Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №2

Вариант 465592

Выполнил:

Горин Семён Дмитриевич

Группа P3108

Проверила:

Заболотняя Ольга Михайловна

**Содержание**

[Задание 2](#_Toc196220420)

[Реализация запросов на SQL 3](#_Toc196220421)

[Выводы 7](#_Toc196220422)

# Задание

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

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ИД.

Фильтры (AND):

a) Н\_ЛЮДИ.ФАМИЛИЯ < Афанасьев.

b) Н\_ВЕДОМОСТИ.ИД < 1490007.

c) Н\_ВЕДОМОСТИ.ИД < 1490007.

Вид соединения: INNER JOIN.

1. 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.ИД, Н\_СЕССИЯ.ДАТА.

Фильтры (AND):

a) Н\_ЛЮДИ.ОТЧЕСТВО = Георгиевич.

b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 117219.

Вид соединения: LEFT JOIN.

1. Вывести число названий дисциплин без учета повторений.

При составлении запроса нельзя использовать DISTINCT.

1. Найти группы, в которых в 2011 году было более 5 обучающихся студентов на ФКТИУ.

Для реализации использовать подзапрос.

1. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка), у которых средняя оценка не больше максимальной оценк(е|и) в группе 1101.
2. Получить список студентов, зачисленных после первого сентября 2012 года на первый курс заочной формы обучения (специальность: 230101). В результат включить:

номер группы;

номер, фамилию, имя и отчество студента;

номер и состояние пункта приказа;

Для реализации использовать подзапрос с IN.

1. Сформировать запрос для получения числа в группе No 3100 троечников.

# Реализация запросов на SQL

**BEGIN;**

/\* Выборка №1

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

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ИД.

Фильтры (AND):

a) Н\_ЛЮДИ.ФАМИЛИЯ < Афанасьев.

b) Н\_ВЕДОМОСТИ.ИД < 1490007.

c) Н\_ВЕДОМОСТИ.ИД < 1490007.

Вид соединения: INNER JOIN.

\*/

**SELECT**

Н\_ЛЮДИ**.**ОТЧЕСТВО **AS** second\_name**,**

Н\_ВЕДОМОСТИ**.**ИД **as** document\_id

**FROM**

Н\_ЛЮДИ **INNER** **JOIN** Н\_ВЕДОМОСТИ **ON** Н\_ЛЮДИ**.**ИД **=** Н\_ВЕДОМОСТИ**.**ЧЛВК\_ИД

**WHERE**

Н\_ЛЮДИ**.**ФАМИЛИЯ **<** 'Афанасьев'

**AND** Н\_ВЕДОМОСТИ**.**ИД **<** 1490007**;**

/\* Выборка №2

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.ИД, Н\_СЕССИЯ.ДАТА.

Фильтры (AND):

a) Н\_ЛЮДИ.ОТЧЕСТВО = Георгиевич.

b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 117219.

Вид соединения: LEFT JOIN.

1.0.1 Хотят чтобы для каждого из студентов были выведены отчество, номер ведомости и дата сессии соответствующие условию(при этом если студент не участвовал в экзаменах информация о нем все еще выводится).

\*/

**SELECT**

Н\_ЛЮДИ**.**ОТЧЕСТВО **AS** second\_name**,**

Н\_ВЕДОМОСТИ**.**ИД **AS** document\_id**,**

Н\_СЕССИЯ**.**ДАТА **AS** date\_of\_exam

**FROM**

Н\_ВЕДОМОСТИ **LEFT** **JOIN** **(**Н\_ЛЮДИ **LEFT** **JOIN** Н\_СЕССИЯ **ON** Н\_ЛЮДИ**.**ИД **=** Н\_СЕССИЯ**.**ЧЛВК\_ИД**)** **ON** Н\_ЛЮДИ**.**ИД **=** Н\_ВЕДОМОСТИ**.**ЧЛВК\_ИД

**WHERE**

Н\_ЛЮДИ**.**ОТЧЕСТВО **=** 'Георгиевич'

**AND** Н\_ВЕДОМОСТИ**.**ЧЛВК\_ИД **>** 117219**;**

/\* Выборка №3

3. Вывести число названий дисциплин без учета повторений.

При составлении запроса нельзя использовать DISTINCT.

\*/

**SELECT**

**COUNT(\*)** -- 1.0.1 Изменено с COUNT(tmp.НАИМЕНОВАНИЕ)

**FROM**

**(SELECT**

Н\_ДИСЦИПЛИНЫ**.**НАИМЕНОВАНИЕ

**FROM**

Н\_ДИСЦИПЛИНЫ

**GROUP** **BY**

Н\_ДИСЦИПЛИНЫ**.**НАИМЕНОВАНИЕ

**)** **AS** tmp**;**

/\* Вариант с агрегатной функцией в подзапросе \*/

**SELECT** **COUNT(\*)** -- 1.0.1 Изменено с COUNT(tmp.cnt)

**FROM(SELECT** **COUNT(**Н\_ДИСЦИПЛИНЫ**.**НАИМЕНОВАНИЕ**)** **as** cnt

**FROM** Н\_ДИСЦИПЛИНЫ

**GROUP** **BY** Н\_ДИСЦИПЛИНЫ**.**НАИМЕНОВАНИЕ**)** **as** tmp**;**

/\* Выборка №4

4. Найти группы, в которых в 2011 году было более 5 обучающихся студентов на ФКТИУ.

Для реализации использовать подзапрос.

\*/

**SELECT**

Н\_УЧЕНИКИ**.**ГРУППА **AS** **group,** **COUNT(**ЧЛВК\_ИД**)** -- 1.0.1 Добавлен COUNT для проверки верности условия

**FROM**

Н\_УЧЕНИКИ

**WHERE**

Н\_УЧЕНИКИ**.**НАЧАЛО **<=** '12-12-2011' **AND** Н\_УЧЕНИКИ**.**КОНЕЦ **>=** '01-01-2011' -- фильтрация по времени обучения(обучался в 2011 году)

**AND** Н\_УЧЕНИКИ**.**ПЛАН\_ИД **IN** **(** -- фильтрация по факультету обучения

**SELECT** Н\_ПЛАНЫ**.**ИД

**FROM** Н\_ПЛАНЫ **JOIN** Н\_ОТДЕЛЫ **ON** Н\_ПЛАНЫ**.**ОТД\_ИД **=** Н\_ОТДЕЛЫ**.**ИД -- выборка всех таких ID планов, из объединения планов и отделов, что короткое имя отдела - КТиУ.

**WHERE**

Н\_ОТДЕЛЫ**.**КОРОТКОЕ\_ИМЯ **=** 'КТиУ'

**)**

**GROUP** **BY**

Н\_УЧЕНИКИ**.**ГРУППА -- ученики группируются по группам обучения, чтобы в дальнейшем применить фильтрацию(больше 5 обучающихся на ФКТиУ) по каждой из групп.

**HAVING**

**COUNT(**ЧЛВК\_ИД**)** **>** 5**;**

/\* Выборка №5

5. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка), у которых средняя оценка не больше максимальной оценк(е|и) в группе 1101.

\*/

**WITH** max\_value\_from\_group\_1101\_table **AS** **(SELECT** **MAX(CAST(**Н\_ВЕДОМОСТИ**.**ОЦЕНКА **AS** INTEGER**))** **AS** max\_value\_from\_group\_1101

**FROM** Н\_ВЕДОМОСТИ **JOIN** Н\_УЧЕНИКИ **ON** Н\_ВЕДОМОСТИ**.**ЧЛВК\_ИД **=** Н\_УЧЕНИКИ**.**ЧЛВК\_ИД

**WHERE** -- выборка максимальной оценки в группе 1101

Н\_УЧЕНИКИ**.**ГРУППА **=** '1101' -- фильтрация по группе

**AND** Н\_ВЕДОМОСТИ**.**ОЦЕНКА **IN** **(**'1'**,** '2'**,** '3'**,** '4'**,** '5'**)** -- фильтрация оценок по численно-измеримым

**)**

**SELECT**

Н\_УЧЕНИКИ**.**ЧЛВК\_ИД **AS** isu\_id**,**

Н\_ЛЮДИ**.**ФАМИЛИЯ **AS** last\_name**,**

Н\_ЛЮДИ**.**ИМЯ **AS** first\_name**,**

Н\_ЛЮДИ**.**ОТЧЕСТВО **AS** second\_name**,**

**AVG(CAST(**Н\_ВЕДОМОСТИ**.**ОЦЕНКА **AS** INTEGER**))** **AS** avg\_score -- средняя оценка учащегося

**FROM**

**(**Н\_ВЕДОМОСТИ **JOIN** Н\_УЧЕНИКИ **ON** Н\_ВЕДОМОСТИ**.**ЧЛВК\_ИД **=** Н\_УЧЕНИКИ**.**ЧЛВК\_ИД**)** **JOIN** Н\_ЛЮДИ **ON** Н\_УЧЕНИКИ**.**ЧЛВК\_ИД **=** Н\_ЛЮДИ**.**ИД

**WHERE**

Н\_УЧЕНИКИ**.**ГРУППА **=** '4100' -- фильтрация по группе

**AND** Н\_ВЕДОМОСТИ**.**ОЦЕНКА **IN** **(**'1'**,** '2'**,** '3'**,** '4'**,** '5'**)** -- фильтрация оценок по численно-измеримым

**GROUP** **BY**

Н\_УЧЕНИКИ**.**ЧЛВК\_ИД**,** Н\_ЛЮДИ**.**ФАМИЛИЯ**,** Н\_ЛЮДИ**.**ИМЯ**,** Н\_ЛЮДИ**.**ОТЧЕСТВО -- группировка для нахождения среднего балла учащегося

**HAVING**

**AVG(CAST(**Н\_ВЕДОМОСТИ**.**ОЦЕНКА **AS** INTEGER**))** **<=** **(SELECT** max\_value\_from\_group\_1101 **FROM** max\_value\_from\_group\_1101\_table**);** -- фильтрация, после которой остаются лишь учащиеся со средним баллом меньшим максимальной оценки в группе 1101 UPD 1.0.1 Изменено в соответствии с правками (исправлен затуп)

/\* Выборка №6

6. Получить список студентов, зачисленных после первого сентября 2012 года на первый курс заочной формы обучения (специальность: 230101). В результат включить:

номер группы;

номер, фамилию, имя и отчество студента;

номер и состояние пункта приказа;

Для реализации использовать подзапрос с IN.

\*/

**SELECT**

Н\_УЧЕНИКИ**.**ГРУППА **AS** **group,**

Н\_УЧЕНИКИ**.**ЧЛВК\_ИД **AS** isu\_id**,**

Н\_ЛЮДИ**.**ФАМИЛИЯ **AS** last\_name**,**

Н\_ЛЮДИ**.**ИМЯ **AS** first\_name**,**

Н\_ЛЮДИ**.**ОТЧЕСТВО **AS** second\_name**,**

Н\_УЧЕНИКИ**.**П\_ПРКОК\_ИД **AS** admission\_order\_item**,**

Н\_УЧЕНИКИ**.**ПРИЗНАК **AS** current\_state\_in\_learning

**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

**WHERE**

Н\_УЧЕНИКИ**.**НАЧАЛО **>** '01-09-2012' -- Проверка на то что студент поступил после 1 сентября 2012 года

**AND** Н\_ПЛАНЫ**.**КУРС **=** 1 -- Проверка на то что студент поступил на 1 курс обучения

**AND** Н\_УЧЕНИКИ**.**СОСТОЯНИЕ **=** 'утвержден' -- Проверка что приказ был утвержден

**AND** Н\_ПЛАНЫ**.**ФО\_ИД **IN** **(SELECT** ИД **FROM** Н\_ФОРМЫ\_ОБУЧЕНИЯ **WHERE** НАИМЕНОВАНИЕ **=** 'Заочная'**)** -- Проверка на то что студент поступил на заочную форму обучения

**AND** Н\_ПЛАНЫ**.**НАПС\_ИД **IN** **(SELECT** Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ**.**ИД

**FROM**

Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ **JOIN** Н\_НАПР\_СПЕЦ **ON** Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ**.**НС\_ИД **=** Н\_НАПР\_СПЕЦ**.**ИД -- Проверка что студент поступил на специальность 230101

**WHERE**

Н\_НАПР\_СПЕЦ**.**КОД\_НАПРСПЕЦ **=** '230101'**);**

/\* Выборка №7

7. Сформировать запрос для получения числа в группе No 3100 троечников.

\*/

**WITH** bad\_studs\_table **AS** **(SELECT** Н\_УЧЕНИКИ**.**ЧЛВК\_ИД **FROM** Н\_УЧЕНИКИ **JOIN** Н\_ВЕДОМОСТИ **ON** Н\_УЧЕНИКИ**.**ЧЛВК\_ИД **=** Н\_ВЕДОМОСТИ**.**ЧЛВК\_ИД -- выборка студентов из группы 3100 со средним баллом ниже 3.5 (т.е. студентов-троечников)

**WHERE**

Н\_УЧЕНИКИ**.**ГРУППА **=** '3100'

**AND** Н\_ВЕДОМОСТИ**.**ОЦЕНКА **IN** **(**'1'**,** '2'**,** '3'**,** '4'**,** '5'**)**

**GROUP** **BY** Н\_УЧЕНИКИ**.**ЧЛВК\_ИД -- группировка по ID человека, чтобы в последствии отфильтровать по среднему баллу

**HAVING**

**AVG(CAST(**Н\_ВЕДОМОСТИ**.**ОЦЕНКА **AS** INTEGER**))** **<** 3.5**)** -- вычисление среднего балла каждого из студентов, и фильтрация тех чей балл выше или равен 3.5

**SELECT** **COUNT(**ЧЛВК\_ИД**)** **AS** bad\_studs\_amount

**FROM** bad\_studs\_table**;**

**END;**

# Выводы

В ходе лабораторной работы я научился писать сложные запросы к СУБД, сумел разобраться в хитросплетениях БД “Учебный процесс” и на практике понял как не стоит составлять ER-модели.