

Университет ИТМО

Лабораторная работа №3

«Базы данных»

Студент: **Фарзекаев Артур Робертович**

Кафедра: **ВТ**

Дисциплина: **Базы Данных**

Факультет: **ПИиКТ**

Группа: **Р3100**

Вариант: **961**

Преподаватель: **Николаев Владимир**

Вячеславович.

г. Санкт-Петербург.

2018 г.

Задание.

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.ИД, Н_ВЕДОМОСТИ.ЧЛВК_ИД.

Фильтры (AND):

а) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = 3.

б) Н_ВЕДОМОСТИ.ДАТА > 1998-01-05.

с) Н_ВЕДОМОСТИ.ДАТА < 1998-01-05.

Вид соединения: INNER JOIN.

```
ucheb=> SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД, Н_ВЕДОМОСТИ.ЧЛВК_ИД FROM Н_ТИПЫ_ВЕДОМОСТЕЙ INNER JOIN Н_ВЕДОМОСТИ ON
(Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = Н_ВЕДОМОСТИ.ТВ_ИД) WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = 3 AND
Н_ВЕДОМОСТИ.ДАТА BETWEEN '1998-01-05' AND '1998-01-05' LIMIT 5;
ИД | ЧЛВК_ИД
-----+-----
(0 rows)
```

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.ИД, Н_СЕССИЯ.ЧЛВК_ИД.

Фильтры (AND):

а) Н_ЛЮДИ.ИД = 163484.

б) Н_ВЕДОМОСТИ.ИД > 1426978.

Вид соединения: RIGHT JOIN.

```
ucheb=> SELECT Н_ЛЮДИ.ИД, Н_ВЕДОМОСТИ.ИД, Н_СЕССИЯ.ЧЛВК_ИД FROM Н_ЛЮДИ RIGHT JOIN Н_ВЕДОМОСТИ ON
ucheb-> (Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД) RIGHT JOIN Н_СЕССИЯ ON (Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД)
ucheb-> WHERE Н_ЛЮДИ.ИД = '163484' AND Н_ВЕДОМОСТИ.ИД > '1426978';
ИД | ИД | ЧЛВК_ИД
-----+-----
(0 rows)
```

3. Вывести число студентов ФКТИУ, которые младше 20 лет.

Ответ должен содержать только одно число.

```
ucheb=> SELECT COUNT(ЧЛВК_ИД) FROM Н_УЧЕНИКИ JOIN Н_ЛЮДИ ON (Н_УЧЕНИКИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД)
ucheb-> JOIN Н_ПЛАНЫ on Н_УЧЕНИКИ.ПЛАН_ИД=Н_ПЛАНЫ.ИД join Н_ОТДЕЛЫ on
ucheb-> (Н_ОТДЕЛЫ.ИД = Н_ПЛАНЫ.ОТД_ИД) where КОРОТКОЕ_ИМЯ = 'КТИУ'
ucheb-> AND date_part('year',age(Н_ЛЮДИ.ДАТА_РОЖДЕНИЯ::date)) < 20;
count
-----
3000
(1 row)
```

4. В таблице Н_ГРУППЫ_ПЛАНОВ найти номера планов, по которым обучается (обучалось) ровно 2 групп на очной форме обучения.
Для реализации использовать подзапрос.

```

ucheb=> SELECT ПЛАН_ИД FROM Н_ГРУППЫ_ПЛАНОВ WHERE ПЛАН_ИД IN
ucheb-> (SELECT ПЛАН_ИД FROM Н_ГРУППЫ_ПЛАНОВ GROUP BY ПЛАН_ИД HAVING COUNT(*) = 2)
ucheb-> AND ПЛАН_ИД IN (SELECT Н_ГРУППЫ_ПЛАНОВ.ПЛАН_ИД FROM Н_ГРУППЫ_ПЛАНОВ INNER JOIN Н_ПЛАНЫ ON
ucheb(> (Н_ГРУППЫ_ПЛАНОВ.ПЛАН_ИД = Н_ПЛАНЫ.ИД) INNER JOIN Н_ФОРМЫ_ОБУЧЕНИЯ ON
ucheb-> (Н_ПЛАНЫ.ФО_ИД = Н_ФОРМЫ_ОБУЧЕНИЯ.ИД) where Н_ФОРМЫ_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Очная')
ucheb-> GROUP BY ПЛАН_ИД ORDER BY ПЛАН_ИД;
  ПЛАН_ИД
-----
         25
         29
         32
         33
--More--

```

5. Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст), где средний возраст меньше среднего возраста в группе 1100.

```

ucheb=> SELECT "Н_УЧЕНИКИ"."ГРУППА", ROUND(AVG(date_part('year',age(Н_ЛЮДИ.ДАТА_РОЖДЕНИЯ::date))))
ucheb-> AS "СРЕДНИЙ_ВОЗРАСТ" FROM "Н_УЧЕНИКИ" JOIN "Н_ЛЮДИ" ON ("Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
ucheb-> GROUP BY ("Н_УЧЕНИКИ"."ГРУППА") HAVING (AVG(date_part('year',age(Н_ЛЮДИ.ДАТА_РОЖДЕНИЯ::date)))) <
ucheb-> (select ROUND(AVG(date_part('year',age(Н_ЛЮДИ.ДАТА_РОЖДЕНИЯ::date))))
ucheb(> FROM "Н_УЧЕНИКИ" JOIN "Н_ЛЮДИ" ON ("Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД") WHERE ("Н_УЧЕНИКИ"."ГРУППА" = '1100'));
  ГРУППА | СРЕДНИЙ_ВОЗРАСТ
-----+-----
2538    |                29
254     |                24
3108    |                26
351     |                23
1100    |                29
3125    |                26
2539    |                28
4895    |                21
--More--

```

6. Получить список студентов, отчисленных до первого сентября 2012 года с заочной формы обучения. В результат включить:

номер группы;

номер, фамилию, имя и отчество студента;

номер пункта приказа;

Для реализации использовать подзапрос с EXISTS.

```

ucheb=> SELECT Н_УЧЕНИКИ.ГРУППА, Н_ЛЮДИ.ИД, Н_ЛЮДИ.ФАМИЛИЯ, Н_ЛЮДИ.ИМЯ, Н_ЛЮДИ.ОТЧЕСТВО,
ucheb-> Н_ВЕДОМОСТИ.НОМЕР_ДОКУМЕНТА FROM Н_УЧЕНИКИ JOIN Н_ЛЮДИ ON (Н_ЛЮДИ.ИД=Н_УЧЕНИКИ.ЧЛВК_ИД)
ucheb-> JOIN Н_ОБУЧЕНИЯ USING (ЧЛВК_ИД) JOIN Н_ПЛАНЫ ON (Н_ПЛАНЫ.ИД = Н_УЧЕНИКИ.ПЛАН_ИД) JOIN Н_ВЕДОМОСТИ ON
ucheb-> (Н_ЛЮДИ.ИД=Н_ВЕДОМОСТИ.ЧЛВК_ИД) JOIN Н_ФОРМЫ_ОБУЧЕНИЯ ON (Н_ПЛАНЫ.ФО_ИД=Н_ФОРМЫ_ОБУЧЕНИЯ.ИД)
ucheb-> JOIN Н_НАПР_СПЕЦ ON (Н_НАПР_СПЕЦ.ИД=Н_ПЛАНЫ.НАПС_ИД) WHERE EXISTS(SELECT КОНЕЦ FROM Н_УЧЕНИКИ WHERE
ucheb-> КОНЕЦ < '2012-09-01') AND EXISTS(SELECT НАИМЕНОВАНИЕ FROM Н_ФОРМЫ_ОБУЧЕНИЯ WHERE НАИМЕНОВАНИЕ = 'заочная');
  ГРУППА | ИД | ФАМИЛИЯ | ИМЯ | ОТЧЕСТВО | НОМЕР_ДОКУМЕНТА
-----+-----
(0 rows)

```

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

```
ucheb=> SELECT Н_люди.ФАМИЛИЯ, Н_люди.ИМЯ, Н_люди.ОТЧЕСТВО FROM Н_люди WHERE ФАМИЛИЯ IN (SELECT Н_люди.ФАМИЛИЯ FROM Н_люди
WHERE EXISTS (SELECT ФАМИЛИЯ FROM Н_ученики WHERE (ФАМИЛИЯ = Н_люди.ФАМИЛИЯ)) GROUP BY ФАМИЛИЯ HAVING COUNT (ФАМИЛИЯ) >=2)
AND Н_люди.ИД IN (SELECT DISTINCT ИД FROM Н_люди) ORDER BY Н_люди.ФАМИЛИЯ;
  ФАМИЛИЯ      |      ИМЯ      |      ОТЧЕСТВО
-----+-----+-----
Абрамов         | Владислав     | Александрович
Абрамов         | Роман         | Анатольевич
Абрамов         | Михаил        | Олегович
Абрамов         | Максим        | Александрович
Абрамов         | Владимир      | Андреевич
Абрамов         | Иван          | Владимирович
Абрамов         | Марк          | Александрович
Авдеев          | Александр     | Михайлович
Авдеев          | Сергей        | Владимирович
--More--
```

Вывод:

В ходе выполнения данной лабораторной работы я составил и выполнил запросы с использованием подзапросов, соединений, таких как INNER JOIN, RIGHT JOIN.