



Вариант №785461
Лабораторная работа №2
По дисциплине
Базы Данных

Выполнил студент группы Р3117:
Изаак Герман Константинович

Преподаватель:
Чупанов Аликылыч Алибекович

1. Текст задания

Для выполнения лабораторной работы №2 необходимо:

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.ЧЛВК_ИД.
Фильтры (AND):
а) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = 1.
б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 117219.
с) Н_ВЕДОМОСТИ.ЧЛВК_ИД = 117219.
Вид соединения: INNER 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.ИД, Н_ВЕДОМОСТИ.ЧЛВК_ИД,
Н_СЕССИЯ.ДАТА.
Фильтры (AND):
а) Н_ЛЮДИ.ОТЧЕСТВО = Георгиевич.
б) Н_ВЕДОМОСТИ.ИД > 1426978.
Вид соединения: LEFT JOIN.
3. Составить запрос, который ответит на вопрос, есть ли среди студентов ФКТИУ те, кто старше 25 лет.
4. Выдать различные имена преподавателей и число людей с каждой из этих имен, ограничив список именами, встречающимися ровно 10 раз на заочной форме обучения.
Для реализации использовать подзапрос.
5. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср_оценка), у которых средняя оценка больше средней оценк(е|и) в группе 3100.
6. Получить список студентов, отчисленных до первого сентября 2012 года с заочной формы обучения (специальность: 230101). В результат включить:
номер группы;
номер, фамилию, имя и отчество студента;
номер пункта приказа;
Для реализации использовать подзапрос с EXISTS.
7. Вывести список студентов, имеющих одинаковые отчества, но не совпадающие даты рождения.

2. Реализация запросов на SQL

```
--First request
SELECT "Н ТИПЫ ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
FROM "Н ТИПЫ ВЕДОМОСТЕЙ"
INNER JOIN "Н_ВЕДОМОСТИ" ON "Н ТИПЫ_ВЕДОМОСТЕЙ"."ИД" = "Н_ВЕДОМОСТИ"."ВЕД_ИД"
WHERE "Н ТИПЫ_ВЕДОМОСТЕЙ"."ИД" = 1
AND "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" > 117219
```

```

AND "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = 117219;

--Second request
SELECT "Н_ЛЮДИ"."ИД", "Н_ВЕДОМОСТИ"."ЧЛВК_ИД", "Н_СЕССИЯ"."ДАТА"
FROM "Н_ЛЮДИ"
LEFT JOIN "Н_ВЕДОМОСТИ" ON "Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
LEFT JOIN "Н_СЕССИЯ" ON "Н_ЛЮДИ"."ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД"
WHERE "Н_ЛЮДИ"."ОТЧЕСТВО" = 'Георгиевич'
AND "Н_ВЕДОМОСТИ"."ИД" > 1426978;

--Third request
SELECT
    CASE
        WHEN COUNT(*) > 0 THEN 'Да, существуют'
        ELSE 'Нет, не существуют'
    END AS "Есть ли студенты ФКТИУ старше 25 лет"
FROM
    "Н_ЛЮДИ"
INNER JOIN
    "Н_ОБУЧЕНИЯ"
    ON "Н_ЛЮДИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД"
INNER JOIN
    "Н_УЧЕНИКИ"
    ON "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
INNER JOIN
    "Н_ПЛАНЫ"
    ON "Н_УЧЕНИКИ"."ПЛАН_ИД" = "Н_ПЛАНЫ"."ИД"
INNER JOIN
    "Н_ОТДЕЛЫ"
    ON "Н_ПЛАНЫ"."ОТД_ИД" = "Н_ОТДЕЛЫ"."ИД"
WHERE
    "Н_ОТДЕЛЫ"."КОРОТКОЕ_ИМЯ" = 'КТИУ'
    AND EXTRACT(YEAR FROM AGE("Н_ЛЮДИ"."ДАТА_РОЖДЕНИЯ")) > 25;

--Fourth request

-- Имена преподавателей (люди не из Н_УЧЕНИКИ)
WITH Преподаватели AS (
    SELECT "ИМЯ"
    FROM "Н_ЛЮДИ"
    WHERE "ИД" NOT IN (SELECT "ЧЛВК_ИД" FROM "Н_УЧЕНИКИ")
),

-- Имена учеников на заочной форме (ровно 10 повторов)
УченикиЗаочной AS (
    SELECT "Н_ЛЮДИ"."ИМЯ"
    FROM "Н_ЛЮДИ"
    JOIN "Н_УЧЕНИКИ" ON "Н_ЛЮДИ"."ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
    JOIN "Н_ПЛАНЫ" ON "Н_УЧЕНИКИ"."ПЛАН_ИД" = "Н_ПЛАНЫ"."ИД"
    JOIN "Н_ФОРМЫ_ОБУЧЕНИЯ" ON "Н_ПЛАНЫ"."ФО_ИД" = "Н_ФОРМЫ_ОБУЧЕНИЯ"."ИД"
    WHERE "Н_ФОРМЫ_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Заочная'
    GROUP BY "Н_ЛЮДИ"."ИМЯ"
    HAVING COUNT(*) = 10
),

-- Имена преподавателей, которые встречаются ровно 10 раз у заочников
ПодходящиеИмена AS (
    SELECT "ИМЯ"
    FROM Преподаватели
    WHERE "ИМЯ" IN (SELECT "ИМЯ" FROM УченикиЗаочной)
)

-- Итоговый подсчет всех людей с подходящими именами
SELECT

```

```

        "ИМЯ" AS "Имя преподавателя",
        COUNT(*) AS "Общее количество людей с этим именем"
FROM
    "Н_ЛЮДИ"
WHERE
    "ИМЯ" IN (SELECT "ИМЯ" FROM ПодходящиеИмена)
GROUP BY
    "ИМЯ";

--Fifth request
WITH СредняяОценка3100 AS (
    SELECT
        AVG("ОЦЕНКА"::numeric) AS Средняя
    FROM
        "Н_ВЕДОМОСТИ"
    INNER JOIN "Н_УЧЕНИКИ"
        ON "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
    INNER JOIN "Н_ГРУППЫ_ПЛАНОВ"
        ON "Н_УЧЕНИКИ"."ГРУППА" = "Н_ГРУППЫ_ПЛАНОВ"."ГРУППА"
    WHERE
        "Н_ГРУППЫ_ПЛАНОВ"."ГРУППА" = '3100'
        AND "Н_ВЕДОМОСТИ"."ОЦЕНКА" ~ '^[2-5]$'
),

Студенты4100 AS (
    SELECT
        "Н_УЧЕНИКИ"."ЧЛВК_ИД",
        "Н_ЛЮДИ"."ФАМИЛИЯ",
        "Н_ЛЮДИ"."ИМЯ",
        "Н_ЛЮДИ"."ОТЧЕСТВО",
        "Н_ВЕДОМОСТИ"."ОЦЕНКА"::numeric AS Оценка
    FROM
        "Н_УЧЕНИКИ"
    INNER JOIN "Н_ГРУППЫ_ПЛАНОВ"
        ON "Н_УЧЕНИКИ"."ГРУППА" = "Н_ГРУППЫ_ПЛАНОВ"."ГРУППА"
    INNER JOIN "Н_ЛЮДИ"
        ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
    LEFT JOIN "Н_ВЕДОМОСТИ"
        ON "Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
    WHERE
        "Н_ГРУППЫ_ПЛАНОВ"."ГРУППА" = '4100'
        AND "Н_ВЕДОМОСТИ"."ОЦЕНКА" ~ '^[2-5]$'
)

SELECT
    Студенты4100."ЧЛВК_ИД" AS "Номер",
    Студенты4100."ФАМИЛИЯ" AS "Фамилия",
    Студенты4100."ИМЯ" AS "Имя",
    Студенты4100."ОТЧЕСТВО" AS "Отчество",
    AVG(Студенты4100.Оценка) AS "Ср_оценка"
FROM
    Студенты4100
CROSS JOIN СредняяОценка3100
GROUP BY
    Студенты4100."ЧЛВК_ИД",
    Студенты4100."ФАМИЛИЯ",
    Студенты4100."ИМЯ",
    Студенты4100."ОТЧЕСТВО"
HAVING
    AVG(Студенты4100.Оценка) > (SELECT Средняя FROM СредняяОценка3100);

--Sixth request
SELECT
    "Н_УЧЕНИКИ"."ГРУППА" AS "Номер группы",

```

```

        "Н_ЛЮДИ"."ФАМИЛИЯ" AS "Фамилия",
        "Н_ЛЮДИ"."ИМЯ" AS "Имя",
        "Н_ЛЮДИ"."ОТЧЕСТВО" AS "Отчество",
        "Н_УЧЕНИКИ"."КОНЕЦ_ПО_ПРИКАЗУ" AS "Номер пункта приказа"
FROM
    "Н_УЧЕНИКИ"
INNER JOIN
    "Н_ЛЮДИ" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
WHERE
    "Н_УЧЕНИКИ"."КОНЕЦ_ПО_ПРИКАЗУ" < TIMESTAMP '2012-09-01 00:00:00'
    AND EXISTS (
        SELECT 1
        FROM
            "Н_ПЛАНЫ"
        INNER JOIN
            "Н_ФОРМЫ_ОБУЧЕНИЯ" ON "Н_ПЛАНЫ"."ФО_ИД" = "Н_ФОРМЫ_ОБУЧЕНИЯ"."ИД"
        INNER JOIN
            "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ" ON "Н_ПЛАНЫ"."НАПС_ИД" =
"Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."ИД"
        INNER JOIN
            "Н_НАПР_СПЕЦ" ON "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."НС_ИД" =
"Н_НАПР_СПЕЦ"."ИД"
        WHERE
            "Н_ПЛАНЫ"."ИД" = "Н_УЧЕНИКИ"."ПЛАН_ИД"
            AND "Н_ФОРМЫ_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Заочная'
            AND "Н_НАПР_СПЕЦ"."КОД_НАПРСПЕЦ" = '230101'
    );

--Seventh request
SELECT
    "Н_ЛЮДИ"."ФАМИЛИЯ",
    "Н_ЛЮДИ"."ИМЯ",
    "Н_ЛЮДИ"."ОТЧЕСТВО",
    "Н_ЛЮДИ"."ДАТА_РОЖДЕНИЯ"
FROM
    "Н_ЛЮДИ"
WHERE
    EXISTS (
        SELECT 1
        FROM "Н_ЛЮДИ" AS "Н_ЛЮДИ_2"
        WHERE
            "Н_ЛЮДИ"."ОТЧЕСТВО" = "Н_ЛЮДИ_2"."ОТЧЕСТВО"          -- Одинаковое
            AND "Н_ЛЮДИ"."ДАТА_РОЖДЕНИЯ" <> "Н_ЛЮДИ_2"."ДАТА_РОЖДЕНИЯ" --
Разные даты
    )
    AND "Н_ЛЮДИ"."ОТЧЕСТВО" IS NOT NULL
ORDER BY
    "Н_ЛЮДИ"."ОТЧЕСТВО",
    "Н_ЛЮДИ"."ДАТА_РОЖДЕНИЯ";

```

3. Вывод по работе

При выполнении лабораторной работы я познакомился с основными операторами PostgreSQL. Научился использовать различные типы соединений таблиц, использовать подзапросы, фильтровать, группировать и сортировать данные.