

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И
КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине

«Базы данных»

Вариант № 2834923

Выполнил:

Студент группы Р3107

Пшеничников Артём Дмитриевич

Преподаватель:

Байрамова Хумай Бахруз Кызы

Содержание

Задание

Реализация запросов

Вывод

Задание

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

Реализация запросов

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н_ОЦЕНКИ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ОЦЕНКИ.ПРИМЕЧАНИЕ, Н_ВЕДОМОСТИ.ЧЛВК_ИД.

Фильтры (AND):

а) Н_ОЦЕНКИ.КОД < 4.

б) Н_ВЕДОМОСТИ.ДАТА < 2022-06-08.

с) Н_ВЕДОМОСТИ.ДАТА = 2010-06-18.

Вид соединения: LEFT JOIN.

```
SELECT
    "Н_ОЦЕНКИ"."ПРИМЕЧАНИЕ",
    "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
FROM
    "Н_ОЦЕНКИ"
    LEFT JOIN
    "Н_ВЕДОМОСТИ" ON "Н_ОЦЕНКИ"."КОД" = "Н_ВЕДОМОСТИ"."ОЦЕНКА"
WHERE
    "Н_ОЦЕНКИ"."КОД" IN ('3', '2')
    AND "Н_ВЕДОМОСТИ"."ДАТА" < DATE '2022-06-08'
    AND "Н_ВЕДОМОСТИ"."ДАТА" = DATE '2010-06-18';
```

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.ОТЧЕСТВО, Н_ВЕДОМОСТИ.ЧЛВК_ИД, Н_СЕССИЯ.ДАТА.

Фильтры (AND):

а) Н_ЛЮДИ.ФАМИЛИЯ > Ёлкин.

б) Н_ВЕДОМОСТИ.ИД < 1490007.

Вид соединения: INNER JOIN.

```
SELECT
    "Н_люди"."ОТЧЕСТВО",
    "Н_ведомости"."ЧЛВК_ИД",
    "Н_сессия"."ДАТА"
FROM
    "Н_люди"
        INNER JOIN
    "Н_ведомости" ON "Н_люди"."ИД" = "Н_ведомости"."ЧЛВК_ИД"
        INNER JOIN
    "Н_сессия" ON "Н_ведомости"."СЭС_ИД" = "Н_сессия"."ИД"
WHERE
    "Н_люди"."ФАМИЛИЯ" > 'Ёлкин'
AND "Н_ведомости"."ИД" < 1490007;
```

3. Вывести число имен без учета повторений.

При составлении запроса нельзя использовать DISTINCT.

```
SELECT count("ИМЯ") AS "Число имен без учета повторений"
FROM (SELECT "Н_люди"."ИМЯ"
        FROM "Н_люди"
        GROUP BY "Н_люди"."ИМЯ")
    AS "table";
```

4. Выдать различные отчества преподавателей и число людей с каждой из этих отчеств, ограничив список отчествами, встречающимися менее 50 раз на ФКТИУ. Для реализации использовать соединение таблиц.

```
SELECT "Н_люди"."ОТЧЕСТВО",  
       COUNT("Н_люди"."ИД") AS "количество_людей"  
FROM "Н_люди"  
     LEFT JOIN  
     "Н_ОБУЧЕНИЯ" ON "Н_люди"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД"  
     LEFT JOIN  
     "Н_УЧЕНИКИ" ON "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"  
     LEFT JOIN  
     "Н_ПЛАНЫ" ON "Н_УЧЕНИКИ"."ПЛАН_ИД" = "Н_ПЛАНЫ"."ИД"  
     LEFT JOIN  
     "Н_ОТДЕЛЫ" ON "Н_ПЛАНЫ"."ОТД_ИД" = "Н_ОТДЕЛЫ"."ИД"  
WHERE "Н_ОТДЕЛЫ"."КОРОТКОЕ_ИМЯ" = 'КТИУ' -- с ФКТИУ нет результатов  
      AND "Н_люди"."ОТЧЕСТВО" IN (SELECT "Н_люди"."ОТЧЕСТВО"  
                                   FROM "Н_люди"  
                                   LEFT JOIN  
                                   "Н_ОБУЧЕНИЯ" ON "Н_люди"."ИД" =  
                                   "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД"  
                                   WHERE "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" IS NULL  
                                   AND "Н_люди"."ОТЧЕСТВО" IS NOT NULL  
                                   GROUP BY "Н_люди"."ОТЧЕСТВО") -- отчества преподав  
GROUP BY "Н_люди"."ОТЧЕСТВО"  
HAVING COUNT("Н_люди"."ИД") < 50;
```

5. Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст), где средний возраст больше среднего возраста в группе 3100.

```
SELECT "Н_УЧЕНИКИ"."ГРУППА",  
       DATE_PART('year', AVG(AGE("Н_люди"."ДАТА_РОЖДЕНИЯ"))) "СРЕДНИЙ_ВОЗРАСТ"  
FROM "Н_УЧЕНИКИ"  
     JOIN "Н_люди" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_люди"."ИД"  
GROUP BY "Н_УЧЕНИКИ"."ГРУППА"  
HAVING DATE_PART('year', AVG(AGE("Н_люди"."ДАТА_РОЖДЕНИЯ"))) >  
       (SELECT DATE_PART('year', AVG(AGE("Н_люди"."ДАТА_РОЖДЕНИЯ")))  
        "СРЕДНИЙ_ВОЗРАСТ"  
        FROM "Н_УЧЕНИКИ"  
             JOIN "Н_люди" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_люди"."ИД"  
        WHERE CAST("Н_УЧЕНИКИ"."ГРУППА" AS INTEGER) = 3100  
        GROUP BY "Н_УЧЕНИКИ"."ГРУППА");
```

6. Получить список студентов, отчисленных после 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 "Н_УЧЕНИКИ"."ГРУППА",
       "Н_УЧЕНИКИ"."ИД",
       "Н_ЛЮДИ"."ФАМИЛИЯ",
       "Н_ЛЮДИ"."ИМЯ",
       "Н_ЛЮДИ"."ОТЧЕСТВО",
       "Н_УЧЕНИКИ"."ПРИЗНАК",
       "Н_ОТДЕЛЫ"."КОРОТКОЕ_ИМЯ"

from "Н_УЧЕНИКИ"
      LEFT JOIN
"Н_ОБУЧЕНИЯ" ON "Н_УЧЕНИКИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД"
      LEFT JOIN
"Н_ЛЮДИ" ON "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
      LEFT JOIN
"Н_ПЛАНЫ" ON "Н_УЧЕНИКИ"."ПЛАН_ИД" = "Н_ПЛАНЫ"."ИД"
      LEFT JOIN
"Н_ОТДЕЛЫ" ON "Н_ПЛАНЫ"."ОТД_ИД" = "Н_ОТДЕЛЫ"."ИД"
WHERE                                     -- очная и заочная => все
      "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" IS NOT NULL
AND "Н_ОТДЕЛЫ"."КОРОТКОЕ_ИМЯ" = 'КТиУ' -- Программная инженерия нет
AND "Н_УЧЕНИКИ"."ИД" IN (SELECT "ИД"
                           FROM "Н_УЧЕНИКИ"
                           WHERE "ПРИЗНАК" = 'отчисл'
                           AND "КОНЕЦ" > DATE '2012-09-01');
```

7. Сформировать запрос для получения числа на ФКТИУ хорошистов.

```
SELECT COUNT(avg_rating)
FROM (SELECT AVG(CAST("Н_ОЦЕНКИ"."КОД" AS INTEGER)) AS "avg_rating"
      FROM "Н_УЧЕНИКИ"
        JOIN
        "Н_ЛЮДИ" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
        LEFT JOIN
        "Н_ВЕДОМОСТИ" ON "Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
        LEFT JOIN
        "Н_ОЦЕНКИ" ON "Н_ВЕДОМОСТИ"."ОЦЕНКА" = "Н_ОЦЕНКИ"."КОД"
        LEFT JOIN
        "Н_ПЛАНЫ" ON "Н_УЧЕНИКИ"."ПЛАН_ИД" = "Н_ПЛАНЫ"."ИД"
        LEFT JOIN
        "Н_ОТДЕЛЫ" ON "Н_ПЛАНЫ"."ОТД_ИД" = "Н_ОТДЕЛЫ"."ИД"
      WHERE "Н_ОТДЕЛЫ"."КОРОТКОЕ_ИМЯ" = 'КТИУ' -- с ФКТИУ нет результатов
        AND "Н_ОЦЕНКИ"."КОД" IN ('5', '4', '3', '2')
      GROUP BY "Н_ЛЮДИ"."ИД"
      HAVING AVG(CAST("Н_ОЦЕНКИ"."КОД" AS INTEGER)) >= 3.5) as grades;
```

Заключение

В ходе лабораторной работы я освоил написание SQL-запросов, включая JOIN, подзапросы и агрегатные функции. Также научился фильтровать данные и обрабатывать ошибки.