

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики”

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине

“Базы данных”

вариант №17333737312

Выполнил:

студент группы Р3119

Бардин Петр Алексеевич

Преподаватель:

Байрамова Хумай Бахруз Кызы

Содержание

1	Задание	2
2	Ход работы	3
3	Вывод	6

1 Задание

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

Введите вариант:

Внимание! У разных вариантов разный текст задания!

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

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.
Вывести атрибуты: Н_ЛЮДИ.ОТЧЕСТВО, Н_ВЕДОМОСТИ.ДАТА.
Фильтры (AND):
а) Н_ЛЮДИ.ФАМИЛИЯ < Афанасьев.
б) Н_ВЕДОМОСТИ.ИД > 1490007.
с) Н_ВЕДОМОСТИ.ИД < 1457443.
Вид соединения: INNER JOIN.
2. 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.ИД.
Фильтры: (AND)
а) Н_ЛЮДИ.ИМЯ = Ярослав.
б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 113409.
Вид соединения: INNER JOIN.
3. Составить запрос, который ответит на вопрос, есть ли среди студентов группы 3102 люди без ИНН.
4. Выдать различные отчества студентов и число людей с каждой из этих отчеств, ограничив список отчествами, встречающимися более 10 раз на ФКТИУ.
Для реализации использовать соединение таблиц.
5. Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст), где средний возраст равен минимальному возрасту в группе 1100.
6. Получить список студентов, отчисленных до первого сентября 2012 года с очной формы обучения. В результат включить:
номер группы;
номер, фамилию, имя и отчество студента;
номер пункта приказа;
Для реализации использовать подзапрос с IN.
7. Вывести список студентов, имеющих одинаковые фамилии, но не совпадающие даты рождения.

2 Ход работы

Код и отчет по данной лабораторной размещены в системе контроля версий Git на сервисе Github:

https://github.com/BardinPetr/itmo-labs/tree/main/db/year_1/lab_2/readme.md.

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

- Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.
- Вывести атрибуты: Н_ЛЮДИ.ОТЧЕСТВО, Н_ВЕДОМОСТИ.ДАТА.
- Фильтры (AND):
 - Н_ЛЮДИ.ФАМИЛИЯ < Афанасьев.
 - Н_ВЕДОМОСТИ.ИД > 1490007.
 - Н_ВЕДОМОСТИ.ИД < 1457443.
- Вид соединения: INNER JOIN.

```
SELECT human."ОТЧЕСТВО" , ved."ДАТА"  
FROM "Н_ЛЮДИ" as human  
      INNER JOIN "Н_ВЕДОМОСТИ" as ved ON human."ИД" = ved."ЧЛВК_ИД"  
WHERE human."ФАМИЛИЯ" < 'Афанасьев'  
      AND ved."ИД" > 1490007 AND ved."ИД" < 1457443;
```

2. 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.ИД.
- Фильтры (AND):
 - Н_ЛЮДИ.ИМЯ = Ярослав.
 - Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 113409.
 - Вид соединения: INNER JOIN.
- Вид соединения: INNER JOIN.

```
SELECT human."ИД" , learn."НЗК" , stud."ИД"  
FROM "Н_ЛЮДИ" as human  
      INNER JOIN "Н_ОБУЧЕНИЯ" learn on human."ИД" = learn."ЧЛВК_ИД"  
      INNER JOIN "Н_УЧЕНИКИ" stud on human."ИД" = stud."ЧЛВК_ИД"  
WHERE human."ИМЯ" = 'Ярослав'  
      AND learn."ЧЛВК_ИД" = 113409;
```

3. Составить запрос, который ответит на вопрос, есть ли среди студентов группы 3102 люди без ИНН.

```
SELECT count(human."ИНН" IS NULL) > 0 as has_person_without_tax_id
FROM "H_ЛЮДИ" as human
      JOIN "H_УЧЕНИКИ" stud ON human."ИД" = stud."ЧЛВК_ИД"
WHERE stud."ГРУППА" = '3102';
```

4. Выдать различные отчества студентов и число людей с каждой из этих отчеств, ограничив список отчествами, встречающимися более 10 раз на ФКТИУ. Для реализации использовать соединение таблиц.

```
SELECT human."ОТЧЕСТВО" as middlename, count(human."ИД") as count
FROM "H_ЛЮДИ" as human
      JOIN "H_УЧЕНИКИ" stud ON human."ИД" = stud."ЧЛВК_ИД"
      JOIN "H_ПЛАНЫ" plan on stud."ПЛАН_ИД" = plan."ИД"
      JOIN "H_ОТДЕЛЫ" otd on plan."ОТД_ИД" = otd."ИД"
WHERE otd."КОРОТКОЕ_ИМЯ" = 'КТИУ'
      AND human."ОТЧЕСТВО" <> ' . '
GROUP BY human."ОТЧЕСТВО"
HAVING count(human."ИД") > 10
ORDER BY count DESC;
```

5. Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст), где средний возраст равен минимальному возрасту в группе 1100.

```
WITH min_years as
      (SELECT min(trunc(date_part('year', age(human."ДАТА_РОЖДЕНИЯ")))) as min
      FROM "H_ЛЮДИ" as human
            INNER JOIN "H_УЧЕНИКИ" stud
                  ON human."ИД" = stud."ЧЛВК_ИД"
            WHERE stud."ГРУППА" = '1100'
            LIMIT 1)
SELECT stud."ГРУППА" as study_group,
      trunc(avg(date_part('year', age(human."ДАТА_РОЖДЕНИЯ")))) as average_age_y
FROM "H_ЛЮДИ" as human
      JOIN "H_УЧЕНИКИ" stud ON human."ИД" = stud."ЧЛВК_ИД"
GROUP BY study_group
HAVING
      trunc(avg(
            date_part('year', age(human."ДАТА_РОЖДЕНИЯ"))
      )) = (SELECT min FROM min_years);
```

6. Получить список студентов, отчисленных до первого сентября 2012 года с очной формы обучения. Для реализации использовать подзапрос с IN. В результат включить:

- номер группы;
- номер, фамилию, имя и отчество студента;
- номер пункта приказа;

```
SELECT stud."ГРУППА" ,
       human."ИД" ,
       human."ФАМИЛИЯ" ,
       human."ИМЯ" ,
       human."ОТЧЕСТВО" ,
       stud."В_СВЯЗИ_С" as protocol
FROM "Н_ЛЮДИ" human
      JOIN "Н_УЧЕНИКИ" stud on human."ИД" = stud."ЧЛВК_ИД"
WHERE stud."ПЛАН_ИД" IN
      (SELECT "ИД"
       FROM "Н_ПЛАНЫ"
       WHERE "ФО_ИД" = (SELECT "ИД"
                        FROM "Н_ФОРМЫ_ОБУЧЕНИЯ"
                        WHERE "НАИМЕНОВАНИЕ" = 'Очная'))
      AND stud."ПРИЗНАК" = 'отчисл'
      AND stud."КОНЕЦ" < '2012-09-01' ;
```

7. Вывести список студентов, имеющих одинаковые фамилии, но не совпадающие даты рождения.

```
SELECT "ИД" , "ФАМИЛИЯ" , "ДАТА_РОЖДЕНИЯ"
FROM "Н_ЛЮДИ"
WHERE "ФАМИЛИЯ" IN (SELECT "ФАМИЛИЯ"
                     FROM "Н_ЛЮДИ"
                     GROUP BY "ФАМИЛИЯ"
                     HAVING count(DISTINCT "ДАТА_РОЖДЕНИЯ") > 1)
ORDER BY "ФАМИЛИЯ"
```

3 Вывод

В ходе выполнения работы на практике отработано составление запросов к базе данных, включающих в себя объединения таблиц, сортировку, выборку, группировку. Изучены способы последовательного выполнения запросов и использованием результатов предыдущих. Изучены и применены некоторые функции обработки данных и агрегации.