

**Министерство образования и науки Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО**  
**ОБРАЗОВАНИЯ**

**“ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИТМО”**

**Факультет программной инженерии и компьютерной техники (ПИКТ)**

**Направление подготовки (специальность) – 09.03.04 (Нейротехнологии и**  
**программная инженерия)**

**Базы данных**

**Лабораторная работа № 2**

**Выполнил**  
**студент**  
**Мясников Артём Валерьевич**  
**Группа № Р3123**

**г. Санкт-Петербург**  
**2025 г.**

## Оглавление

<b>Отчет:</b> .....	3
<b>Список заданий:</b> .....	3
<b>Номер 1:</b> .....	3
<b>Номер 2:</b> .....	3
<b>Номер 3:</b> .....	4
<b>Номер 4:</b> .....	4
<b>Номер 5:</b> .....	4
<b>Номер 6:</b> .....	5
<b>Номер 7:</b> .....	5
<b>Дополнительный номер:</b> .....	6
<b>Вывод:</b> .....	7

## Вариант: 522

### Отчет:

### Список заданий:

Список заданий представлен на рисунке

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.ИД.  
Фильтры (AND):  
а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Экзаменационный лист.  
б) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 142390.  
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.ИМЯ, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.УЧГОД.  
Фильтры (AND):  
а) Н\_ЛЮДИ.ИД > 152862.  
б) Н\_ВЕДОМОСТИ.ДАТА > 2022-06-08.  
с) Н\_СЕССИЯ.УЧГОД < 2001/2002.  
Вид соединения: LEFT JOIN.
3. Вывести число фамилий без учета повторов.  
При составлении запроса нельзя использовать DISTINCT.
4. Найти группы, в которых в 2011 году было ровно 10 обучающихся студентов на кафедре вычислительной техники.  
Для реализации использовать соединение таблиц.
5. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка), у которых средняя оценка равна средней оценк(е)и в группе 1100.
6. Получить список студентов, отчисленных до первого сентября 2012 года с очной формы обучения (специальность: 230101). 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 СПбГУ ИТМО (данные, о которых отсутствуют в таблице Н\_УЧЕНИКИ). В запросе нельзя использовать DISTINCT.

### Список заданий

#### Номер 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.ИД. Фильтры (AND): а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Экзаменационный лист. б) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 142390. Вид соединения: RIGHT JOIN.

```
-- Number 1
SELECT "Н_ТИПЫ_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н_ВЕДОМОСТИ"."ИД"
FROM "Н_ТИПЫ_ВЕДОМОСТЕЙ"
RIGHT OUTER JOIN "Н_ВЕДОМОСТИ" ON "Н_ВЕДОМОСТИ"."ТВ_ИД"="Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД"
WHERE "Н_ТИПЫ_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" < 'Экзаменационный лист'
AND "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" < 142390;
```

#### Номер 1

#### Номер 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.ИМЯ, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.УЧГОД. Фильтры (AND): а) Н\_ЛЮДИ.ИД > 152862. б) Н\_ВЕДОМОСТИ.ДАТА > 2022-06-08. с) Н\_СЕССИЯ.УЧГОД < 2001/2002.

```
-- Number 2
SELECT "Н_люди"."ИМЯ", "Н_ВЕДОМОСТИ"."ИД", "Н_СЕССИЯ"."УЧГОД"
FROM "Н_люди"
LEFT OUTER JOIN "Н_ВЕДОМОСТИ" ON "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_люди"."ИД"
LEFT OUTER JOIN "Н_СЕССИЯ" ON "Н_СЕССИЯ"."ЧЛВК_ИД" = "Н_люди"."ИД"
WHERE "Н_люди"."ИД" > 152862
AND "Н_ВЕДОМОСТИ"."ДАТА" > '2022-06-08'
AND "Н_СЕССИЯ"."УЧГОД" < '2001/2002';
```

*Номер 2*

### Номер 3:

Вывести число фамилий без учета повторений. При составлении запроса нельзя использовать DISTINCT.

```
-- Number 3
SELECT COUNT(*) AS Unique_last_names_count
FROM (
    SELECT "Н_люди"."ФАМИЛИЯ"
    FROM "Н_люди"
    GROUP BY "Н_люди"."ФАМИЛИЯ"
)
AS Unique_last_names;
```

*Номер 3*

### Номер 4:

Найти группы, в которых в 2011 году было ровно 10 обучающихся студентов на кафедре вычислительной техники. Для реализации использовать соединение таблиц.

```
-- Number 4
SELECT "Н_УЧЕНИКИ"."ГРУППА"
FROM "Н_УЧЕНИКИ"
INNER JOIN "Н_ПЛАНЫ" ON "Н_УЧЕНИКИ"."ПЛАН_ИД" = "Н_ПЛАНЫ"."ПЛАН_ИД"
INNER JOIN "Н_ОТДЕЛЫ" ON "Н_ОТДЕЛЫ"."ИД" = "Н_ПЛАНЫ"."ОТД_ИД"
WHERE "Н_УЧЕНИКИ"."КОНЕЦ" >= '2011-01-01'::timestamp
AND "Н_УЧЕНИКИ"."НАЧАЛО" < '2012-01-01'::timestamp
AND (
    "Н_ОТДЕЛЫ"."ИД" = (
        SELECT "Н_ОТДЕЛЫ"."ОТД_ИД"
        FROM "Н_ОТДЕЛЫ"
        WHERE "ИМЯ_В_ИМИН_ПАДЕЖЕ" = 'кафедра вычислительной техники'
    )
    OR "Н_ОТДЕЛЫ"."ИМЯ_В_ИМИН_ПАДЕЖЕ" = 'кафедра вычислительной техники'
)
GROUP BY "Н_УЧЕНИКИ"."ГРУППА", "Н_ОТДЕЛЫ"."ИМЯ_В_ИМИН_ПАДЕЖЕ"
HAVING COUNT("Н_УЧЕНИКИ"."ГРУППА") = 10
ORDER BY "Н_УЧЕНИКИ"."ГРУППА";
```

*Номер 4*

### Номер 5:

Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка), у которых средняя оценка равна средней оценк(е|и) в группе 1100.

```
-- Number 5
SELECT "Н_УЧЕНИКИ"."ИД", "Н_ЛЮДИ"."ФАМИЛИЯ", ROUND((AVG("Н_ВЕДОМОСТИ"."ОЦЕНКА"::int)::numeric), 2) AS СРЕДНЯЯ_ОЦЕНКА
FROM "Н_УЧЕНИКИ"
INNER JOIN "Н_ЛЮДИ" ON "Н_ЛЮДИ"."ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
INNER JOIN "Н_ВЕДОМОСТИ" ON "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
WHERE "Н_УЧЕНИКИ"."ГРУППА" = '4100'
AND "Н_ВЕДОМОСТИ"."ОЦЕНКА" IN ('5', '4', '3', '2')
GROUP BY "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_УЧЕНИКИ"."ИД"
HAVING ROUND((AVG("Н_ВЕДОМОСТИ"."ОЦЕНКА"::int)::numeric), 2) = (
SELECT ROUND((AVG("Н_ВЕДОМОСТИ"."ОЦЕНКА"::int)::numeric), 2)
FROM "Н_УЧЕНИКИ"
INNER JOIN "Н_ЛЮДИ" ON "Н_ЛЮДИ"."ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
INNER JOIN "Н_ВЕДОМОСТИ" ON "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
WHERE "Н_УЧЕНИКИ"."ГРУППА" = '1100'
AND "Н_ВЕДОМОСТИ"."ОЦЕНКА" IN ('5', '4', '3', '2')
)
ORDER BY "Н_УЧЕНИКИ"."ИД";
```

*Номер 5*

## Номер 6:

Получить список студентов, отчисленных до первого сентября 2012 года с очной формы обучения (специальность: 230101). В результат включить: номер группы; номер, фамилию, имя и отчество студента; номер пункта приказа; Для реализации использовать подзапрос с IN.

```
-- Number 6
SELECT DISTINCT "Н_УЧЕНИКИ"."ГРУППА", "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ЛЮДИ"."ИМЯ", "Н_ЛЮДИ"."ОТЧЕСТВО", "Н_УЧЕНИКИ"."П_ПРКОК_ИД"
FROM "Н_УЧЕНИКИ"
INNER JOIN "Н_ОБУЧЕНИЯ" ON "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
INNER JOIN "Н_ЛЮДИ" ON "Н_ЛЮДИ"."ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
INNER JOIN "Н_ПЛАНЫ" ON "Н_ПЛАНЫ"."ПЛАН_ИД" = "Н_УЧЕНИКИ"."ПЛАН_ИД"
INNER JOIN "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ" ON "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."НАПС_ИД" = "Н_ПЛАНЫ"."НАПС_ИД"
INNER JOIN "Н_НАПР_СПЕЦ" ON "Н_НАПР_СПЕЦ"."ИД" = "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."НС_ИД"
INNER JOIN "Н_ФОРМЫ_ОБУЧЕНИЯ" ON "Н_ФОРМЫ_ОБУЧЕНИЯ"."ИД" = "Н_ПЛАНЫ"."ФО_ИД"
WHERE "Н_УЧЕНИКИ"."ПРИЗНАК" = 'отчисл'
AND "Н_УЧЕНИКИ"."КОНЕЦ" < '2012-09-01'::timestamp
AND "Н_НАПР_СПЕЦ"."КОД_НАПРСПЕЦ" = '230101'
AND "Н_УЧЕНИКИ"."ИД" IN (
SELECT "Н_УЧЕНИКИ"."ИД"
FROM "Н_УЧЕНИКИ"
INNER JOIN "Н_ПЛАНЫ" ON "Н_ПЛАНЫ"."ПЛАН_ИД" = "Н_УЧЕНИКИ"."ПЛАН_ИД"
INNER JOIN "Н_ФОРМЫ_ОБУЧЕНИЯ" ON "Н_ФОРМЫ_ОБУЧЕНИЯ"."ИД" = "Н_ПЛАНЫ"."ФО_ИД"
WHERE "Н_ФОРМЫ_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Очная'
)
ORDER BY "Н_УЧЕНИКИ"."ГРУППА", "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ЛЮДИ"."ИМЯ";
```

*Номер 6*

## Номер 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.

```
-- Number 7
SELECT "Н_ЛЮДИ"."ИД", "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ЛЮДИ"."ИМЯ", "Н_ЛЮДИ"."ОТЧЕСТВО", "Н_ЛЮДИ"."ДАТА_РОЖДЕНИЯ"::date
FROM "Н_ЛЮДИ"
LEFT OUTER JOIN "Н_УЧЕНИКИ" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
WHERE "Н_УЧЕНИКИ"."ЧЛВК_ИД" IS NULL
OR "Н_УЧЕНИКИ"."ПРИЗНАК" = 'отчисл'
ORDER BY "Н_ЛЮДИ"."ИД";
```

**Дополнительный номер:**

Для каждой группы посчитать, сколько отличников, хорошистов и простых студентов

```
-- доп. для каждой группы посчитать сколько отличников, хорошистов и простых студентов
WITH student_categories AS (
    SELECT
        "Н_УЧЕНИКИ"."ГРУППА",
        "Н_УЧЕНИКИ"."ИД",
        CASE
            WHEN EXISTS (
                SELECT 1
                FROM "Н_ВЕДОМОСТИ"
                WHERE "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
                AND "Н_ВЕДОМОСТИ"."ОЦЕНКА" = 'незач'
            ) THEN 'двоечник'
            WHEN avg_grade >= 4.5 THEN 'отличник'
            WHEN avg_grade >= 3.5 THEN 'хорошист'
            WHEN avg_grade >= 2.5 THEN 'троечник'
            WHEN avg_grade < 2.5 THEN 'двоечник'
            ELSE 'особый случай'
        END AS "ТИП_СТУДЕНТА"
    FROM "Н_УЧЕНИКИ"
    INNER JOIN (
        SELECT
            "Н_ВЕДОМОСТИ"."ЧЛВК_ИД",
            ROUND(AVG(
                CASE
                    WHEN "Н_ВЕДОМОСТИ"."ОЦЕНКА" = '5' THEN 5
                    WHEN "Н_ВЕДОМОСТИ"."ОЦЕНКА" = '4' THEN 4
                    WHEN "Н_ВЕДОМОСТИ"."ОЦЕНКА" = '3' THEN 3
                    WHEN "Н_ВЕДОМОСТИ"."ОЦЕНКА" = '2' THEN 2
                    WHEN "Н_ВЕДОМОСТИ"."ОЦЕНКА" = 'зачет' THEN 5
                    ELSE NULL
                END
            )::numeric, 2) AS avg_grade
        FROM "Н_ВЕДОМОСТИ"
        GROUP BY "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
    ) AS student_grades ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = student_grades."ЧЛВК_ИД"
)
SELECT
    student_categories."ГРУППА",
    COUNT(*) FILTER (WHERE student_categories."ТИП_СТУДЕНТА" = 'отличник') AS "Отличников",
    COUNT(*) FILTER (WHERE student_categories."ТИП_СТУДЕНТА" = 'хорошист') AS "Хорошистов",
    COUNT(*) FILTER (WHERE student_categories."ТИП_СТУДЕНТА" = 'троечник') AS "Троечников",
    COUNT(*) FILTER (WHERE student_categories."ТИП_СТУДЕНТА" = 'двоечник') AS "Двоечников",
    COUNT(*) FILTER (WHERE student_categories."ТИП_СТУДЕНТА" = 'особый случай') AS "Особых случаев",
    COUNT(*) AS "Всего"
FROM student_categories
GROUP BY student_categories."ГРУППА"
ORDER BY student_categories."ГРУППА"::int;
```

**Вывод:**

В ходе лабораторной работы я освоил практические навыки работы с SQL, научившись составлять сложные запросы с разными типами соединений (RIGHT JOIN, LEFT JOIN) и многоуровневой фильтрацией. Особое внимание уделил работе с подзапросами, агрегирующими функциями и поиску данных.