



Вариант №1501  
Лабораторная работа №3  
По дисциплине  
Базы Данных

Выполнил студент группы Р3115:  
Барсуков Максим

Преподаватель:  
Горбунов Михаил Витальевич  
Николаев Владимир Вячеславович

## 1. Текст задания

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

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.ДАТА.  
Фильтры (AND):  
а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.  
б) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 163249.  
Вид соединения: RIGHT 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.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД.  
Фильтры: (AND)  
а) Н\_ЛЮДИ.ИМЯ < Александр.  
б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163276.  
с) Н\_УЧЕНИКИ.ГРУППА = 3100.  
Вид соединения: INNER JOIN.
3. Вывести число фамилий без учета повторений.  
При составлении запроса нельзя использовать DISTINCT.
4. Найти группы, в которых в 2011 году было ровно 10 обучающихся студентов на ФКТИУ.  
Для реализации использовать подзапрос.
5. Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст), где средний возраст меньше среднего возраста в группе 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 EXISTS.
7. Вывести список людей, не являющихся или не являвшихся студентами ФКТИУ (данные, о которых отсутствуют в таблице Н\_УЧЕНИКИ). В запросе нельзя использовать DISTINCT.

## 2. Реализация запросов на SQL

```
-- 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.ДАТА.
-- Фильтры (AND):
-- а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.
-- б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 163249.
-- Вид соединения: RIGHT JOIN.

SELECT "Н_ТИПЫ_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н_ВЕДОМОСТИ"."ДАТА"
FROM "Н_ВЕДОМОСТИ"
  RIGHT JOIN "Н_ТИПЫ_ВЕДОМОСТЕЙ" ON "Н_ВЕДОМОСТИ"."ВЕД_ИД" =
"Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД"
WHERE "Н_ТИПЫ_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" = 'Ведомость'
  AND "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" > 163249;

-----

-- 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.ИМЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ИД.
-- Фильтры: (AND)
-- а) Н_ЛЮДИ.ИМЯ < Александр.
-- б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163276.
-- с) Н_УЧЕНИКИ.ГРУППА = 3100.
-- Вид соединения: INNER JOIN.

SELECT "Н_ЛЮДИ".ИМЯ, "Н_ОБУЧЕНИЯ".ЧЛВК_ИД, "Н_УЧЕНИКИ".ИД
FROM "Н_ЛЮДИ"
  JOIN "Н_ОБУЧЕНИЯ" ON "Н_ОБУЧЕНИЯ".ЧЛВК_ИД = "Н_ЛЮДИ".ИД
  JOIN "Н_УЧЕНИКИ" ON "Н_УЧЕНИКИ".ЧЛВК_ИД = "Н_ЛЮДИ".ИД
WHERE "Н_ЛЮДИ".ИМЯ < 'Александр'
  AND "Н_ОБУЧЕНИЯ".ЧЛВК_ИД > 163276
  AND "Н_УЧЕНИКИ".ГРУППА = '3100';

-----

-- 3.
-- Вывести число фамилий без учета повторений.
-- При составлении запроса нельзя использовать DISTINCT.

SELECT count("ФАМИЛИЯ") FROM
  (SELECT "Н_ЛЮДИ".ФАМИЛИЯ AS "ФАМИЛИЯ"
   FROM "Н_ЛЮДИ"
   GROUP BY "Н_ЛЮДИ".ФАМИЛИЯ)
AS "Число фамилий без повторений";

-- Проверка: SELECT COUNT(DISTINCT "Н_ЛЮДИ".ФАМИЛИЯ) FROM "Н_ЛЮДИ";

-----

-- 4.
-- Найти группы, в которых в 2011 году было ровно 10 обучающихся студентов на ФКТИУ.
-- Для реализации использовать подзапрос.

SELECT "ГРУППЫ_КТИУ_2011".ГРУППА, "ГРУППЫ_КТИУ_2011".КОЛИЧЕСТВО FROM
  (SELECT "Н_УЧЕНИКИ".ГРУППА, count("Н_УЧЕНИКИ".ИД) AS "КОЛИЧЕСТВО" FROM
    "Н_УЧЕНИКИ"
    JOIN "Н_ПЛАНЫ"
      ON "Н_УЧЕНИКИ".ПЛАН_ИД = "Н_ПЛАНЫ".ИД
      AND "Н_ПЛАНЫ".УЧЕВНЫЙ_ГОД = '2010/2011'
    JOIN "Н_ОТДЕЛЫ"
      ON "Н_ОТДЕЛЫ".ИД = "Н_ПЛАНЫ".ОТД_ИД
      AND "Н_ОТДЕЛЫ".КОРОТКОЕ_ИМЯ = 'КТИУ'
    GROUP BY "Н_УЧЕНИКИ".ГРУППА
  ) AS "ГРУППЫ_КТИУ_2011"
WHERE "ГРУППЫ_КТИУ_2011".КОЛИЧЕСТВО = 10;

-----

-- 5.
-- Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний
  возраст),
-- где средний возраст меньше 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'
  GROUP BY "Н_УЧЕНИКИ".ГРУППА
);

-----

-- 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 EXISTS.

SELECT "ВНЕШ_УЧЕНИКИ"."ГРУППА",
       "ВНЕШ_УЧЕНИКИ"."ИД",
       "Н_ЛЮДИ"."ФАМИЛИЯ",
       "Н_ЛЮДИ"."ИМЯ",
       "Н_ЛЮДИ"."ОТЧЕСТВО",
       "ВНЕШ_УЧЕНИКИ"."П_ПРКОК_ИД"
FROM "Н_УЧЕНИКИ" "ВНЕШ_УЧЕНИКИ"
JOIN "Н_ЛЮДИ" ON "Н_ЛЮДИ"."ИД" = "ВНЕШ_УЧЕНИКИ"."ЧЛВК_ИД"
JOIN "Н_ПЛАНЫ" ON "ВНЕШ_УЧЕНИКИ"."ПЛАН_ИД" = "Н_ПЛАНЫ"."ИД"
JOIN "Н_ФОРМЫ_ОБУЧЕНИЯ" ON "Н_ПЛАНЫ"."ФО_ИД" = "Н_ФОРМЫ_ОБУЧЕНИЯ"."ИД"
AND ("Н_ФОРМЫ_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Заочная' OR
"Н_ФОРМЫ_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Очная')

JOIN "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ" ON "Н_ПЛАНЫ"."НАПС_ИД" = "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."ИД"
JOIN "Н_НАПР_СПЕЦ" ON "Н_НАПР_СПЕЦ"."ИД" = "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."НС_ИД"
AND "Н_НАПР_СПЕЦ"."НАИМЕНОВАНИЕ" = 'Программная инженерия'

WHERE EXISTS (
  SELECT *
  FROM "Н_УЧЕНИКИ" "ВНУТР_УЧЕНИКИ"
  WHERE "ВНУТР_УЧЕНИКИ"."ПРИЗНАК" = 'отчисл'
  AND "ВНУТР_УЧЕНИКИ"."СОСТОЯНИЕ" = 'утвержден'
  AND "ВНУТР_УЧЕНИКИ"."ИД" = "ВНЕШ_УЧЕНИКИ"."ИД"
  AND DATE("ВНУТР_УЧЕНИКИ"."КОНЕЦ") < '2012-09-01'
);

-----

-- 7.
-- Вывести список людей, не являющихся или не являвшихся студентами ФКТИУ
-- (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_ОТДЕЛЫ"."ИД"
  AND "Н_ОТДЕЛЫ"."КОРОТКОЕ_ИМЯ" = 'КТИУ'
  JOIN "Н_ЛЮДИ" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
  WHERE "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "people"."ИД"
);

```

### 3. Вывод

При выполнении лабораторной работы я познакомился с основными функциями языка SQL и диалекта PostgreSQL. Научился писать запросы, получать, агрегировать, отсеивать и сортировать полученные данные с использованием различных синтаксических конструкций языка. В результате был освоен язык DML SQL, предназначенный для работы с данными, хранящимися внутри базы данных.