**Отчет по лабораторной работе №2.**

Вариант 0900909889899. Раевский Григорий, Р3121.

1. **Текст задания**
   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.ИД < 100012.  
      b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 163249.  
      Вид соединения: LEFT JOIN.
   2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
      Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
      Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.  
      Фильтры: (AND)  
      a) Н\_ЛЮДИ.ФАМИЛИЯ < Афанасьев.  
      b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 113409.  
      c) Н\_УЧЕНИКИ.НАЧАЛО = 1996-09-01.  
      Вид соединения: INNER JOIN.
   3. Вывести число студентов группы 3102, которые не имеет отчества.  
      Ответ должен содержать только одно число.
   4. Выдать различные фамилии людей и число людей с каждой из этих фамилий, ограничив список фамилиями, встречающимися ровно 10 раз на на очной форме обучения.  
      Для реализации использовать подзапрос.
   5. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка), у которых средняя оценка равна средней оценк(е|и) в группе 1100.
   6. Получить список студентов, зачисленных после первого сентября 2012 года на первый курс очной формы обучения (специальность: Программная инженерия). В результат включить:  
      номер группы;  
      номер, фамилию, имя и отчество студента;  
      номер и состояние пункта приказа;   
      Для реализации использовать подзапрос с EXISTS.
   7. Сформировать запрос для получения числа в группе No 3100 отличников.
2. **Запросы для PostgreSQL**

2.1 Решение:  
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.ИД < 100012.  
b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 163249.  
Вид соединения: LEFT JOIN.

SELECT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ИД

FROM Н\_ЛЮДИ

LEFT JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД=Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ИД < 163249 AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД >100012;

* 1. Решение:

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.  
Фильтры: (AND)  
a) Н\_ЛЮДИ.ФАМИЛИЯ < Афанасьев.  
b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 113409.  
c) Н\_УЧЕНИКИ.НАЧАЛО = 1996-09-01.  
Вид соединения: INNER JOIN.

SELECT Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО

FROM Н\_ЛЮДИ

INNER JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

INNER JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ФАМИЛИЯ < 'Афанасьев' AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 113409 AND

Н\_УЧЕНИКИ.НАЧАЛО = '1996-09-01';

* 1. Решение:

Вывести число студентов группы 3102, которые не имеет отчества.  
Ответ должен содержать только одно число.

SELECT COUNT(\*) FROM Н\_УЧЕНИКИ INNER JOIN Н\_ЛЮДИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД WHERE Н\_УЧЕНИКИ.ГРУППА = '3102' AND Н\_ЛЮДИ.ОТЧЕСТВО IS NULL;

* 1. Решение:

Выдать различные фамилии людей и число людей с каждой из этих фамилий, ограничив список фамилиями, встречающимися ровно 10 раз на на очной форме обучения.  
Для реализации использовать подзапрос.

SELECT Н\_ЛЮДИ.ФАМИЛИЯ, COUNT(\*) AS КОЛИЧЕСТВО

FROM Н\_ЛЮДИ

INNER JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

INNER JOIN Н\_ФОРМЫ\_ОБУЧЕНИЯ ON Н\_УЧЕНИКИ.ВИД\_ОБУЧ\_ИД = Н\_ФОРМЫ\_ОБУЧЕНИЯ.ИД

WHERE Н\_ФОРМЫ\_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Очная'

AND Н\_ЛЮДИ.ФАМИЛИЯ IN

(SELECT Н\_ЛЮДИ.ФАМИЛИЯ

FROM Н\_ЛЮДИ

INNER JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

INNER JOIN Н\_ФОРМЫ\_ОБУЧЕНИЯ ON Н\_УЧЕНИКИ.ВИД\_ОБУЧ\_ИД = Н\_ФОРМЫ\_ОБУЧЕНИЯ.ИД

WHERE Н\_ФОРМЫ\_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Очная'

GROUP BY Н\_ЛЮДИ.ФАМИЛИЯ

HAVING COUNT(\*) = 10)

GROUP BY Н\_ЛЮДИ.ФАМИЛИЯ;

* 1. Решение:

Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка), у которых средняя оценка равна средней оценк(е|и) в группе 1100.

SELECT H.ИД AS Номер, H.ФАМИЛИЯ, H.ИМЯ, H.ОТЧЕСТВО, AVG(CASE

WHEN Вед.ОЦЕНКА = 'осв' THEN 0

WHEN Вед.ОЦЕНКА = 'неявка' THEN 0

WHEN Вед.ОЦЕНКА = 'зачет' THEN 5

WHEN Вед.ОЦЕНКА = 'незач' THEN 0

WHEN Вед.ОЦЕНКА = '99' THEN 6

ELSE CAST(Вед.ОЦЕНКА AS INT) END) AS Ср\_оценка

FROM Н\_ЛЮДИ H

JOIN Н\_УЧЕНИКИ Уч ON Уч.ЧЛВК\_ИД = H.ИД

JOIN Н\_ВЕДОМОСТИ Вед ON Вед.ЧЛВК\_ИД = H.ИД

WHERE Уч.ГРУППА = '4100'

GROUP BY H.ИД, H.ФАМИЛИЯ, H.ИМЯ, H.ОТЧЕСТВО

HAVING AVG(CASE WHEN Вед.ОЦЕНКА = 'осв' THEN 0

WHEN Вед.ОЦЕНКА = 'зачет' THEN 5

WHEN Вед.ОЦЕНКА = 'неявка' THEN 0

WHEN Вед.ОЦЕНКА = 'незач' THEN 0

WHEN Вед.ОЦЕНКА = '99' THEN 6

ELSE CAST(Вед.ОЦЕНКА AS INT) END) = (SELECT AVG(CASE WHEN Вед.ОЦЕНКА = 'осв' THEN 0

WHEN Вед.ОЦЕНКА = 'зачет' THEN 5

WHEN Вед.ОЦЕНКА = 'неявка' THEN 0

WHEN Вед.ОЦЕНКА = 'незач' THEN 0

WHEN Вед.ОЦЕНКА = '99' THEN 6

ELSE CAST(Вед.ОЦЕНКА AS INT) END)

FROM Н\_УЧЕНИКИ Уч2

JOIN Н\_ВЕДОМОСТИ Вед ON Вед.ЧЛВК\_ИД = Уч2.ЧЛВК\_ИД

WHERE Уч2.ГРУППА = '1100');

* 1. Решение:

Получить список студентов, зачисленных после первого сентября 2012 года на первый курс очной формы обучения (специальность: Программная инженерия). В результат включить:  
номер группы;  
номер, фамилию, имя и отчество студента;  
номер и состояние пункта приказа;  
Для реализации использовать подзапрос с EXISTS.

SELECT Уч.ГРУППА,

Уч.ИД,

Н\_ЛЮДИ.ФАМИЛИЯ,

Н\_ЛЮДИ.ИМЯ,

Н\_ЛЮДИ.ОТЧЕСТВО,

Уч.П\_ПРКОК\_ИД

FROM Н\_УЧЕНИКИ as Уч

JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Уч.ЧЛВК\_ИД

JOIN Н\_ПЛАНЫ ON Уч.ПЛАН\_ИД = Н\_ПЛАНЫ.ИД

JOIN Н\_ФОРМЫ\_ОБУЧЕНИЯ ON Н\_ПЛАНЫ.ФО\_ИД = Н\_ФОРМЫ\_ОБУЧЕНИЯ.ИД

AND (Н\_ФОРМЫ\_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Очная')

JOIN Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ ON Н\_ПЛАНЫ.НАПС\_ИД = Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ.ИД

JOIN Н\_НАПР\_СПЕЦ ON Н\_НАПР\_СПЕЦ.ИД = Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ.НС\_ИД

AND Н\_НАПР\_СПЕЦ.НАИМЕНОВАНИЕ = 'Программная инженерия'

WHERE EXISTS (

SELECT \*

FROM Н\_УЧЕНИКИ as Уч2

WHERE Уч2.ПРИЗНАК = 'обучен'

AND Уч2.СОСТОЯНИЕ = 'утвержден'

AND Уч2.ИД = Уч.ИД

AND Уч2.КОНЕЦ > '2018-09-01'

);

* 1. Решение:

Сформировать запрос для получения числа в группе No 3100 отличников.

SELECT COUNT(\*) as "Отличники"

FROM Н\_УЧЕНИКИ Уч

INNER JOIN Н\_ВЕДОМОСТИ ON Уч.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

WHERE Уч.ГРУППА = '3100' AND Н\_ВЕДОМОСТИ.ОЦЕНКА= '5';

1. Выводы

Эта лабораторная работа помогла мне разобраться с различными запросами. Я разобрался с синтаксисом PostgreSQL, понял, как работает JOIN, научился сортировать различные данные. Также я узнал, как работать с базами данных через терминал

1. Примеры cube,rollup,grouping sets:
2. ПРИМЕР CUBE:

Select Ч.ИМЯ, Ч.ФАМИЛИЯ, В.ОЦЕНКА, В.ДАТА, В.СОСТОЯНИЕ from Н\_ЛЮДИ Ч INNER JOIN Н\_ВЕДОМОСТИ В ON Ч.ИД = В.ЧЛВК\_ИД

Where В.ИД BETWEEN 40500 AND 40550

GROUP by cube(Ч.ИМЯ,Ч.ФАМИЛИЯ,В.ОЦЕНКА,В.ДАТА,В.СОСТОЯНИЕ);

И

SELECT В.ИД,В.НОМЕР\_ДОКУМЕНТА,В.ДАТА,В.ОЦЕНКА,ТВ.НАИМЕНОВАНИЕ,ТВ.КТО\_СОЗДАЛ FROM Н\_ВЕДОМОСТИ В

INNER JOIN Н\_ТИПЫ\_ВЕДОМОСТЕЙ ТВ ON В.ТВ\_ИД = ТВ.ИД

WHERE В.ИД BETWEEN 40500 AND 40550

GROUP BY CUBE(В.ИД,В.НОМЕР\_ДОКУМЕНТА,В.ДАТА,В.ОЦЕНКА,ТВ.НАИМЕНОВАНИЕ,ТВ.КТО\_СОЗДАЛ);

1. ПРИМЕР ROLLUP

SELECT В.ИД,В.НОМЕР\_ДОКУМЕНТА,В.ДАТА,В.ОЦЕНКА,ТВ.НАИМЕНОВАНИЕ,ТВ.КТО\_СОЗДАЛ FROM Н\_ВЕДОМОСТИ В

INNER JOIN Н\_ТИПЫ\_ВЕДОМОСТЕЙ ТВ ON В.ТВ\_ИД = ТВ.ИД

WHERE В.ИД BETWEEN 40500 AND 40550

GROUP BY ROLLUP(В.ИД,В.НОМЕР\_ДОКУМЕНТА,В.ДАТА,В.ОЦЕНКА,ТВ.НАИМЕНОВАНИЕ,ТВ.КТО\_СОЗДАЛ);

И

Select Ч.ИМЯ, Ч.ФАМИЛИЯ, В.ОЦЕНКА, В.ДАТА, В.СОСТОЯНИЕ from Н\_ЛЮДИ Ч

INNER JOIN Н\_ВЕДОМОСТИ В ON Ч.ИД = В.ЧЛВК\_ИД

WHERE В.ИД BETWEEN 40500 AND 40550

GROUP by ROLLUP(Ч.ИМЯ,Ч.ФАМИЛИЯ,В.ОЦЕНКА,В.ДАТА,В.СОСТОЯНИЕ);

1. ПРИМЕР GROUPING SETS

SELECT В.ИД,В.НОМЕР\_ДОКУМЕНТА,В.ДАТА,В.ОЦЕНКА,ТВ.НАИМЕНОВАНИЕ,ТВ.КТО\_СОЗДАЛ

FROM Н\_ВЕДОМОСТИ В INNER JOIN Н\_ТИПЫ\_ВЕДОМОСТЕЙ ТВ ON В.ТВ\_ИД = ТВ.ИД

WHERE В.ИД BETWEEN 40500 AND 40550

GROUP BY GROUPING SETS((В.ИД,В.ДАТА),(В.ИД,В.НОМЕР\_ДОКУМЕНТА ,В.ОЦЕНКА),(В.ИД,ТВ.НАИМЕНОВАНИЕ,ТВ.КТО\_СОЗДАЛ));

И

Select Ч.ИМЯ, Ч.ФАМИЛИЯ, В.ОЦЕНКА, В.ДАТА, В.СОСТОЯНИЕ from Н\_ЛЮДИ Ч INNER JOIN Н\_ВЕДОМОСТИ В ON Ч.ИД = В.ЧЛВК\_ИД

Where В.ИД BETWEEN 40500 AND 40550

GROUP by GROUPING SETS((Ч.ИМЯ,В.ОЦЕНКА),(Ч.ИМЯ,Ч.ФАМИЛИЯ,В.СОСТОЯНИЕ),(В.ОЦЕНКА,В.ДАТА,В.СОСТОЯНИЕ));

1. Допзадания:
2. Сделать таблицу "группа, фио, средняя оценка, ранг (как средняя оценка в рамках группы)"

SELECT

CONCAT(H.ФАМИЛИЯ, ' ',H.ИМЯ, ' ', H.ОТЧЕСТВО) AS ФИО,

Уч.ГРУППА,

Ср\_оценка\_группы,

AVG(CASE

WHEN Вед.ОЦЕНКА = 'осв' THEN 0

WHEN Вед.ОЦЕНКА = 'неявка' THEN 0

WHEN Вед.ОЦЕНКА = 'зачет' THEN 5

WHEN Вед.ОЦЕНКА = 'незач' THEN 0

WHEN Вед.ОЦЕНКА = '99' THEN 6

ELSE CAST(Вед.ОЦЕНКА AS INT)

END) AS Ср\_оценка,

round(AVG(CASE

WHEN Вед.ОЦЕНКА = 'осв' THEN 0

WHEN Вед.ОЦЕНКА = 'неявка' THEN 0

WHEN Вед.ОЦЕНКА = 'зачет' THEN 5

WHEN Вед.ОЦЕНКА = 'незач' THEN 0

WHEN Вед.ОЦЕНКА = '99' THEN 6

ELSE CAST(Вед.ОЦЕНКА AS INT)

END) / Ср\_оц\_гр.Ср\_оценка\_группы\*10)-1 AS Ранг

FROM

Н\_ЛЮДИ H

JOIN Н\_УЧЕНИКИ Уч ON Уч.ЧЛВК\_ИД = H.ИД

JOIN Н\_ВЕДОМОСТИ Вед ON Вед.ЧЛВК\_ИД = H.ИД

JOIN (

SELECT

AVG(CASE

WHEN Вед.ОЦЕНКА = 'осв' THEN 0

WHEN Вед.ОЦЕНКА = 'неявка' THEN 0

WHEN Вед.ОЦЕНКА = 'зачет' THEN 5

WHEN Вед.ОЦЕНКА = 'незач' THEN 0

WHEN Вед.ОЦЕНКА = '99' THEN 6

ELSE CAST(Вед.ОЦЕНКА AS INT)

END) AS Ср\_оценка\_группы

FROM

Н\_УЧЕНИКИ Уч

JOIN Н\_ВЕДОМОСТИ Вед ON Вед.ЧЛВК\_ИД = Уч.ЧЛВК\_ИД

) AS Ср\_оц\_гр ON 1 = 1

GROUP BY

CONCAT(H.ФАМИЛИЯ, ' ',H.ИМЯ, ' ', H.ОТЧЕСТВО),

Уч.ГРУППА,

Ср\_оц\_гр.Ср\_оценка\_группы

ORDER BY Ср\_оценка DESC;

1. Запрос, выводящий группа-фио-дисц-оценка-накопительная сумма.

WITH PeopleAndScores AS (

SELECT

У.ЧЛВК\_ИД AS Person\_ID,

У.ГРУППА AS Group\_Number,

Л.ФАМИЛИЯ || ' ' || Л.ИМЯ || ' ' || Л.ОТЧЕСТВО AS Full\_Name,

С.ДИС\_ИД AS Discipline\_ID,

В.ДАТА AS Exam\_Date,

CASE

WHEN В.ОЦЕНКА = 'осв' THEN 0

WHEN В.ОЦЕНКА = 'неявка' THEN 0

WHEN В.ОЦЕНКА = 'зачет' THEN 5

WHEN В.ОЦЕНКА = 'незач' THEN 0

WHEN В.ОЦЕНКА = '99' THEN 6

ELSE CAST(В.ОЦЕНКА AS INT)

END AS Score

FROM Н\_УЧЕНИКИ У

JOIN Н\_ЛЮДИ Л ON У.ЧЛВК\_ИД = Л.ИД

JOIN Н\_ВЕДОМОСТИ В ON У.ЧЛВК\_ИД = В.ЧЛВК\_ИД

JOIN Н\_СТРОКИ\_ПЛАНОВ С ON У.ПЛАН\_ИД = С.ПЛАН\_ИД

),

Disciplines AS (

SELECT ИД AS Discipline\_ID, КОРОТКОЕ\_ИМЯ AS Discipline\_Name

FROM Н\_ДИСЦИПЛИНЫ

),

OrderedScores AS (

SELECT

P.Group\_Number,

P.Full\_Name,

D.Discipline\_Name,

P.Score

FROM PeopleAndScores P

JOIN Disciplines D ON P.Discipline\_ID = D.Discipline\_ID

ORDER BY P.Group\_Number, P.Full\_Name, D.Discipline\_Name, P.Score

),

ScoresWithCumulativeSum AS (

SELECT

O.Group\_Number,

O.Full\_Name,

O.Discipline\_Name,

O.Score,

SUM(O.Score) OVER (PARTITION BY O.Group\_Number, O.Full\_Name, O.Discipline\_Name ORDER BY O.Score ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS Cumulative\_Sum

FROM OrderedScores O

)

SELECT \* FROM ScoresWithCumulativeSum

WHERE Cumulative\_Sum > 0

LIMIT 10;

1. Запрос, который выведет иерархию университета(факультет,кафедра)

WITH

Faculty AS (

SELECT Н\_ОТДЕЛЫ.ИД AS Faculty\_ID, Н\_ОТДЕЛЫ.КОРОТКОЕ\_ИМЯ AS Faculty\_Name

FROM Н\_ОТДЕЛЫ

JOIN Н\_ХАРАКТЕРИСТИКИ\_ОТДЕЛОВ ON Н\_ОТДЕЛЫ.ИД = Н\_ХАРАКТЕРИСТИКИ\_ОТДЕЛОВ.ОТД\_ИД

JOIN Н\_СВОЙСТВА\_ОТДЕЛОВ ON Н\_ХАРАКТЕРИСТИКИ\_ОТДЕЛОВ.СВОТД\_ИД = Н\_СВОЙСТВА\_ОТДЕЛОВ.ИД

WHERE Н\_СВОЙСТВА\_ОТДЕЛОВ.НАИМЕНОВАНИЕ = 'Факультет'

),

Department AS (

SELECT Н\_ОТДЕЛЫ.ОТД\_ИД AS Department\_Faculty\_ID, Н\_ОТДЕЛЫ.КОРОТКОЕ\_ИМЯ AS Department\_Name

FROM Н\_ОТДЕЛЫ

JOIN Н\_ХАРАКТЕРИСТИКИ\_ОТДЕЛОВ ON Н\_ОТДЕЛЫ.ИД = Н\_ХАРАКТЕРИСТИКИ\_ОТДЕЛОВ.ОТД\_ИД

JOIN Н\_СВОЙСТВА\_ОТДЕЛОВ ON Н\_ХАРАКТЕРИСТИКИ\_ОТДЕЛОВ.СВОТД\_ИД = Н\_СВОЙСТВА\_ОТДЕЛОВ.ИД

WHERE Н\_СВОЙСТВА\_ОТДЕЛОВ.НАИМЕНОВАНИЕ = 'Кафедра'

)

SELECT Name

FROM (

SELECT Faculty.Faculty\_ID, Faculty.Faculty\_Name AS Name, 1 as OrderRank

FROM Faculty

UNION ALL

SELECT Department.Department\_Faculty\_ID, lpad(Department.Department\_Name, length(Department.Department\_Name) + 3, ' ') AS Name, 2 as OrderRank

FROM Department

) t

ORDER BY Faculty\_ID, OrderRank, Name;

1. Получить список студентов, зачисленных в дату, ближайшую к 1 сент 2012 году(в будущее и в прошлое(смортрим растояние для юближайшего и выбираем его(или, если равны, то обе берем))) на первый курс очной формы обучения (специальность: Программная инженерия). В результат включить:  
   номер группы;  
   номер, фамилию, имя и отчество студента;  
   номер и состояние пункта приказа;   
   Для реализации использовать подзапрос с EXISTS.

SELECT Уч.ГРУППА,

Уч.ИД,

Н\_ЛЮДИ.ФАМИЛИЯ,

Н\_ЛЮДИ.ИМЯ,

Н\_ЛЮДИ.ОТЧЕСТВО,

Уч.П\_ПРКОК\_ИД,

Уч.КОНЕЦ as ДАТА\_ЗАЧИСЛЕНИЯ

FROM Н\_УЧЕНИКИ as Уч

JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Уч.ЧЛВК\_ИД

JOIN Н\_ПЛАНЫ ON Уч.ПЛАН\_ИД = Н\_ПЛАНЫ.ИД

JOIN Н\_ФОРМЫ\_ОБУЧЕНИЯ ON Н\_ПЛАНЫ.ФО\_ИД = Н\_ФОРМЫ\_ОБУЧЕНИЯ.ИД

AND (Н\_ФОРМЫ\_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Очная')

JOIN Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ ON Н\_ПЛАНЫ.НАПС\_ИД = Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ.ИД

JOIN Н\_НАПР\_СПЕЦ ON Н\_НАПР\_СПЕЦ.ИД = Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ.НС\_ИД

AND Н\_НАПР\_СПЕЦ.НАИМЕНОВАНИЕ = 'Программная инженерия'

WHERE EXISTS (

SELECT 1

FROM Н\_УЧЕНИКИ as Уч2

WHERE Уч2.ПРИЗНАК = 'обучен'

AND Уч2.СОСТОЯНИЕ = 'утвержден'

AND Уч2.ИД = Уч.ИД

AND ABS(EXTRACT(DAY FROM (Уч2.КОНЕЦ - DATE '2012-09-01'))) =

(

SELECT MIN(ABS(EXTRACT(DAY FROM (Уч3.КОНЕЦ - DATE '2012-09-01'))))

FROM Н\_УЧЕНИКИ as Уч3

WHERE Уч3.ПРИЗНАК = 'обучен'

AND Уч3.СОСТОЯНИЕ = 'утвержден'

)

);