



Факультет программной инженерии и компьютерной техники

Информационные системы и базы данных

Лабораторная работа №3

Преподаватель: Николаев Владимир Вячеславович

Выполнил: Ле Чонг Дат

Группа: Р33302

2023 г.

SQL запросы

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

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

- Таблицы: Н_ОЦЕНКИ, Н_ВЕДОМОСТИ.
- Вывести атрибуты: Н_ОЦЕНКИ.КОД, Н_ВЕДОМОСТИ.ДАТА.
- Фильтры (AND):
 - а) Н_ОЦЕНКИ.ПРИМЕЧАНИЕ = 'освобождение'.
 - б) Н_ВЕДОМОСТИ.ИД < 1426978.
- Вид соединения: LEFT JOIN.

```
SELECT
    Н_ОЦЕНКИ.КОД,
    Н_ВЕДОМОСТИ.ДАТА
FROM
    Н_ОЦЕНКИ
LEFT JOIN Н_ВЕДОМОСТИ ON Н_ОЦЕНКИ.КОД = Н_ВЕДОМОСТИ.ОЦЕНКА
WHERE
    Н_ОЦЕНКИ.ПРИМЕЧАНИЕ = 'освобождение'
    AND Н_ВЕДОМОСТИ.ИД < 1426978;
```

```

ucheb=> SELECT
        Н_ОЦЕНКИ.КОД,
        Н_ВЕДОМОСТИ.ДАТА
FROM
        Н_ОЦЕНКИ
LEFT JOIN Н_ВЕДОМОСТИ ON Н_ОЦЕНКИ.КОД = Н_ВЕДОМОСТИ.ОЦЕНКА
WHERE
        Н_ОЦЕНКИ.ПРИМЕЧАНИЕ = 'освобождение'
        AND Н_ВЕДОМОСТИ.ИД < 1426978;

```

КОД	ДАТА
осв	2010-06-02 00:00:00
осв	2010-06-02 00:00:00
осв	2010-06-02 00:00:00
осв	2010-06-02 00:00:00
осв	2010-06-02 00:00:00
осв	2010-06-02 00:00:00
осв	2010-06-02 00:00:00
осв	2010-05-24 00:00:00
осв	2010-05-24 00:00:00
осв	2010-05-24 00:00:00
осв	2010-05-24 00:00:00
осв	2010-05-24 00:00:00
осв	2010-05-24 00:00:00
осв	2010-05-24 00:00:00
осв	2010-05-24 00:00:00
осв	2011-02-09 00:00:00
осв	2011-02-09 00:00:00
осв	2011-02-09 00:00:00
осв	2011-02-09 00:00:00

Рис. 1: Task 1

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.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ИД, Н_СЕССИЯ.ЧЛВК_ИД.
- Фильтры (AND):
 - а) Н_ЛЮДИ.ИД < 163484.
 - б) Н_ВЕДОМОСТИ.ИД > 1250972.
 - в) Н_СЕССИЯ.ИД > 1975.
- Вид соединения: INNER JOIN.

```

SELECT
        Н_ЛЮДИ.ФАМИЛИЯ,
        Н_ВЕДОМОСТИ.ИД,
        Н_СЕССИЯ.ЧЛВК_ИД
FROM
        Н_ЛЮДИ
INNER JOIN Н_ВЕДОМОСТИ ON Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД
INNER JOIN Н_СЕССИЯ ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД
WHERE
        Н_ЛЮДИ.ИД < 163484

```

```

AND H_ВЕДОМОСТИ.ИД > 1250972
AND H_СЕССИЯ.ИД > 1975;

```

```

ucheb=> SELECT
        H_ЛЮДИ.ФАМИЛИЯ,
        H_ВЕДОМОСТИ.ИД,
        H_СЕССИЯ.ЧЛВК_ИД
FROM
        H_ЛЮДИ
        INNER JOIN H_ВЕДОМОСТИ ON H_ЛЮДИ.ИД = H_ВЕДОМОСТИ.ЧЛВК_ИД
        INNER JOIN H_СЕССИЯ ON H_ЛЮДИ.ИД = H_СЕССИЯ.ЧЛВК_ИД
WHERE
        H_ЛЮДИ.ИД < 163484
        AND H_ВЕДОМОСТИ.ИД > 1250972
        AND H_СЕССИЯ.ИД > 1975;
ФАМИЛИЯ | ИД | ЧЛВК_ИД
-----+-----+-----
(0 строк)

```

Рис. 2: Task 2

3. Вывести число студентов группы 3102, которые старше 25 лет. Ответ должен содержать только одно число.

```

SELECT
        COUNT(*)
FROM
        H_УЧЕНИКИ
        LEFT JOIN H_ЛЮДИ ON H_УЧЕНИКИ.ЧЛВК_ИД = H_ЛЮДИ.ИД
WHERE
        H_УЧЕНИКИ.ГРУППА = '3102'
        AND EXTRACT(YEAR FROM AGE(H_ЛЮДИ.ДАТА_РОЖДЕНИЯ)) > 25;

```

```

ucheb=> \t
Режим вывода только кортежей включён.
ucheb=> SELECT
        COUNT(*)
FROM
        H_УЧЕНИКИ
        LEFT JOIN H_ЛЮДИ ON H_УЧЕНИКИ.ЧЛВК_ИД = H_ЛЮДИ.ИД
WHERE
        H_УЧЕНИКИ.ГРУППА = '3102'
        AND EXTRACT(YEAR FROM AGE(H_ЛЮДИ.ДАТА_РОЖДЕНИЯ)) > 25;
154

```

Рис. 3: Task 3

4. Выдать различные фамилии людей и число людей с каждой из этих фамилий, ограничив список фамилиями, встречающимися менее 10 раз на кафедре вычислительной техники. Для реализации использовать подзапрос.

```

SELECT
    H_ЛЮДИ.ФАМИЛИЯ,
    COUNT(*)
FROM
    H_ЛЮДИ
WHERE
    H_ЛЮДИ.ФАМИЛИЯ NOT IN (
        SELECT
            H_ЛЮДИ.ФАМИЛИЯ
        FROM
            H_ЛЮДИ
            LEFT JOIN H_ВЕДОМОСТИ ON H_ЛЮДИ.ИД = H_ВЕДОМОСТИ.ЧЛВК_ИД
            LEFT JOIN H_ОТДЕЛЫ ON H_ВЕДОМОСТИ.ОТД_ИД = H_ОТДЕЛЫ.ИД
        WHERE
            H_ОТДЕЛЫ.ИМЯ_В_ИМИН_ПАДЕЖЕ = 'кафедра вычислительной техники'
        GROUP BY
            H_ЛЮДИ.ФАМИЛИЯ
        HAVING
            COUNT(*) >= 10
    )
GROUP BY
    H_ЛЮДИ.ФАМИЛИЯ;

```

```

ucheb=> SELECT
        Н_ЛЮДИ.ФАМИЛИЯ,
        COUNT(*)
FROM
        Н_ЛЮДИ
WHERE
        Н_ЛЮДИ.ФАМИЛИЯ NOT IN (
                SELECT
                        Н_ЛЮДИ.ФАМИЛИЯ
                FROM
                        Н_ЛЮДИ
                LEFT JOIN Н_ВЕДОМОСТИ ON Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД
                LEFT JOIN Н_ОТДЕЛЫ ON Н_ВЕДОМОСТИ.ОТД_ИД = Н_ОТДЕЛЫ.ИД
                WHERE
                        Н_ОТДЕЛЫ.ИМЯ_В_ИМИН_ПАДЕЖЕ = 'кафедра вычислительной техники'
                GROUP BY
                        Н_ЛЮДИ.ФАМИЛИЯ
                HAVING
                        COUNT(*) >= 10
        )
GROUP BY
        Н_ЛЮДИ.ФАМИЛИЯ;

```

ФАМИЛИЯ	count
Лу Цзюньянь	1
Кайдановский	1
Орлов	6
Залесная	1
Королев	5
Панкратов	2
Баданов	1
Раскилас	1
Гарбуза	1
Громов	3
Мишичев	1
Завадис	1
Панасенко	1
Гагарский	1
Лушин	1
Дименко	1
Прядка	1
Баринова	1
Новикова	3
Чжан	6
Дворкин	2
Медов	1
Дудинский	1
Ворожбянский	1

Рис. 4: Task 4

5. Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст), где средний возраст больше минимального возраста в группе 3100.

```

WITH JoinedData AS (
    SELECT
        Н_УЧЕНИКИ.ГРУППА,
        AGE(Н_ЛЮДИ.ДАТА_РОЖДЕНИЯ) AS Age
    FROM
        Н_ЛЮДИ
        INNER JOIN Н_УЧЕНИКИ ON Н_ЛЮДИ.ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
),

```

```

MinAgeForGroup AS (
    SELECT
        MIN(Age) AS MinAge
    FROM
        JoinedData
    WHERE
        ГРУППА = '3100'
)
SELECT
    ГРУППА,
    AVG(Age)
FROM
    JoinedData
GROUP BY
    ГРУППА
HAVING
    AVG(Age) > (SELECT MinAge FROM MinAgeForGroup);

учеб=> WITH JoinedData AS (
    SELECT
        Н_УЧЕНИКИ.ГРУППА,
        AGE(Н_ЛЮДИ.ДАТА_РОЖДЕНИЯ) AS Age
    FROM
        Н_ЛЮДИ
    INNER JOIN Н_УЧЕНИКИ ON Н_ЛЮДИ.ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
),
MinAgeForGroup AS (
    SELECT
        MIN(Age) AS MinAge
    FROM
        JoinedData
    WHERE
        ГРУППА = '3100'
)
SELECT
    ГРУППА,
    AVG(Age) > (SELECT MinAge FROM MinAgeForGroup);
ГРУППА | avg
-----+-----
354     | 26 years 1 mon 19 days 19:25:42.894171
2508    | 34 years 5 mons 37 days 00:24:49.676028
5509    | 36 years 3 mons 23 days 13:05:27.304145
551     | 24 years 9 mons 25 days 18:00:00.012343
5114    | 29 years 7 mons 35 days 04:30:00
395     | 25 years 11 mons 22 days 23:50:49.644107
250     | 32 years 2 mons 15 days 12:53:20.0288
3104    | 31 years 8 mons 29 days 02:18:47.741031
219     | 36 years 8 mons 20 days 03:05:08.583771
555     | 26 years 9 mons 42 days 00:00:00.006646
6125    | 36 years 1 mon 29 days 08:06:00
2125    | 31 years 7 mons 38 days 11:41:32.2944

```

Рис. 5: Task 5

6. Получить список студентов, отчисленных после 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
    U.ГРУППА,
    U.ЧЛВК_ИД,
    L.ФАМИЛИЯ,
    L.ИМЯ,
    L.ОТЧЕСТВО,
    U.ИД
FROM
    Н_УЧЕНИКИ U
    INNER JOIN Н_ЛЮДИ L ON L.ИД = U.ЧЛВК_ИД
WHERE
    U.КОНЕЦ_ПО_ПРИКАЗУ = '2012-08-31 00:00:00.00000'
    AND EXISTS (
        SELECT 1
        FROM Н_ПЛАНЫ P
        INNER JOIN Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ NS ON NS.ИД = P.НАПС_ИД
        INNER JOIN Н_НАПР_СПЕЦ NNS ON NNS.ИД = NS.НС_ИД
        INNER JOIN Н_ФОРМЫ_ОБУЧЕНИЯ FO ON P.ФО_ИД = FO.ИД
        WHERE
            P.ПЛАН_ИД = U.ПЛАН_ИД
            AND FO.НАИМЕНОВАНИЕ IN ('Заочная', 'Очная')
            AND NNS.НАИМЕНОВАНИЕ = 'Программная инженерия'
    );
```



```

ucheб=> SELECT
        U.ГРУППА,
        U.ЧЛВК_ИД,
        L.ФАМИЛИЯ,
        L.ИМЯ,
        L.ОТЧЕСТВО,
        U.ИД
FROM
        H_УЧЕНИКИ U
        INNER JOIN H_ЛЮДИ L ON L.ИД = U.ЧЛВК_ИД
WHERE
        U.КОНЕЦ_ПО_ПРИКАЗУ = '2012-08-31 00:00:00.00000'
        AND EXISTS (
                SELECT 1
                FROM H_ПЛАНЫ P
                INNER JOIN H_НАПРАВЛЕНИЯ СПЕЦИАЛ NS ON NS.ИД = P.НАПС_ИД
                INNER JOIN H_НАПР_СПЕЦ NNS ON NNS.ИД = NS.НС_ИД
                INNER JOIN H_ФОРМЫ_ОБУЧЕНИЯ FO ON P.ФО_ИД = FO.ИД
                WHERE
                        P.ПЛАН_ИД = U.ПЛАН_ИД
                        AND FO.НАИМЕНОВАНИЕ IN ('Заочная', 'Очная')
                        AND NNS.НАИМЕНОВАНИЕ = 'Программная инженерия'
        );
ГРУППА | ЧЛВК_ИД | ФАМИЛИЯ | ИМЯ | ОТЧЕСТВО | ИД
-----+-----+-----+-----+-----+-----
(0 строк)

```

Рис. 6: Task 6

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

```

SELECT *
FROM H_ЛЮДИ
WHERE H_ЛЮДИ.ИД NOT IN (
        SELECT ЧЛВК_ИД
        FROM H_УЧЕНИКИ
);

```

```

ucheб=> SELECT *
FROM H_ЛЮДИ
WHERE H_ЛЮДИ.ИД NOT IN (
        SELECT ЧЛВК_ИД
        FROM H_УЧЕНИКИ
);

```

ИД	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ПИН	ИИН	ДАТА_РОЖДЕНИЯ	ПОЛ	МЕСТО_РОЖДЕНИЯ	ИНСТРАН	КТО_СОЗДАЛ	КОГДА_СОЗДАЛ	КТО_ИЗМЕНИЛ	КОГДА_ИЗМЕНИЛ
128214	Парфенов	Петр	Сергеевич			1979-12-06 00:00:00	М		0	ISU_UCHEB	2011-11-08 15:56:03	ISU_UCHEB	2011-11-08 15:56:03 9
999-09-09 00:00:00													
132493	Алексина	Алла	Фелипповна			2037-03-28 00:00:00	Ж		0	ISU_UCHEB	2011-11-08 15:56:03	ISU_UCHEB	2011-11-08 15:56:03 9
999-09-09 00:00:00													
143440	Рыбац	Анатолий	Львович			2018-12-23 00:00:00	М		0	ISU_UCHEB	2011-11-08 15:56:03	ISU_UCHEB	2011-11-08 15:56:03 9
999-09-09 00:00:00													
103749	Денисова	Эльвира	Викторовна			1970-05-01 00:00:00	Ж		0	ISU_UCHEB	2011-11-08 15:56:03	ISU_UCHEB	2011-11-08 15:56:03 9
999-09-09 00:00:00													
103888	Завгин	Виктор	Фомич			2047-07-26 00:00:00	М		0	ISU_UCHEB	2011-11-08 15:56:03	ISU_UCHEB	2011-11-08 15:56:03 9
999-09-09 00:00:00													
103844	Богдан	Валентин	Николаевич			2047-01-11 00:00:00	М		0	ISU_UCHEB	2011-11-08 15:56:03	ISU_UCHEB	2011-11-08 15:56:03 9
999-09-09 00:00:00													
103946	Новгородская	Татьяна	Алексеевна			2048-11-28 00:00:00	Ж		0	ISU_UCHEB	2011-11-08 15:56:03	ISU_UCHEB	2011-11-08 15:56:03 9

Рис. 7: Task 7

Conclusion

In this SQL lab session, we have explored various aspects of SQL queries, ranging from basic selection and joining of tables to more complex operations such as subqueries and common table expressions. Through the exercises, we have gained a deeper understanding of how to effectively manipulate and retrieve data from relational databases. The ability to formulate precise and optimized SQL queries is essential for any data-related task, and the skills acquired in this lab will be invaluable for future database management and data analysis projects. Furthermore, this session has reinforced the importance of clean and well-structured SQL code for enhancing readability and maintainability. Overall, the lab has been a significant step in mastering SQL and its applications in real-world data scenarios.