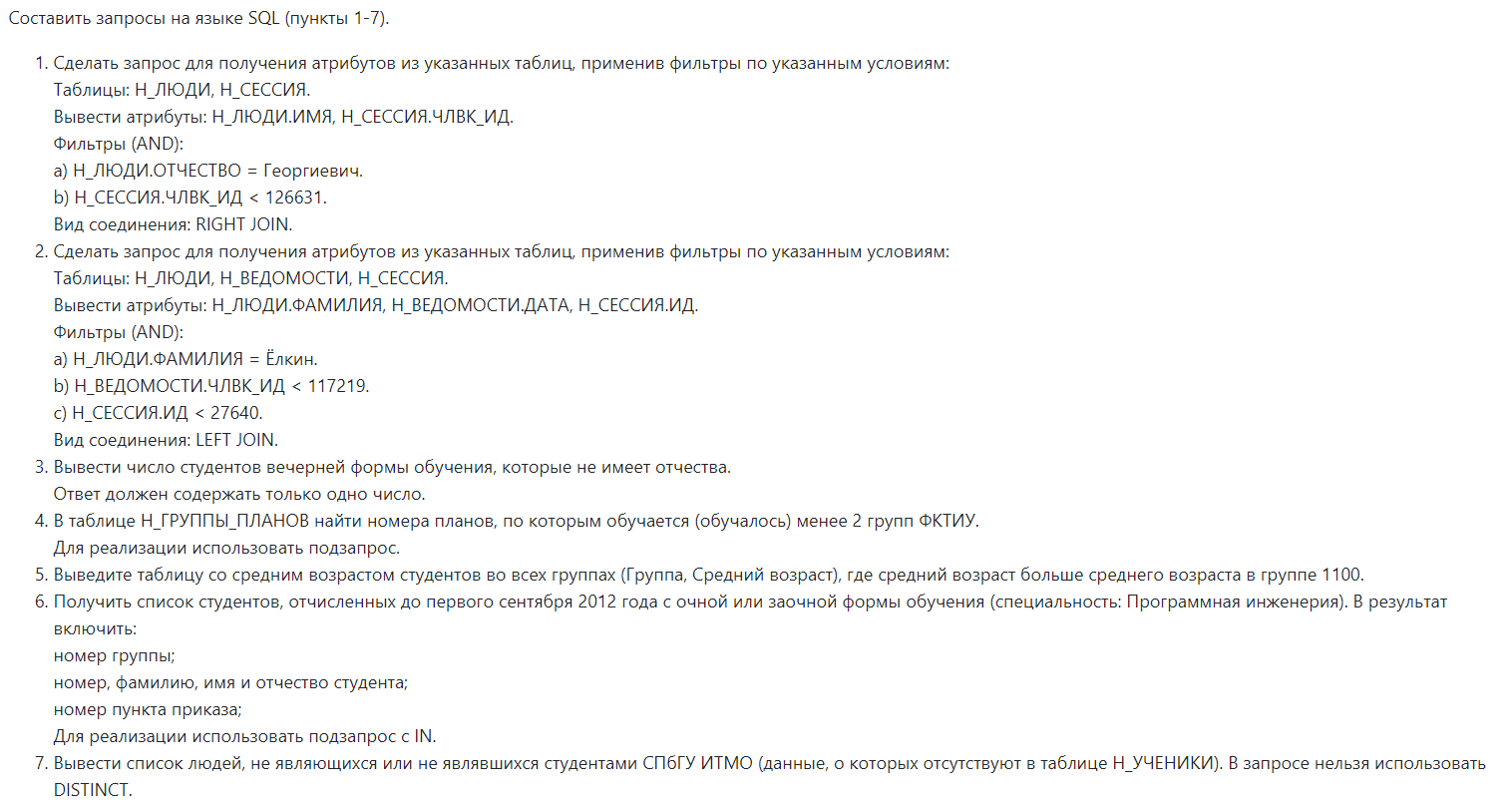
[**Реализацию запросов на SQL** 4](#_Toc131095625)

[**Вывод** 5](#_Toc131095626)

# **Задание**

По варианту, выданному преподавателем, составить и выполнить запросы к базе данных "Учебный процесс".

**Вариант**



# **Реализацию запросов на SQL**

--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.ЧЛВК\_ИД.  
--Фильтры (AND):  
--a) Н\_ЛЮДИ.ОТЧЕСТВО = Георгиевич.  
--b) Н\_СЕССИЯ.ЧЛВК\_ИД < 126631.  
--Вид соединения: RIGHT JOIN.  
SELECT "Н\_ЛЮДИ"."ИМЯ", "Н\_СЕССИЯ"."ЧЛВК\_ИД"  
FROM "Н\_ЛЮДИ"  
 RIGHT JOIN "Н\_СЕССИЯ" ON "Н\_ЛЮДИ"."ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ОТЧЕСТВО" = 'Георгиевич'  
 AND "Н\_СЕССИЯ"."ЧЛВК\_ИД" < 126631;  
  
--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 = Ёлкин.  
--b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219.  
--c) Н\_СЕССИЯ.ИД < 27640.  
--Вид соединения: LEFT JOIN.  
SELECT "Н\_ЛЮДИ"."ФАМИЛИЯ", "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_СЕССИЯ"."ИД"  
FROM "Н\_ЛЮДИ"  
 LEFT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
 LEFT JOIN "Н\_СЕССИЯ" ON "Н\_СЕССИЯ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
WHERE "Н\_ЛЮДИ"."ФАМИЛИЯ" = 'Ёлкин'  
 AND "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" < 117219  
 AND "Н\_СЕССИЯ"."ИД" < 27640;  
  
--3)Вывести число студентов вечерней формы обучения, которые не имеет отчества.  
--Ответ должен содержать только одно число.  
SELECT *count*(\*)  
FROM "Н\_ПЛАНЫ"  
 JOIN "Н\_ФОРМЫ\_ОБУЧЕНИЯ" ON "Н\_ПЛАНЫ"."ФО\_ИД" = "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."ИД"  
 JOIN "Н\_УЧЕНИКИ" ON "Н\_УЧЕНИКИ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ФО\_ИД"  
 JOIN "Н\_ЛЮДИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ОТЧЕСТВО" IS NULL  
 AND "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."ИМЯ\_В\_ИМИН\_ПАДЕЖЕ" = 'вечерняя';  
  
--4)В таблице Н\_ГРУППЫ\_ПЛАНОВ найти номера планов, по которым обучается (обучалось) менее 2 групп ФКТИУ.  
--Для реализации использовать подзапрос.  
SELECT "Н\_ГРУППЫ\_ПЛАНОВ"."ПЛАН\_ИД"  
FROM "Н\_ГРУППЫ\_ПЛАНОВ"  
 JOIN "Н\_ПЛАНЫ" ON "Н\_ГРУППЫ\_ПЛАНОВ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ИД"  
WHERE "ОТД\_ИД" IN (SELECT "ИД"  
 FROM "Н\_ОТДЕЛЫ"  
 WHERE "КОРОТКОЕ\_ИМЯ" = 'КТиУ')  
GROUP BY "Н\_ГРУППЫ\_ПЛАНОВ"."ПЛАН\_ИД"  
HAVING *COUNT*(\*) < 2;  
  
  
--5)Выведите таблицу со средним возрастом студентов во всех группах  
-- (Группа, Средний возраст), где средний возраст больше среднего 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 до первого сентября 2012 года с очной или заочной формы обучения  
-- (специальность: Программная инженерия). В результат включить:  
--номер группы;  
--номер, фамилию, имя и отчество студента;  
--номер пункта приказа;  
--Для реализации использовать подзапрос с IN.  
SELECT  
 "ГРУППА",  
 "Н\_УЧЕНИКИ"."ИД",  
 "ФАМИЛИЯ",  
 "ИМЯ",  
 "ОТЧЕСТВО",  
 "В\_СВЯЗИ\_С"  
FROM  
 "Н\_УЧЕНИКИ"  
 JOIN "Н\_ЛЮДИ" ON "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
 JOIN "Н\_ПЛАНЫ" ON "Н\_УЧЕНИКИ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ПЛАН\_ИД"  
 JOIN "Н\_НАПР\_СПЕЦ" ON "Н\_ПЛАНЫ"."НАПС\_ИД" = "Н\_НАПР\_СПЕЦ"."ИД"  
WHERE  
 "Н\_УЧЕНИКИ"."ПРИЗНАК" = 'отчисл'  
 AND "Н\_УЧЕНИКИ"."СОСТОЯНИЕ" = 'утвержден'  
 AND "Н\_НАПР\_СПЕЦ"."НАИМЕНОВАНИЕ" = 'Программная инженерия'  
 AND "Н\_УЧЕНИКИ"."КОНЕЦ" < '2012-09-01'::date  
 AND "ФО\_ИД" IN (1, 3);  
  
  
--7)Вывести список людей, не являющихся или не являвшихся студентами  
-- СПбГУ ИТМО (данные, о которых отсутствуют в таблице Н\_УЧЕНИКИ).  
-- В запросе нельзя использовать DISTINCT.  
SELECT "people"."ИД", "people"."ФАМИЛИЯ", "people"."ИМЯ", "people"."ОТЧЕСТВО"  
FROM "Н\_ЛЮДИ" AS people  
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\_ПЛАНЫ"."ИД"  
 AND "Н\_ОТДЕЛЫ"."КОРОТКОЕ\_ИМЯ" = 'СПбГУ'  
 WHERE "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "people"."ИД"));

# **Вывод**

При выполнении данной лабораторной работы я ещё больше погрузился в мир SQL запросов, изучил разные способы фильтрации, соединение таблиц и подзапросы.