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 по

Базам данных №2

Вариант 3

Работу выполнил:

Касьяненко В.М.

Группа:

P3120

Преподаватель:

Королёва Ю.А.

Санкт-Петербург,

2023

Текст задания

Лабораторная работа #2

Задание.

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

Команда для подключения к базе данных ucheb:

psql -h pg -d ucheb

Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Реализацию запросов на SQL.
3. Выводы по работе.

Темы для подготовки к защите лабораторной работы:

1. SQL
2. Соединение таблиц
3. Подзапросы
4. Представления
5. Последовательности

Запросы

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
   Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.
   Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ИД.
   Фильтры (AND):
   a) Н_ЛЮДИ.ФАМИЛИЯ < Иванов.
   b) Н_ВЕДОМОСТИ.ИД = 1250981.
   c) Н_ВЕДОМОСТИ.ИД = 1250972.
   Вид соединения: RIGHT JOIN.

```
SELECT "Н_ЛЮДИ"."ИМЯ","Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ВЕДОМОСТИ"."ИД"
FROM "Н_ЛЮДИ"
RIGHT JOIN "Н_ВЕДОМОСТИ" ON "Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
WHERE "Н_ЛЮДИ"."ФАМИЛИЯ" < 'Иванов'
AND "Н_ВЕДОМОСТИ"."ИД" IN (1250981, 1250972);
```

Output | Result 1

0 rows

"ИМЯ"	"ФАМИЛИЯ"	"ИД"

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.ДАТА, Н_СЕССИЯ.ИД.
   Фильтры (AND):
   a) Н_ЛЮДИ.ИД < 100012.
   b) Н_ВЕДОМОСТИ.ИД = 1457443.
   Вид соединения: RIGHT JOIN.

```
SELECT "Н_ЛЮДИ"."ОТЧЕСТВО", "Н_ВЕДОМОСТИ"."ДАТА", "Н_СЕССИЯ"."ИД"
FROM "Н_ЛЮДИ"
RIGHT JOIN "Н_ВЕДОМОСТИ" ON "Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
RIGHT JOIN "Н_СЕССИЯ" ON "Н_ВЕДОМОСТИ"."СЭС_ИД" = "Н_СЕССИЯ"."СЭС_ИД"
WHERE "Н_ЛЮДИ"."ИД" < 100012
AND "Н_ВЕДОМОСТИ"."ИД" = 1457443
```

Output | Result 2

0 rows

"ОТЧЕСТВО"	"ДАТА"	"ИД"

3. Вывести число студентов ФКТИУ, которые старше 25 лет.
   Ответ должен содержать только одно число.

```
SELECT COUNT("Н_УЧЕНИКИ"."ИД")
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"
JOIN "Н_ЛЮДИ" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
AND EXTRACT(YEAR FROM age("Н_ЛЮДИ"."ДАТА_РОЖДЕНИЯ")) > 25
AND "Н_ОТДЕЛЫ"."КОРОТКОЕ_ИМЯ" = 'КТиУ'
```

Output | COUNT("Н_УЧЕНИКИ"."ИД"):bigint

1 row

	count
1	15271

4. В таблице Н_ГРУППЫ_ПЛАНОВ найти номера планов, по которым обучается (обучалось) более 2 групп на заочной форме обучения.
   Для реализации использовать соединение таблиц.

```sql
SELECT "Н_ГРУППЫ_ПЛАНОВ"."ПЛАН_ИД", count("Н_ГРУППЫ_ПЛАНОВ"."ГРУППА")
FROM "Н_ГРУППЫ_ПЛАНОВ" JOIN "Н_ПЛАНЫ" ON "Н_ПЛАНЫ"."ИД" = "Н_ГРУППЫ_ПЛАНОВ"."ПЛАН_ИД"
    JOIN "Н_ФОРМЫ_ОБУЧЕНИЯ" ON "Н_ПЛАНЫ"."ФО_ИД" = "Н_ФОРМЫ_ОБУЧЕНИЯ"."ИД"
WHERE "Н_ФОРМЫ_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Заочная'
GROUP BY "Н_ГРУППЫ_ПЛАНОВ"."ПЛАН_ИД"
HAVING count("Н_ГРУППЫ_ПЛАНОВ"."ГРУППА") > 2;
```

Output | Result 4

0 rows

ПЛАН_ИД	count

5. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср_оценка), у которых средняя оценка меньше минимальной оценк(е|и) в группе 3100.

```sql
SELECT "Н_ЛЮДИ"."ИД", "ФАМИЛИЯ", "ИМЯ", "ОТЧЕСТВО", AVG(CAST("ОЦЕНКА" AS NUMERIC)) AS "СРЕДНЯЯ_ОЦЕНКА"
FROM "Н_ЛЮДИ"
JOIN "Н_УЧЕНИКИ" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД" AND "Н_УЧЕНИКИ"."ГРУППА" = '4100'
JOIN "Н_ВЕДОМОСТИ"
ON "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД" AND "ОЦЕНКА" NOT IN ('осв', 'неявка', 'зачет', 'незач')
GROUP BY "Н_ЛЮДИ"."ИД", "ФАМИЛИЯ", "ИМЯ", "ОТЧЕСТВО"
HAVING AVG(CAST ("ОЦЕНКА" AS NUMERIC)) < (SELECT MIN (EXP.MARK)
FROM (
SELECT AVG(CAST ("ОЦЕНКА" AS NUMERIC)) AS MARK
FROM "Н_УЧЕНИКИ"
JOIN "Н_ВЕДОМОСТИ" ON "Н_УЧЕНИКИ"."ГРУППА" = '3100'
AND "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
AND "ОЦЕНКА" NOT IN('осв', 'неявка', 'зачет', 'незач')
) EXP);
```

Output | Result 5

108 rows

	"ИД"	"ФАМИЛИЯ"	"ИМЯ"	"ОТЧЕСТВО"	"СРЕДНЯЯ_ОЦЕНКА"
1	121861	Земцов	Андрей	Николаевич	3.4666666666666667
2	121908	Андрианов	Олег	Борисович	3.8545454545454545
3	137822	Каширин	Иван	Сергеевич	3.5714285714285714
4	149561	Щеглов	Константин	Андреевич	2.9024390243902439
5	119275	Каракулев	Александр	Дмитриевич	3.8045977011494253
6	120222	Абиева	Людмила	Ибрагимовна	3.339622641509434
7	120225	Валеев	Айрат	Фаритович	3.4363636363636364
8	142219	Парамонов	Иван	Витальевич	3.6875

6. Получить список студентов, отчисленных до первого сентября 2012 года с очной формы обучения (специальность: 230101). В результат включить:
номер группы;
номер, фамилию, имя и отчество студента;
номер пункта приказа;
Для реализации использовать соединение таблиц.

```sql
SELECT "Н_УЧЕНИКИ"."ГРУППА", "Н_УЧЕНИКИ"."ИД", "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ЛЮДИ"."ИМЯ", "Н_ЛЮДИ"."ОТЧЕСТВО",
    "Н_УЧЕНИКИ"."П_ПРКОК_ИД"
FROM "Н_УЧЕНИКИ"
JOIN "Н_ЛЮДИ" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
JOIN "Н_ПЛАНЫ" ON "Н_УЧЕНИКИ"."ПЛАН_ИД" = "Н_ПЛАНЫ"."ИД"
JOIN "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ" ON "Н_ПЛАНЫ"."НАПС_ИД" = "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."ИД"
JOIN "Н_НАПР_СПЕЦ" ON "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."НС_ИД" = "Н_НАПР_СПЕЦ"."ИД"
WHERE "Н_УЧЕНИКИ"."КОНЕЦ" < '2012-09-01'
AND "Н_УЧЕНИКИ"."ПРИЗНАК" = 'отчисл'
```

Output | Result 6

53 rows

	"ГРУППА"	"ИД"	"ФАМИЛИЯ"	"ИМЯ"	"ОТЧЕСТВО"	"П_ПРКОК_ИД"
1	3100	230358	Корепанов	Андрей	Анатольевич	706513
2	2101	225936	Салихова	Чулпан	Радиковна	596755
3	2101	225516	Кошева	Карина	Руслановна	595278
4	2101	222211	Канцемал	Кирилл	Алексеевич	572910
5	4100	222751	Лагунов	Михаил	Андреевич	587814
6	4101	176693	Дмитриев	Алексей	Геннадьевич	465608
7	1100	176653	Мягков	Илья	Игоревич	465628
8	1101	183111	Шахов	Денис	Валерьевич	480726
9	3895	107104	Царев	Дмитрий	Викторович	265098

7. Вывести список людей, не являющихся или не являвшихся студентами СПбГУ ИТМО (данные, о которых отсутствуют в таблице Н_УЧЕНИКИ). В запросе нельзя использовать DISTINCT.

```sql
SELECT "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ЛЮДИ"."ИМЯ", "Н_ЛЮДИ"."ОТЧЕСТВО"
FROM "Н_ЛЮДИ"
LEFT JOIN "Н_УЧЕНИКИ" ON "Н_ЛЮДИ"."ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
WHERE "Н_УЧЕНИКИ"."ЧЛВК_ИД" IS NULL;
```

Output | ucheb.public.Н_ЛЮДИ

162 rows | Tx: Auto | DDL

	"ФАМИЛИЯ"	"ИМЯ"	"ОТЧЕСТВО"
1	Парфенов	Петр	Сергеевич
2	Аникина	Алла	Филипповна
3	Лифшиц	Анатолий	Львович
4	Денисова	Эльвира	Викторовна
5	Звягин	Виктор	Фомич
6	Блохин	Валентин	Николаевич
7	Новолодская	Татьяна	Алексеевна
8	Беспалов	Виктор	Георгиевич
9	Ищенко	Алексей	Петрович

Дополнительные задания

Задание 1: вывести номер факультета, его наименование, количество студентов бакалавриата и магистров на нем (при помощи маски) и квалификацию факультета.

```
SELECT
"Н_ПЛАНЫ"."ОТД_ИД" AS "Номер факультета",
"Н_ОТДЕЛЫ"."ИМЯ_В_ИМИН_ПАДЕЖЕ" AS "Наименование факультета",
COUNT(CASE WHEN "Н_КВАЛИФИКАЦИИ"."НАИМЕНОВАНИЕ" LIKE 'Бакалавр%' THEN 1 END) AS "Количество бакалавров",
COUNT(CASE WHEN "Н_КВАЛИФИКАЦИИ"."НАИМЕНОВАНИЕ" LIKE 'Магистр%' THEN 1 END) AS "Количество магистров",
"Н_КВАЛИФИКАЦИИ"."НАИМЕНОВАНИЕ" AS "Квалификация факультета"
FROM "Н_ПЛАНЫ"
JOIN "Н_ОТДЕЛЫ" ON "Н_ПЛАНЫ"."ОТД_ИД" = "Н_ОТДЕЛЫ"."ИД"
JOIN "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ" ON "Н_ПЛАНЫ"."НАПС_ИД" = "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."ИД"
JOIN "Н_КВАЛИФИКАЦИИ" ON "Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ"."КВАЛ_ИД" = "Н_КВАЛИФИКАЦИИ"."ИД"
GROUP BY
"Н_ПЛАНЫ"."ОТД_ИД",
"Н_ОТДЕЛЫ"."ИМЯ_В_ИМИН_ПАДЕЖЕ",
"Н_КВАЛИФИКАЦИИ"."НАИМЕНОВАНИЕ"
HAVING
COUNT(CASE WHEN "Н_КВАЛИФИКАЦИИ"."НАИМЕНОВАНИЕ" LIKE 'Бакалавр%' THEN 1 END) +
COUNT(CASE WHEN "Н_КВАЛИФИКАЦИИ"."НАИМЕНОВАНИЕ" LIKE 'Магистр%' THEN 1 END) > 0;
```

Output | Result 1

11 rows

"Номер...	"Наименование факультета"	"Количество бак...	"Количество маги...	"Квалификация факультета"
703	факультет компьютерных технологий и…	0	4	Магистр математики
717	факультет информационных технологий…	0	6	Магистр математики
717	факультет информационных технологий…	29	0	Бакалавр прикладной математик

Задание 2: вывести все группы КТУ, а также количество отличников и двоечников в них.

```
SELECT "Н_УЧЕНИКИ"."ГРУППА",
       COUNT(CASE WHEN "Н_ВЕДОМОСТИ"."ОЦЕНКА" = '5' THEN 1 END) as "Отличники",
       COUNT(CASE WHEN "Н_ВЕДОМОСТИ"."ОЦЕНКА" = '2' THEN 1 END) as "Двоечники"
FROM "Н_УЧЕНИКИ"
JOIN "Н_ПЛАНЫ" ON "Н_УЧЕНИКИ"."ПЛАН_ИД" = "Н_ПЛАНЫ"."ПЛАН_ИД"
JOIN "Н_ОТДЕЛЫ" ON "Н_ПЛАНЫ"."ОТД_ИД" = "Н_ОТДЕЛЫ"."ИД"
JOIN "Н_ВЕДОМОСТИ" ON "Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
WHERE "Н_ОТДЕЛЫ"."КОРОТКОЕ_ИМЯ" = 'КТиУ'
GROUP BY "Н_УЧЕНИКИ"."ГРУППА";
```

Output | Result 2

100 rows

	"ГРУППА"	"Отличники"	"Двоечники"
1	354	2924	39
2	2508	2199	459
3	5509	189	6
4	551	888	6
5	5114	279	6

Задание 3: определить список студентов, имеющих наибольший средний интервал между последовательными датами сдачи экзамена.

```
SELECT "Н_ЛЮДИ"."ИД", AVG(EXTRACT(DAY FROM "interval") + 30 * DATE_PART('month', "interval") +
                          365 * DATE_PART('year', "interval")) as "avg_interval",
       ARRAY_AGG("Н_СЕССИЯ"."ДАТА" ORDER BY "Н_СЕССИЯ"."ДАТА") as "exam_dates"
FROM (
    SELECT
        "Н_СЕССИЯ"."ЧЛВК_ИД",
        "Н_СЕССИЯ"."ДАТА",
        LAG("Н_СЕССИЯ"."ДАТА") OVER (PARTITION BY "Н_СЕССИЯ"."ЧЛВК_ИД" ORDER BY "Н_СЕССИЯ"."ДАТА") as "prev_date",
        "Н_СЕССИЯ"."ДАТА" - LAG("Н_СЕССИЯ"."ДАТА") OVER (PARTITION BY "Н_СЕССИЯ"."ЧЛВК_ИД" ORDER BY "Н_СЕССИЯ"."ДАТА") as "inter
    FROM
        "Н_СЕССИЯ"
    ) as "session_interval"
JOIN "Н_ЛЮДИ" ON "session_interval"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
JOIN "Н_СЕССИЯ" ON "session_interval"."ЧЛВК_ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД"
WHERE "interval" IS NOT NULL
GROUP BY "Н_ЛЮДИ"."ИД"
ORDER BY "avg_interval" DESC
LIMIT 10;
```

session_interval

Output | Result 4

10 rows | CSV

	"ИД"	avg_interval	exam_dates
1	100537	732	{2007-06-15 00:00:00.000000,2007-06-15 00:00:00.000000,2011-01-29 00:00:00.000000,2011…
2	149706	726	{2009-06-22 00:00:00.000000,2011-06-18 00:00:00.000000}
3	132493	722	{2005-06-14 00:00:00.000000,2007-06-06 00:00:00.000000}
4	100283	680.3333333333334	{2005-06-18 00:00:00.000000,2005-06-18 00:00:00.000000,2005-06-18 00:00:00.000000,2007…

Вывод

При выполнении данной лабораторной работы я познакомилась с основными командами языка SQL (и диалекта PostgreSQL), а в особенности – всем, что связано с соединениями и выборками.