Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерных технологий

Лабораторная работа №2

Вариант 65445

Выполнила:

Павличенко Софья Алексеевна, Р3115

Проверил:

Мартин Райла

Санкт-Петербург 2023г.

Оглавление

[Задание 3](#_Toc162811745)

[Реализация запросов на SQL 4](#_Toc162811746)

[Заключение 6](#_Toc162811747)

# Задание

По варианту, выданному преподавателем, составить и выполнить запросы к [базе данных "Учебный процесс"](https://se.ifmo.ru/documents/10180/733702/%D0%91%D0%94+%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D1%8B%D0%B9+%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81.pdf/2eae3fcd-ea34-4496-924b-6ee4e889a9e5).

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):  
   a) Н\_ОЦЕНКИ.КОД = 5.  
   b) Н\_ВЕДОМОСТИ.ДАТА = 2010-06-18.  
   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.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.ЧЛВК\_ИД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ОТЧЕСТВО = Александрович.  
   b) Н\_ВЕДОМОСТИ.ИД = 1426978.  
   c) Н\_СЕССИЯ.ДАТА < 2002-01-04.  
   Вид соединения: RIGHT JOIN.
3. Вывести число дней без учета повторений.  
   При составлении запроса нельзя использовать DISTINCT.
4. Выдать различные отчества людей и число людей с каждой из этих отчеств, ограничив список отчествами, встречающимися менее 50 раз на на заочной форме обучения.  
   Для реализации использовать соединение таблиц.
5. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка), у которых средняя оценка не меньше максимальной оценк(е|и) в группе 3100.
6. Получить список студентов, отчисленных до первого сентября 2012 года с очной или заочной формы обучения. В результат включить:  
   номер группы;  
   номер, фамилию, имя и отчество студента;  
   номер пункта приказа;  
   Для реализации использовать подзапрос с IN.
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.

# Реализация запросов на SQL

1.

SELECT Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ДАТА FROM Н\_ОЦЕНКИ  
JOIN Н\_ВЕДОМОСТИ ON Н\_ВЕДОМОСТИ.ОЦЕНКА = Н\_ОЦЕНКИ.КОД  
WHERE Н\_ОЦЕНКИ.КОД = '5' AND Н\_ВЕДОМОСТИ.ДАТА = '2010-06-18 00:00:00';

2.

SELECT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.ЧЛВК\_ИД FROM Н\_ЛЮДИ  
RIGHT JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  
RIGHT JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД  
WHERE Н\_ЛЮДИ.ОТЧЕСТВО = 'Александрович' AND Н\_ВЕДОМОСТИ.ИД = 1426978 AND Н\_СЕССИЯ.ДАТА < '2002-01-04 00:00:00';

3.

SELECT *COUNT* (*\**)  
FROM (  
 SELECT *COUNT*(Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ) FROM Н\_ЛЮДИ  
 GROUP BY Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ  
) AS unique\_birthdays;

4.

SELECT Н\_ЛЮДИ.ОТЧЕСТВО, *COUNT*(Н\_ЛЮДИ.ИД) AS КОЛИЧЕСТВО FROM Н\_ЛЮДИ  
LEFT JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД  
JOIN Н\_ПЛАНЫ ON Н\_УЧЕНИКИ.ПЛАН\_ИД = Н\_ПЛАНЫ.ИД  
JOIN Н\_ФОРМЫ\_ОБУЧЕНИЯ ON Н\_ФОРМЫ\_ОБУЧЕНИЯ.ИД = Н\_ПЛАНЫ.ФО\_ИД  
GROUP BY Н\_ЛЮДИ.ОТЧЕСТВО  
HAVING *SUM*(CASE WHEN Н\_ФОРМЫ\_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Заочная' THEN 1 ELSE 0 END) < 50  
ORDER BY Н\_ЛЮДИ.ОТЧЕСТВО;

5.

SELECT Н\_УЧЕНИКИ.ИД, Н\_ЛЮДИ.ИМЯ, Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ЛЮДИ.ОТЧЕСТВО,

*AVG*(Н\_ВЕДОМОСТИ.ОЦЕНКА::integer) AS "СРЕДНЯЯ ОЦЕНКА"  
FROM Н\_УЧЕНИКИ  
JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД  
JOIN Н\_ВЕДОМОСТИ ON Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД  
JOIN Н\_ОЦЕНКИ ON Н\_ОЦЕНКИ.КОД = Н\_ВЕДОМОСТИ.ОЦЕНКА AND Н\_ОЦЕНКИ.СОРТ < 5  
WHERE Н\_УЧЕНИКИ.ГРУППА = '4100'   
GROUP BY Н\_УЧЕНИКИ.ИД, Н\_ЛЮДИ.ИМЯ, Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ЛЮДИ.ОТЧЕСТВО  
  
HAVING *AVG*(Н\_ВЕДОМОСТИ.ОЦЕНКА::integer) >= (  
 SELECT *MAX*(Н\_ВЕДОМОСТИ.ОЦЕНКА::integer) FROM Н\_ВЕДОМОСТИ  
 JOIN Н\_ЛЮДИ ON Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД  
 JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД  
 JOIN Н\_ОЦЕНКИ ON Н\_ОЦЕНКИ.КОД = Н\_ВЕДОМОСТИ.ОЦЕНКА AND Н\_ОЦЕНКИ.СОРТ < 5  
 WHERE Н\_УЧЕНИКИ.ГРУППА = '3100');

CREATE VIEW max\_fear AS  
SELECT hunter.id, hunter.name, hunter.surname, *MAX*(prey.weight) from hunter  
JOIN prey ON hunter.id = prey.hunter\_id  
JOIN hunter\_feeling ON hunter.id = hunter\_feeling.hunter\_id  
WHERE hunter\_feeling.feeling = 'страх'  
GROUP BY (hunter.id, hunter.name, hunter.surname);  
  
CREATE MATERIALIZED VIEW avg\_hungry AS  
SELECT hunter.id, hunter.name, hunter.surname, *AVG*(prey.weight) from hunter  
JOIN prey ON hunter.id = prey.hunter\_id  
JOIN hunter\_feeling ON hunter.id = hunter\_feeling.hunter\_id  
WHERE hunter\_feeling.feeling = 'голод'  
GROUP BY (hunter.id, hunter.name, hunter.surname);  
  
SELECT avg\_hungry.id, avg\_hungry.name, avg\_hungry.surname, avg\_hungry.avg FROM avg\_hungry  
JOIN max\_fear ON avg\_hungry.id = max\_fear.id  
WHERE avg\_hungry.avg > max\_fear.max;

6.

SELECT Н\_УЧЕНИКИ.ГРУППА, Н\_УЧЕНИКИ.ИД, Н\_ЛЮДИ.ИМЯ, Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ЛЮДИ.ОТЧЕСТВО, Н\_УЧЕНИКИ.П\_ПРКОК\_ИД FROM Н\_УЧЕНИКИ  
JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД  
WHERE Н\_УЧЕНИКИ.ИД IN  
(SELECT Н\_УЧЕНИКИ.ИД FROM Н\_УЧЕНИКИ  
JOIN Н\_ПЛАНЫ ON Н\_УЧЕНИКИ.ПЛАН\_ИД = Н\_ПЛАНЫ.ИД  
JOIN Н\_ФОРМЫ\_ОБУЧЕНИЯ ON Н\_ФОРМЫ\_ОБУЧЕНИЯ.ИД = Н\_ПЛАНЫ.ФО\_ИД  
WHERE Н\_УЧЕНИКИ.ПРИЗНАК = 'отчисл' AND Н\_УЧЕНИКИ.СОСТОЯНИЕ = 'утвержден'AND Н\_ФОРМЫ\_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ IN ('Заочная', 'Очная')  
AND Н\_УЧЕНИКИ.КОНЕЦ < '2012-09-01 00:00:00');

7.

SELECT Н\_ЛЮДИ.ИД, Н\_ЛЮДИ.ИМЯ, Н\_ЛЮДИ.ФАМИЛИЯ, 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;

# Заключение

В результате выполнения лабораторной работы я научилась создавать инфологическую и даталогическую модели, на их основе реализовывать базу данных с помощью SQL.