## *Барышников Илья*

## Задача №1

Дано 2 таблицы:

1) employee - справочник сотрудников

2) tasks - задачи, назначенные на сотрудников

Необходимо написать запрос, который покажет всех сотрудников, у кого в работе менее трех задач

Результат предоставить в виде: имя сотрудника, кол-во задач в работе.

drop table if exists #employee

create table #employee

(

id smallint,

emp\_name varchar(255),

rating int

)

insert into #employee values (1,'Vasya',5)

insert into #employee values (2,'Vasya',3)

insert into #employee values (3,'Petya',9)

insert into #employee values (4,'Igor',4)

insert into #employee values (5,'Nikita',2)

drop table if exists #tasks

create table #tasks

(

id smallint,

assignee\_id smallint,

rating smallint

)

insert into #tasks values (1,1,5)

insert into #tasks values (2,1,4)

insert into #tasks values (3,2,2)

insert into #tasks values (4,5,5)

insert into #tasks values (5,5,1)

insert into #tasks values (6,5,7)

insert into #tasks values (7,5,9)

insert into #tasks values (8,5,3)

insert into #tasks values (9,5,3)

insert into #tasks values (10,5,5)

insert into #tasks values (11,2,7)

insert into #tasks values (12,3,8)

insert into #tasks values (13,3,8)

insert into #tasks values (14,2,7)

insert into #tasks values (15,3,9)

insert into #tasks values (16,5,4)

insert into #tasks values (17,1,5)

**Запрос:**

SELECT

emp\_name,

count(tsk.id) AS number\_of\_tasks

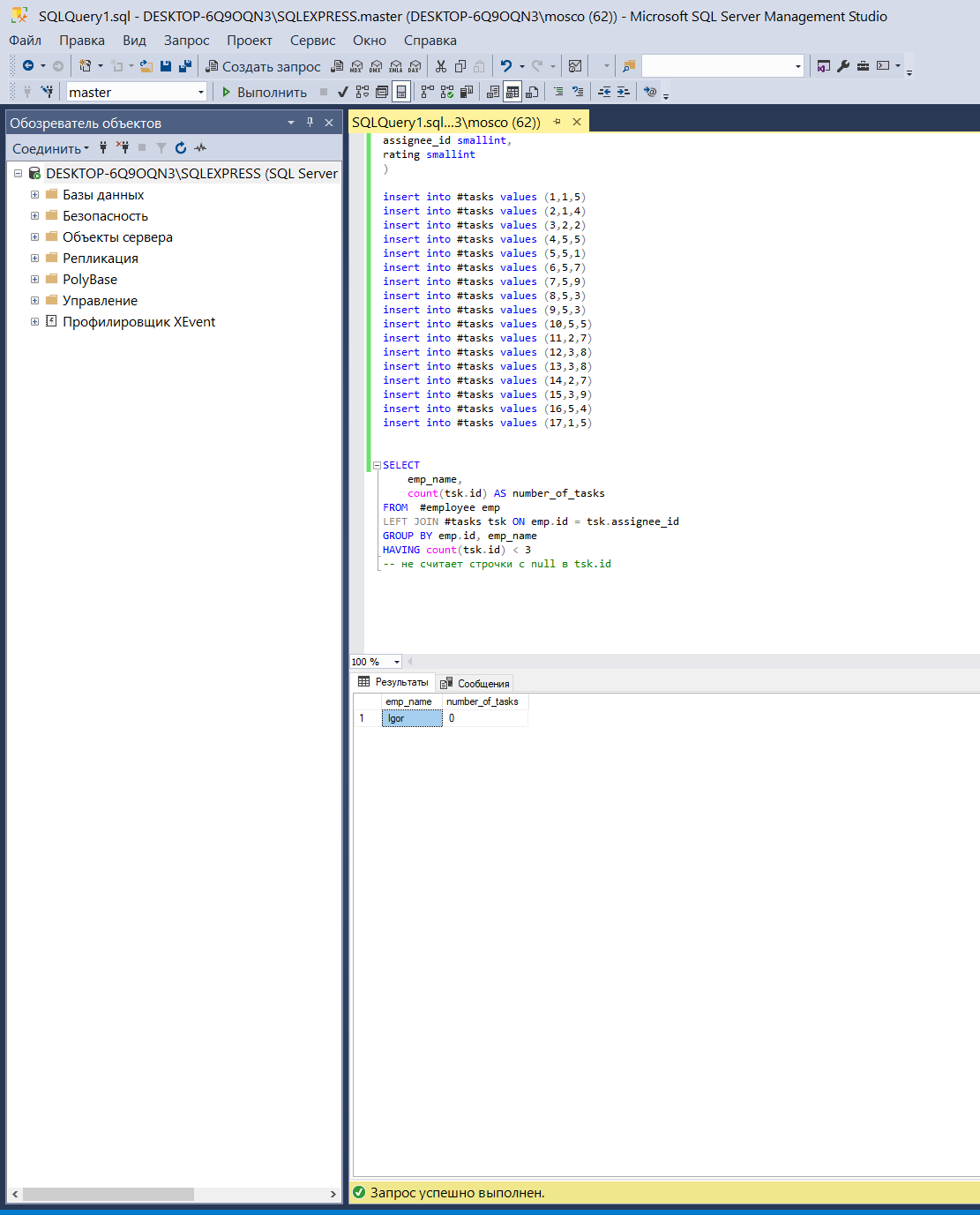
FROM #employee emp

LEFT JOIN #tasks tsk ON emp.id = tsk.assignee\_id

GROUP BY emp.id, emp\_name

HAVING count(tsk.id) < 3

-- не считает строчки с null в tsk.id



**Небольшое пояснение:**

Запрос выше даёт ответ *строго* в соответствии с условием.

Группировка по двум полям не позволила «попасться» на одинаковых именах сотрудников, а функция count(tasks.id) посчитала строки с ненулевыми значениями поля tasks.id (соответственно, и left join был нужен, чтобы была строчка с сотрудником без задач, но с пустыми полями из таблицы tasks).

Однако в случае, если бы два сотрудника с одинаковыми именами попали бы в результирующую таблицу, было бы корректнее или выводить дополнительно их идентификаторы, или формировать поле, где было указано и имя, и идентификаторы (с помощью конкатенации). Например:

SELECT

CAST(emp\_name AS char(4)) + ' (id = ' + CAST(emp.id AS char(4)) + ')' AS emp\_name\_id,

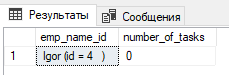
count(tsk.id) AS number\_of\_tasks

FROM #employee emp

LEFT JOIN #tasks tsk ON emp.id = tsk.assignee\_id

GROUP BY emp.id, emp\_name

HAVING count(tsk.id) < 3



## Задача №2

Вам предоставлена выборка данных, в которой содержится информация по заявкам (строчка в таблице) за определенный период времени. (см. прикрепленный к архиву excel файл “Test sample”)

В предоставленный промежуток времени произошло ухудшение конверсии к completed заявкам, что негативно повлияло на выручку компании.

Необходимо разобраться чем обусловлено такое падение и найти «корень проблемы».

Для выполнения задачи необходимо всю выборку сначала перенести в базу данных (представьте, что вы работаете с реальной SQL таблицей) и затем уже работать с SQL таблицей.

Также постройте отчет в Power BI и визуально отобразите следующее:

* Иерархиями отобразите количество заявок в разрезе дней и недель
* Метрику Completed rate
* Таблицу (матрицу) в разрезе дней где будут иерархией отображены отношения up\_stage, mid\_stage, page\_status, sub\_stage к количеству заявок (см. скрин ниже как пример отображения)

Graphical user interface

Description automatically generated with medium confidence

**Решение (изложено подробно, предположение и его проверка начинаются с п.6).**

**(ряд пунктов анализа продублирован в Python – файл Test\_task\_python.ipynb)**

*ПРИМЕЧАНИЕ НА БУДУЩЕЕ: не забывать считать относительные цифры (например, кол-во посещенных страниц ото всех)*

1. Рассчитаем конверсию completed\_rate (отношение заявок с is\_completed к общему числу заявок (строк)) на каждый день. Таким образом, резкое ухудшение конверсии происходило с 20 по 22 мая 2022.

SELECT \*, round(CAST(completed AS float) / CAST(all\_req AS float), 4) AS completed\_rate

FROM

(SELECT ts.rep\_date, ts\_1.completed, count(\*) AS all\_req

FROM Test\_sample ts

INNER JOIN (SELECT rep\_date, count(\*) AS completed

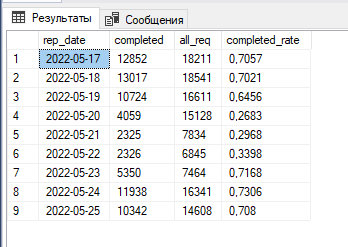
FROM Test\_sample

WHERE is\_completed = 1

GROUP BY rep\_date) ts\_1 ON ts.rep\_date = ts\_1.rep\_date

GROUP BY ts.rep\_date, ts\_1.completed) tss

ORDER BY rep\_date ASC



Оптимизированная версия запроса:

SELECT

ts.rep\_date,

ts\_1.completed,

count(\*) AS all\_req,

round(CAST(ts\_1.completed AS float) / CAST(count(\*) AS float), 4) AS completed\_rate

FROM Test\_sample ts

INNER JOIN (SELECT rep\_date, count(\*) AS completed

FROM Test\_sample

WHERE is\_completed = 1

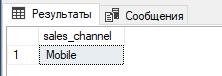
GROUP BY rep\_date) ts\_1 ON ts.rep\_date = ts\_1.rep\_date

GROUP BY ts.rep\_date, ts\_1.completed

ORDER BY ts.rep\_date

1. Поиск причины ухудшения конверсии осуществляем по принципу декомпозиции: деления (группировки) заявок по разным признакам и расчёту конверсии в категориях. Поиск по глобальному каналу продаж (sales\_channel) не имеет смысла, т.к. он единственный – Mobile.

SELECT DISTINCT sales\_channel FROM Test\_sample



1. Далее посмотрим конверсию по основным категориям трафика (traf\_cat\_big) на каждый день. Видим резкое ухудшение 20-22.05 по всем главным категориям – Affiliate (продвижение с оплатой за действие), Contextual (контекстный трафик), Organic (органический).

SELECT

ts.rep\_date,

ts.traf\_cat\_big,

ts\_1.completed,

count(\*) AS all\_req,

round(CAST(ts\_1.completed AS float) / CAST(count(\*) AS float), 4) AS completed\_rate

FROM Test\_sample ts

INNER JOIN (SELECT rep\_date, traf\_cat\_big, count(\*) AS completed

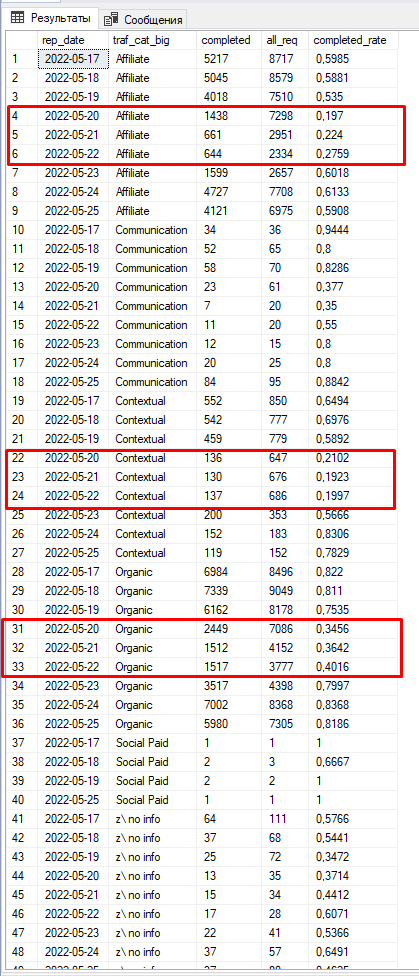
FROM Test\_sample

WHERE is\_completed = 1

GROUP BY rep\_date, traf\_cat\_big) ts\_1 ON ts.rep\_date = ts\_1.rep\_date AND ts.traf\_cat\_big = ts\_1.traf\_cat\_big

GROUP BY ts.rep\_date, ts.traf\_cat\_big, ts\_1.completed

ORDER BY ts.traf\_cat\_big



1. Исследуем конверсию по более мелким категориям (источникам) трафика. Результаты те же (на скринах приведены отдельные категории, по которым много заявок) – идёт повсеместное падение.

SELECT

ts.rep\_date,

ts.traf\_cat\_small,

ts\_1.completed,

count(\*) AS all\_req,

round(CAST(ts\_1.completed AS float) / CAST(count(\*) AS float), 4) AS completed\_rate

FROM Test\_sample ts

INNER JOIN (SELECT rep\_date, traf\_cat\_small, count(\*) AS completed

FROM Test\_sample

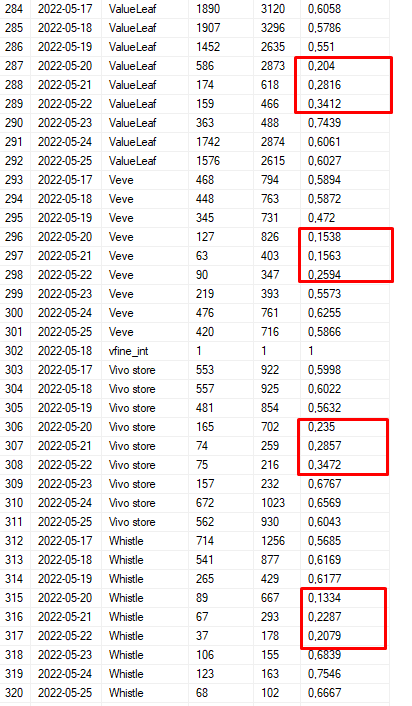
WHERE is\_completed = 1

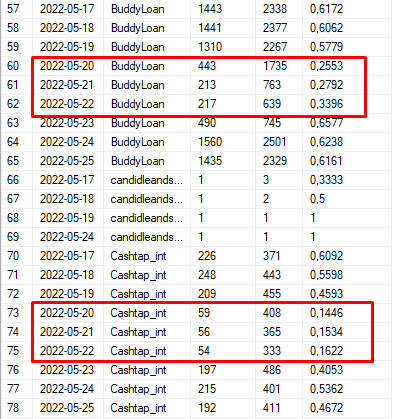
GROUP BY rep\_date, traf\_cat\_small) ts\_1 ON ts.rep\_date = ts\_1.rep\_date AND ts.traf\_cat\_small = ts\_1.traf\_cat\_small

GROUP BY ts.rep\_date, ts.traf\_cat\_small, ts\_1.completed

ORDER BY ts.traf\_cat\_small







1. Далее проверим по полю traf\_cat\_type. Результаты должны получиться аналогичными п.3 с traf\_cat\_big, ведь деление traf\_cat\_type – это деление по принципу бесплатный/платный (т.е. аффилейт-трафик и контекстный – платные, а органический – бесплатный). Предположение подтверждается.

SELECT

ts.rep\_date,

ts.traf\_cat\_type,

ts\_1.completed,

count(\*) AS all\_req,

round(CAST(ts\_1.completed AS float) / CAST(count(\*) AS float), 4) AS completed\_rate

FROM Test\_sample ts

INNER JOIN (SELECT rep\_date, traf\_cat\_type, count(\*) AS completed

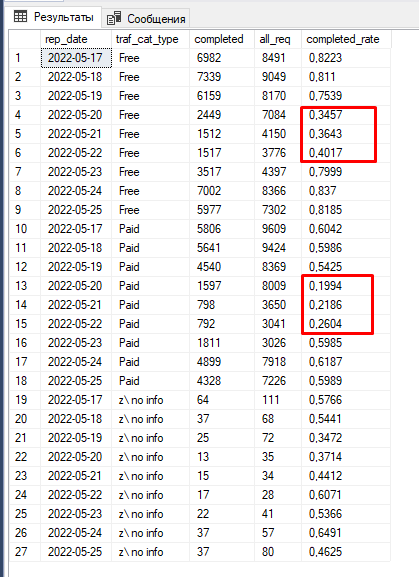
FROM Test\_sample

WHERE is\_completed = 1

GROUP BY rep\_date, traf\_cat\_type) ts\_1 ON ts.rep\_date = ts\_1.rep\_date AND ts.traf\_cat\_type = ts\_1.traf\_cat\_type

GROUP BY ts.rep\_date, ts.traf\_cat\_type, ts\_1.completed

ORDER BY ts.traf\_cat\_type



1. Проверка по полю page\_reached\_CRM («страничка (стадия), достигнутая клиентом»). В данном случае запрос не совсем корректен, т.к. по достижении определенных страниц заявка уже считается заполненной (completed; о чём свидетельствует completed\_rate = 1).

SELECT

ts.rep\_date,

ts.page\_reached\_CRM,

ts\_1.completed,

count(\*) AS all\_req,

round(CAST(ts\_1.completed AS float) / CAST(count(\*) AS float), 4) AS completed\_rate

FROM Test\_sample ts

INNER JOIN (SELECT rep\_date, page\_reached\_CRM, count(\*) AS completed

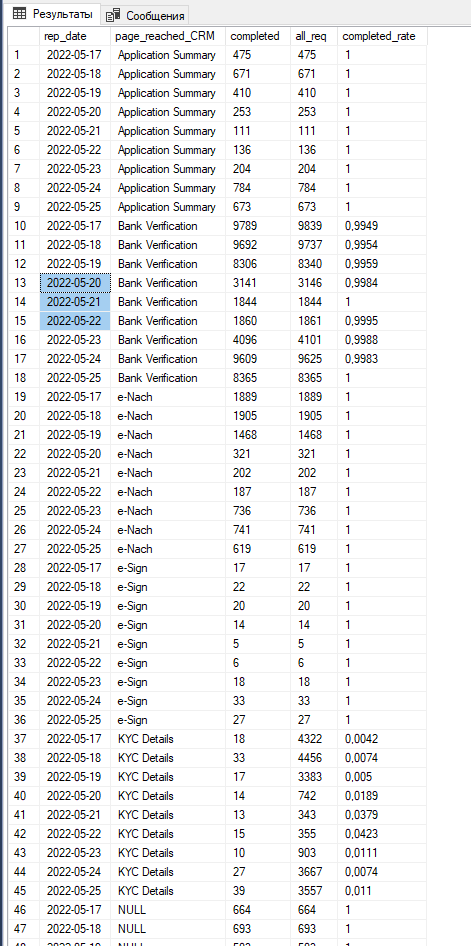
FROM Test\_sample

WHERE is\_completed = 1

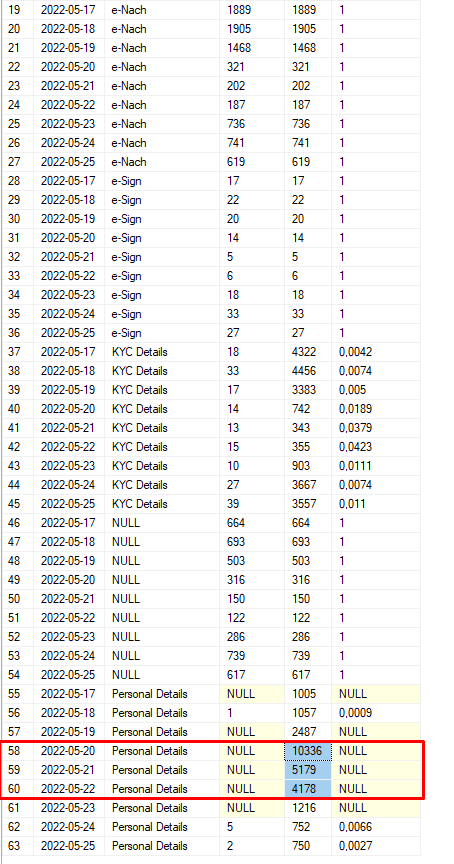
GROUP BY rep\_date, page\_reached\_CRM) ts\_1 ON ts.rep\_date = ts\_1.rep\_date AND ts.page\_reached\_CRM = ts\_1.page\_reached\_CRM

GROUP BY ts.rep\_date, ts.page\_reached\_CRM, ts\_1.completed

ORDER BY ts.page\_reached\_CRM



Также возможно, что по некоторым источникам данные отсутствуют, т.к. у них нет строк с is\_completed=1. В запросе выше изменим INNER JOIN на LEFT JOIN. Результаты во многом аналогичны, однако **появляется аномалия: в день падения рассчитываемой конверсии наблюдается аномалий рост посещений/достижения страницы/раздела с персональными данными (Personal Details)** (скорее всего, на данном этапе не проставляется is\_completed=1, но сам резкий, многократный взлёт позволяет высказать **предположение, что клиенты «останавливались» на странице с персональными данными**).



1. Далее напишем аналогичные запросы с группировкой по стадиям воронки продаж (up\_stage, mid\_stage, а также **группировка по двум полям сразу**) и обратим внимание на стадию введения персональных данных. Стоит обратить внимание на то, что **в то время, как в дни падения конверсии заявки пользователей чаще останавливались на стадии заполнения персональных данных, по остальным страницам/стадиям наблюдается резкое падение** (выделено жёлтым; подобная тенденция прослеживалась и в предыдущих запросах).

В запросе с группировкой по mid\_stage можем увидеть, что в дни падения конверсии пользователи действительно уходили со страницы ввода персональных данных.

SELECT

ts.rep\_date,

ts.up\_stage,

ts\_1.completed,

count(\*) AS all\_req,

round(CAST(ts\_1.completed AS float) / CAST(count(\*) AS float), 4) AS completed\_rate

FROM Test\_sample ts

LEFT JOIN (SELECT rep\_date, up\_stage, count(\*) AS completed

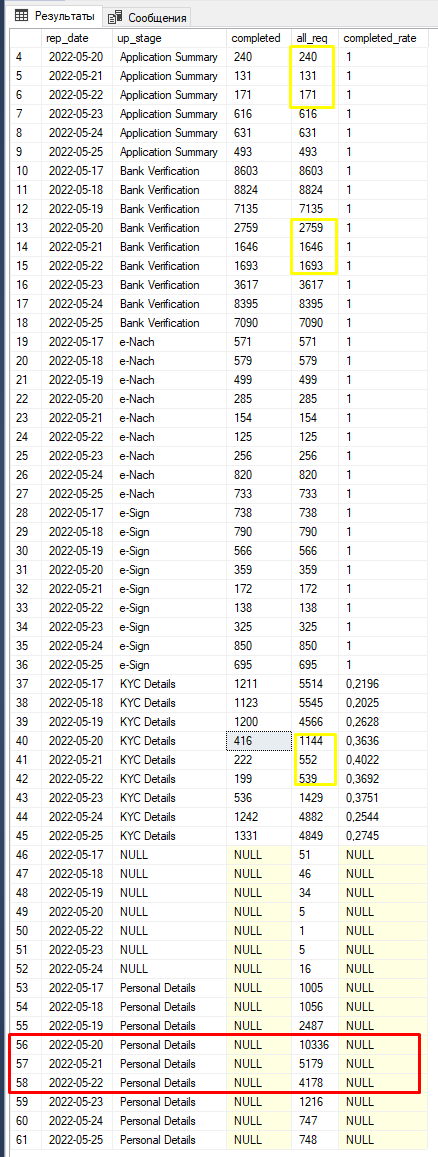
FROM Test\_sample

WHERE is\_completed = 1

GROUP BY rep\_date, up\_stage) ts\_1 ON ts.rep\_date = ts\_1.rep\_date AND ts.up\_stage = ts\_1.up\_stage

GROUP BY ts.rep\_date, ts.up\_stage, ts\_1.completed

ORDER BY ts.up\_stage, ts.rep\_date



SELECT

ts.rep\_date,

ts.mid\_stage,

ts\_1.completed,

count(\*) AS all\_req,

round(CAST(ts\_1.completed AS float) / CAST(count(\*) AS float), 4) AS completed\_rate

FROM Test\_sample ts

LEFT JOIN (SELECT rep\_date, mid\_stage, count(\*) AS completed

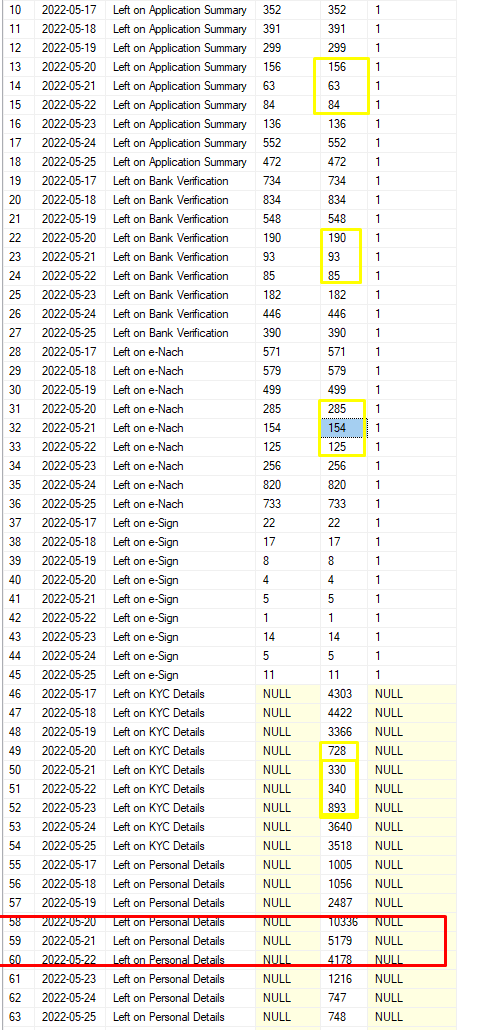
FROM Test\_sample

WHERE is\_completed = 1

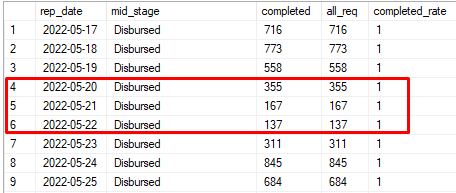
GROUP BY rep\_date, mid\_stage) ts\_1 ON ts.rep\_date = ts\_1.rep\_date AND ts.mid\_stage = ts\_1.mid\_stage

GROUP BY ts.rep\_date, ts.mid\_stage, ts\_1.completed

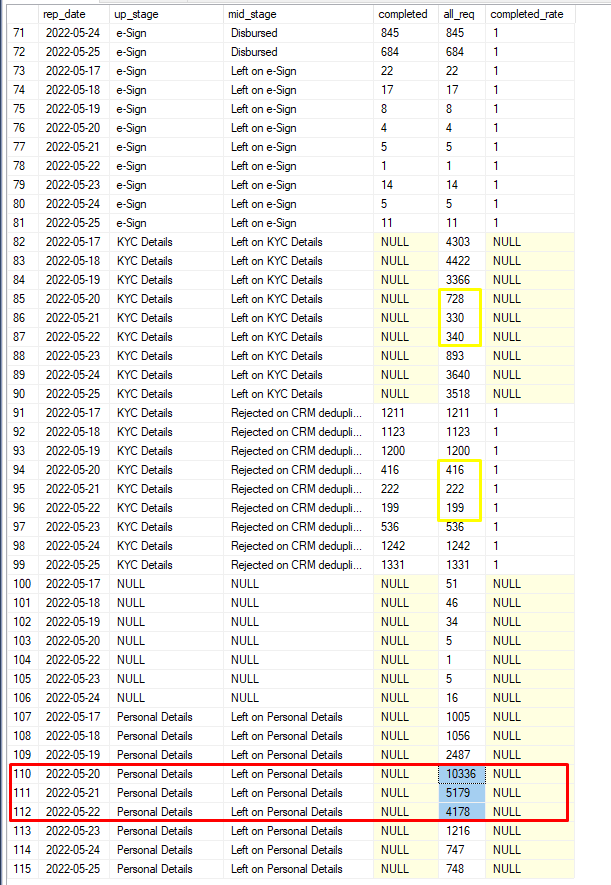
ORDER BY ts.mid\_stage, ts.rep\_date



В этом же запросе *косвенно* можно увидеть, как это повлияло на падение выручки (по падению количества страниц disbursed – “выплачено”, т.е.клиент получил кредит).



Группировка по обоим полям.

SELECT

ts.rep\_date,

ts.up\_stage,

ts.mid\_stage,

ts\_1.completed,

count(\*) AS all\_req,

round(CAST(ts\_1.completed AS float) / CAST(count(\*) AS float), 4) AS completed\_rate

FROM Test\_sample ts

LEFT JOIN (SELECT rep\_date, up\_stage, mid\_stage, count(\*) AS completed

FROM Test\_sample

WHERE is\_completed = 1

GROUP BY rep\_date, up\_stage, mid\_stage) ts\_1 ON ts.rep\_date = ts\_1.rep\_date AND ts.mid\_stage = ts\_1.mid\_stage AND ts.up\_stage = ts\_1.up\_stage

GROUP BY ts.rep\_date, ts.up\_stage, ts.mid\_stage, ts\_1.completed

ORDER BY ts.up\_stage, ts.mid\_stage, ts.rep\_date

1. Перепроверим себя, сгруппировав данные по полю page\_reached\_CRM и page\_status. Изменим сортировку – посмотрим, какое событие каждый день происходило чаще. До и после «сбоя» клиенты, как правило, чаще всего покидали страницу Bank Verification, во время – страницу Personal Details.

SELECT

ts.rep\_date,

ts.page\_reached\_CRM,

ts.page\_status,

ts\_1.completed,

count(\*) AS all\_req,

round(CAST(ts\_1.completed AS float) / CAST(count(\*) AS float), 4) AS completed\_rate

FROM Test\_sample ts

LEFT JOIN (SELECT rep\_date, page\_reached\_CRM, page\_status, count(\*) AS completed

FROM Test\_sample

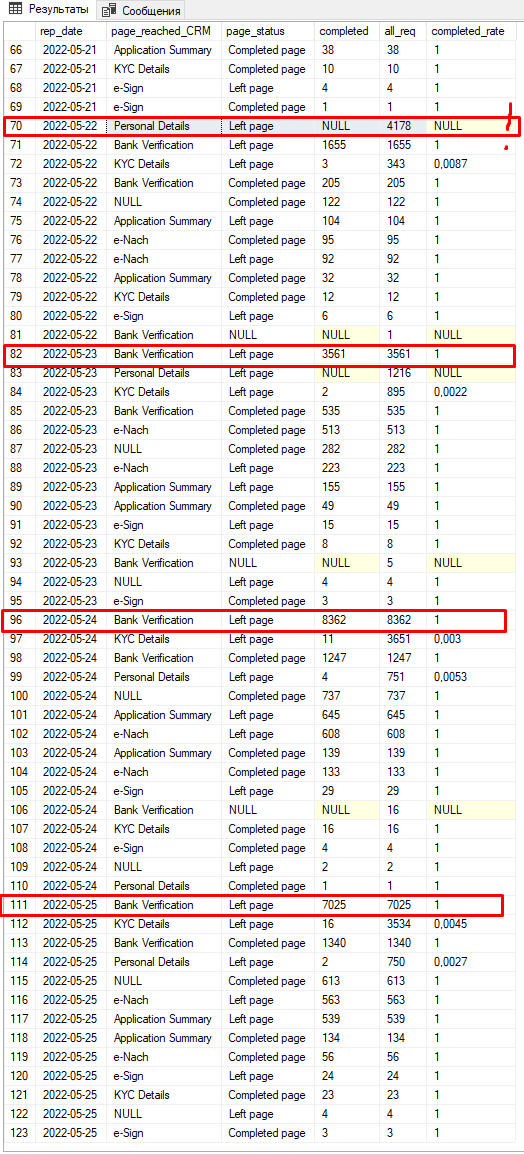
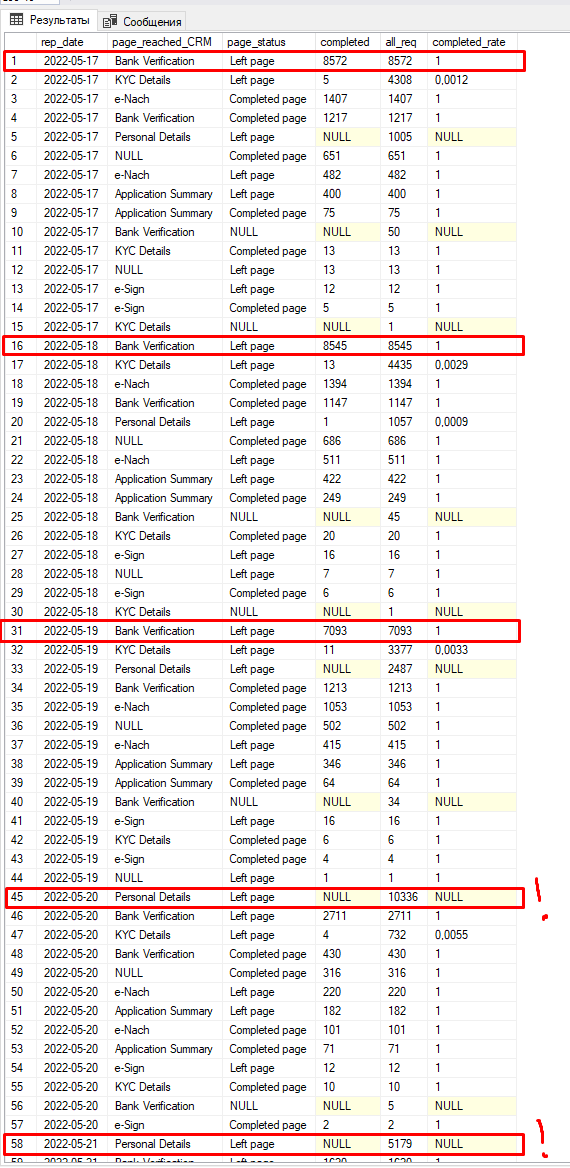
WHERE is\_completed = 1

GROUP BY rep\_date, page\_reached\_CRM, page\_status) ts\_1 ON ts.rep\_date = ts\_1.rep\_date AND ts.page\_status = ts\_1.page\_status AND ts.page\_reached\_CRM = ts\_1.page\_reached\_CRM

GROUP BY ts.rep\_date, ts.page\_reached\_CRM, ts.page\_status, ts\_1.completed

ORDER BY rep\_date, count(\*) DESC

--ORDER BY ts.page\_reached\_CRM, ts.page\_status, ts.rep\_date



**Итог: в период с 20 по 22 мая 2022 наблюдалось резкое падение конверсии по заполненным заявкам и падение выручки из-за бага на странице с вводом персональных данных**. **Пользователи не могли пройти этот этап при получении займа.** Среди возможных причин - некорректное обновление ПО, в котором могла «слететь» вёрстка, некорректно проверяться формат данных, введённых в форме и т.д.