HashAggregate (cost=2241.03..2277.16 rows=1606 width=523) (actual time=18.605..18.627 rows=27 loops=1)

Group Key: concat\_ws(' '::text, «Н\_ЛЮДИ».«ФАМИЛИЯ», «Н\_ЛЮДИ».«ИМЯ», «Н\_ЛЮДИ».«ОТЧЕСТВО»), «Н\_ОТДЕЛЫ».«ИМЯ\_В\_ИМИН\_ПАДЕЖЕ», «Н\_УЧЕНИКИ».«ПРИЗНАК», «Н\_ПЛАНЫ».«КУРС», «Н\_УЧЕНИКИ».«НАЧАЛО», CASE WHEN ((«Н\_КВАЛИФИКАЦИИ».«НАИМЕНОВАНИЕ»)::text ~~ 'Бакалавр%'::text) THEN 'Бакалавр'::text WHEN ((«Н\_КВАЛИФИКАЦИИ».«НАИМЕНОВАНИЕ»)::text ~~ 'Магистр%'::text) THEN 'Магистр'::text ELSE NULL::text END

* Это операция агрегации по хэшу, используемая для группировки данных.
* "Group Key" указывает столбцы, по которым происходит группировка.
* В данном случае, группировка происходит по конкатенации значений столбцов "Н\_ЛЮДИ"."ФАМИЛИЯ", "Н\_ЛЮДИ"."ИМЯ", "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_ОТДЕЛЫ"."ИМЯ\_В\_ИМИН\_ПАДЕЖЕ", "Н\_УЧЕНИКИ"."ПРИЗНАК", "Н\_ПЛАНЫ"."КУРС", "Н\_УЧЕНИКИ"."НАЧАЛО" и выражения CASE.

Batches: 1 Memory Usage: 89kB

* Это информация о выполнении операции.
* "Batches" указывает количество пакетов данных, использованных при выполнении операции.
* "Memory Usage" указывает количество используемой памяти.

-> Nested Loop (cost=2.71..2204.89 rows=1606 width=513) (actual time=4.203..17.281 rows=1924 loops=1)

* Это операция вложенного цикла.
* (cost=2.71..2204.89): Это оценка стоимости выполнения операции. Значение "2.71" представляет оценку стоимости выполнения операции в самом оптимистичном сценарии, а "2204.89 " - в наихудшем сценарии. Стоимость измеряется в условных единицах и используется оптимизатором запросов для выбора наиболее эффективного плана выполнения.
* (actual time=4.203..17.281): Это фактическое время выполнения операции. Значение "4.203" указывает на минимальное время выполнения, а "17.281 " - на максимальное время выполнения операции.

-> Nested Loop (cost=2.42..1679.47 rows=168 width=918) (actual time=4.189..14.928 rows=27 loops=1)

-> Nested Loop Semi Join (cost=2.13..1625.00 rows=168 width=873) (actual time=4.180..14.834 rows=27 loops=1)

* Это полу-соединение вложенных циклов.
* Оно используется для объединения двух наборов данных, где только соответствующие строки из внешнего набора остаются в результирующем наборе данных.

-> Nested Loop (cost=1.84..219.08 rows=1196 width=869) (actual time=0.090..6.253 rows=6694 loops=1)

-> Nested Loop (cost=1.55..15.65 rows=41 width=844) (actual time=0.078..0.644 rows=333 loops=1)

-> Nested Loop (cost=1.40..13.91 rows=41 width=430) (actual time=0.070..0.439 rows=333 loops=1)

-> Hash Join (cost=1.25..6.26 rows=9 width=422) (actual time=0.054..0.122 rows=64 loops=1)

Hash Cond: («Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ».«КВАЛ\_ИД» = "Н\_КВАЛИФИКАЦИИ».«ИД»)

* Это операция соединения по хэшу.
* Она используется для объединения двух наборов данных на основе значения хэша.
* В данном случае соединение происходит между таблицами "Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ" и "Н\_КВАЛИФИКАЦИИ" по столбцам "КВАЛ\_ИД" и "ИД" соответственно.

-> Seq Scan on «Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ» (cost=0.00..4.51 rows=151 width=8) (actual time=0.018..0.038 rows=151 loops=1)

* + Это операция последовательного сканирования таблицы.
  + Она просматривает все строки таблицы последовательно для выполнения операции.

-> Hash (cost=1.24..1.24 rows=1 width=422) (actual time=0.022..0.022 rows=9 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 9kB

* + Это операция хэширования.
  + Она используется для создания хэш-таблицы на основе входных данных.
  + В данном случае, создается хэш-таблица для выполнения операции соединения.

-> Seq Scan on «Н\_КВАЛИФИКАЦИИ» (cost=0.00..1.24 rows=1 width=422) (actual time=0.011..0.015 rows=9 loops=1)"

Filter: (((«НАИМЕНОВАНИЕ»)::text ~~ 'Бакалавр%'::text) OR ((«НАИМЕНОВАНИЕ»)::text ~~ 'Магистр%'::text))

Rows Removed by Filter: 7

* Это операция последовательного сканирования таблицы.
* Она просматривает все строки таблицы последовательно для выполнения операции.

-> Index Scan using «ПЛАН\_НАПС\_FK\_I» on «Н\_ПЛАНЫ» (cost=0.15..0.74 rows=11 width=16) (actual time=0.001..0.004 rows=5 loops=64)

Index Cond: («НАПС\_ИД» = «Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ».«ИД»)

* Это операция сканирования индекса.
* Она используется для поиска данных с использованием индекса.
* В данном случае, используется индекс "ПЛАН\_НАПС\_FK\_I" на таблице "Н\_ПЛАНЫ".

-> Memoize (cost=0.15..0.20 rows=1 width=422) (actual time=0.000..0.000 rows=1 loops=333)

Cache Key: «Н\_ПЛАНЫ».«ОТД\_ИД»

Cache Mode: logical

Hits: 331 Misses: 2 Evictions: 0 Overflows: 0 Memory Usage: 1kB

* Это операция кэширования.
* Она используется для сохранения промежуточных результатов в памяти для повторного использования.
* В данном случае, кэшируются данные из таблицы "Н\_ОТДЕЛЫ".

-> Index Scan using «ОТД\_PK» on «Н\_ОТДЕЛЫ» (cost=0.14..0.19 rows=1 width=422) (actual time=0.003..0.003 rows=1 loops=2)

Index Cond: («ИД» = «Н\_ПЛАНЫ».«ОТД\_ИД»)

-> Index Scan using «УЧЕН\_ПЛАН\_FK\_I» on «Н\_УЧЕНИКИ» (cost=0.29..4.62 rows=34 width=33) (actual time=0.002..0.013 rows=20 loops=333)

Index Cond: («ПЛАН\_ИД» = «Н\_ПЛАНЫ».«ИД»)

Filter: (((«ПРИЗНАК»)::text = 'обучен'::text) OR ((«ПРИЗНАК»)::text = 'академ'::text))

Rows Removed by Filter: 5

-> Index Scan using «ВЕД\_ЧЛВК\_FK\_IFK» on «Н\_ВЕДОМОСТИ» «Н\_ВЕДОМОСТИ\_1» (cost=0.29..2.72 rows=3 width=4) (actual time=0.001..0.001 rows=0 loops=6694)

Index Cond: («ЧЛВК\_ИД» = «Н\_УЧЕНИКИ».«ИД»)

Filter: (((«ОЦЕНКА»)::text = 'незач'::text) OR ((«ОЦЕНКА»)::text = '2'::text))

Rows Removed by Filter: 0

-> Index Scan using «ЧЛВК\_PK» on «Н\_ЛЮДИ» (cost=0.28..0.32 rows=1 width=53) (actual time=0.003..0.003 rows=1 loops=27)

Index Cond: («ИД» = «Н\_УЧЕНИКИ».«ЧЛВК\_ИД»)

-> Index Scan using «ВЕД\_ЧЛВК\_FK\_IFK» on «Н\_ВЕДОМОСТИ» (cost=0.29..2.38 rows=68 width=10) (actual time=0.002..0.045 rows=71 loops=27)

Index Cond: («ЧЛВК\_ИД» = «Н\_УЧЕНИКИ».«ИД»)

Planning Time: 4.313 ms

Execution Time: 18.783 ms

EXPLAIN ANALYSE SELECT "Н\_ЛЮДИ"."ФАМИЛИЯ", "Н\_ВЕДОМОСТИ"."ДАТА"  
FROM "Н\_ЛЮДИ"  
RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ИД" > 100012

* Seq Scan "Н\_ЛЮДИ": Производится полное сканирование таблицы для получения всех строк.
* Seq Scan "Н\_ВЕДОМОСТИ": Производится полное сканирование таблицы "Н\_ВЕДОМОСТИ" для получения всех строк.
* Nested Loop: Для каждой строки из первого сканирования (таблицы "Н\_ЛЮДИ") производится поиск соответствующей строки во втором сканировании (таблице "Н\_ВЕДОМОСТИ") по условию "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД". Если соответствующие строки найдены, то выбираются столбцы "Н\_ЛЮДИ"."ФАМИЛИЯ" и "Н\_ВЕДОМОСТИ"."ДАТА".
* Filter: После объединения результатов производится фильтрация по условию "Н\_ЛЮДИ"."ИД" > 100012.

Nested Loop (cost=0.58..216.65 rows=43 width=28) (actual time=0.004..0.005 rows=0 loops=1)

-> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.29 rows=1 width=24) (actual time=0.004..0.004 rows=0 loops=1)"

Index Cond: ("ИД" < 100012)"

-> Index Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "Н\_ВЕДОМОСТИ" (cost=0.29..207.67 rows=68 width=12) (never executed)"

Index Cond: ("ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")

Planning Time: 0.315 ms

Execution Time: 0.037 ms

Этот запрос выполняет вложенный цикл со следующими шагами:

* Сначала выполняется сканирование индекса "ЧЛВК\_PK" на таблице "Н\_ЛЮДИ". Условие для сканирования индекса - значение столбца "ИД" < 100012.
* Затем выполняется сканирование индекса "ВЕД\_ЧЛВК\_FK\_IFK" на таблице "Н\_ВЕДОМОСТИ", но этот шаг фактически не выполняется (never executed), поскольку предыдущий шаг не вернул ни одной строки. Это связано с условием соединения двух таблиц, которое требует совпадения значений столбца "ЧЛВК\_ИД" из таблицы "Н\_ВЕДОМОСТИ" со значением столбца "ИД" из таблицы "Н\_ЛЮДИ".

acid, ansis spark