## Задача 1.

При запуске некой акции необходимо информировать клиента по различным каналам коммуникации (email/viber/sms и тд) о старте акции и что нужно сделать, чтобы получить бонус. Например, с 1 июня запускается акция, тогда клиенту будет направлен email со следующим текстом:

“C 1 по 30 июня совершите покупку от 1000 рублей у Партнера X и получите 15% от суммы покупки бонусами”

Каждая коммуникация имеет свой идентификатор. Под идентификатором коммуникации понимается некий уникальный номер, характеризующий время отправки коммуникации, канал и по какой акции совершена коммуникация.

В хранилище данных есть таблица с историей коммуникаций со следующей структурой и таблица с покупками у партнера:

Таблица коммуникаций

|  |  |
| --- | --- |
| Ow\_id | уникальный идентификатор клиента |
| sent\_date | дата коммуникации |
| channel\_nm | канал коммуникации |
| Partner\_id | Уникальный идентификатор партнера |
| Sent | Бинарный флаг. Коммуникация отправлена |
| Delivered | Бинарный флаг. Коммуникация доставлено |
| Opened | Бинарный флаг. Коммуникация открыта |
| Cliked | Бинарный флаг. Переход по ссылке |
| CELL\_PK | Уникальный идентификатор коммуникации. По такому идентификатору можно вытащить всех клиентов, которым была отправлена выбранная коммуникация в определенное время. |

Пример

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ow\_id | sent\_date | channel\_nm | Partner\_id | Sent | Delivered | Opened | Cliked | CELL\_PK |
| 123 | 18.06.2019 | email | 5555 | 1 | 0 | 0 | 0 | 1788 |
| 123 | 20.06.2019 | viber | 1111 | 1 | 1 | 1 | 0 | 1720 |
| 142 | 18.06.2019 | email | 5555 | 1 | 1 | 1 | 1 | 1788 |
| 123 | 18.06.2019 | viber | 5555 | 1 | 1 | 0 | 0 | 1788 |

покупки партнера

|  |  |
| --- | --- |
| Ow\_id | уникальный идентификатор клиента |
| Partner\_id | Уникальный идентификатор партнера |
| POS | Оборот по карте |
| CASH\_DT | Дата покупки |

Пример

|  |  |  |  |
| --- | --- | --- | --- |
| Ow\_id | Partner\_id | POS | CASH\_DT |
| 123 | 5555 | 1001 | 17.06.2019 |
| 123 | 1111 | 200 | 25.06.2019 |
| 142 | 1111 | 500 | 16.06.2019 |
| 123 | 5555 | 800 | 17.06.2019 |
| 123 | 1111 | 600 | 17.06.2019 |
| 142 | 5555 | 700 | 29.06.2009 |

Необходимо посчитать для каждой коммуникации (CELL\_PK):

* скольким клиентам отправили коммуникации, скольким клиентам была доставлена коммуникация, сколько клиентов открыли коммуникацию, сколько клиентов перешли по ссылке
* сколько клиентов, из тех, кому была отправлена коммуникация совершили покупку у партнера в теч. 60 дней после отправки коммуникации
* сколько клиентов, из тех, кому была доставлена коммуникация совершили покупку у партнера в теч. 60 дней после отправки коммуникации

Необходимо посчитать показатели выше 1-3 запросами. В результате выполнения запросов должна сформироваться таблица следующего вида:

|  |  |  |  |
| --- | --- | --- | --- |
| CELL\_PK | Кол-во клиентов, кому отправлено | Кол-во клиентов, совершивших покупку из отправленных | …… |

## Задача 2.

* Из таблицы покупки у партнера необходимо найти всех клиентов, которые совершали покупки у партнера Partner\_id = 1111 с января 2019 по апрель 2019,

при этом не совершили ни одной покупки у этого же партнера в мае и июне 2019

## Задача 3.

Есть таблица оборота по карте клиента в день, где

|  |  |
| --- | --- |
| Ow\_id | уникальный идентификатор клиента |
| Day | Расчетный день |
| POS | Оборот по карте |

|  |  |  |
| --- | --- | --- |
| Например: |  |  |
| Ow\_id | Day | POS |
| 1 | 12.02.2016 | 2500 |
| 1 | 13.02.2016 | 1000 |
| 1 | 14.02.2016 | 700 |
| 2 | 12.02.2016 | 3000 |
| 2 | 13.02.2016 | 4500 |
| 2 | 14.02.2016 | 300 |

За каждую 1000 рублей клиенту начисляется 1 бонус. При этом, если клиент, например купил на 2500 руб. ему начислят 2 бонуса, а 500 рублей просуммируется к след покупке. Нужно написать скрипт, который считает начисленные бонусы на каждый день (предпочтителен скрипт без использования цикла)

Решение.

Задача 1.

1 запрос на основе объединения таблиц из трех разных запросов. 1 подзапрос подсчитывает количество клиентов с разными результатами отправки коммуникаций, 2ой подзапрос считает количество клиентов совершивших покупку в течение 60 дней, кому была отправлена коммуникация, 3ий подзапрос считает количество клиентов совершивших покупку в течение 60 дней, кому была доставлена коммуникация.

Запрос:

SELECT CELL\_PK, count\_sent, count\_deliver, count\_open, count\_clik,

sent\_purch\_users, deliver\_purch\_users

FROM

(SELECT CELL\_PK,

COUNT(DISTINCT owd\_id) FILTER(WHERE sent=1) as count\_sent,

COUNT(DISTINCT owd\_id) FILTER(WHERE delivered=1) as count\_deliver,

COUNT(DISTINCT owd\_id) FILTER(WHERE opened=1) as count\_open,

COUNT(DISTINCT owd\_id) FILTER(WHERE cliked=1) as count\_clik

FROM interactions

GROUP BY CELL\_PK

ORDER BY CELL\_PK) t1

JOIN

(WITH cte AS

(SELECT owd\_id, sent\_date, parthner\_id, CELL\_PK

FROM interactions

WHERE sent = 1

GROUP BY CELL\_PK, parthner\_id, sent\_date, owd\_id)

SELECT CELL\_PK, COUNT(DISTINCT owd\_id) as sent\_purch\_users

FROM

(SELECT cte.CELL\_PK, cte.sent\_date, cte.parthner\_id, cte.owd\_id,

purchases.CASH\_DT

FROM cte INNER JOIN purchases USING (parthner\_id, owd\_id)

WHERE CASH\_DT BETWEEN sent\_date AND (sent\_date+59)

ORDER BY CELL\_PK, sent\_date, owd\_id) t

GROUP BY CELL\_PK

ORDER BY CELL\_PK) t2

USING (CELL\_PK)

JOIN

(WITH cte AS

(SELECT owd\_id, sent\_date, parthner\_id, CELL\_PK

FROM interactions

WHERE delivered = 1

GROUP BY CELL\_PK, parthner\_id, sent\_date, owd\_id)

SELECT CELL\_PK, COUNT(DISTINCT owd\_id) as deliver\_purch\_users

FROM

(SELECT cte.CELL\_PK, cte.sent\_date, cte.parthner\_id, cte.owd\_id,

purchases.CASH\_DT

FROM cte INNER JOIN purchases USING (parthner\_id, owd\_id)

WHERE CASH\_DT BETWEEN sent\_date AND (sent\_date+59)

ORDER BY CELL\_PK, sent\_date, owd\_id) t

GROUP BY CELL\_PK

ORDER BY CELL\_PK) t3

USING (CELL\_PK);

, где

interactions - таблица с историей коммуникаций,

purchases - таблица с покупками клиентов,

count\_sent - количество клиентов, кому была отправлена коммуникация,

count\_deliver - количество клиентов, кому была доставлена коммуникация,

count\_open - количество клиентов, кто открыл коммуникацию,

count\_clik - количество клиентов, кто нажал на ссылку,

sent\_purch\_users - количество клиентов, кому была отправлена коммуникация, кто совершил покупку у партнера в течение 60 дней после коммуникации,

deliver\_purch\_users - количество клиентов, кому была доставлена коммуникация, кто совершил покупку у партнера в течение 60 дней после коммуникации.

Пример вывода по данным:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **cell\_pk** | **count\_sent** | **count\_deliver** | **count\_open** | **count\_clik** | **sent\_purch\_users** | **deliver\_purch\_users** |
| 1720 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1788 | 2 | 2 | 1 | 1 | 1 | 1 |

Задача 2.

Запрос:

WITH cte AS

(SELECT owd\_id, parthner\_id, CASH\_DT, pos

FROM purchases

WHERE parthner\_id = 1111)

SELECT owd\_id

FROM cte

WHERE (CASH\_DT BETWEEN '2019-01-01' AND '2019-04-30') AND

(owd\_id NOT IN (

SELECT owd\_id FROM cte

WHERE CASH\_DT BETWEEN '2019-05-01' AND '2019-06-30'));

В тестовых данных по указанным условиям задачи не находится ни одно записи. Я добавила по одной записи в таблицу коммуникаций и покупок с новым id клиента.

(125, '2019-06-18', 'viber', 1111, 1, 1, 0, 0, 1720)

и

(125, '2019-01-29', 700, 1111)

Тогда запрос выдаст этого клиента в результате.

|  |
| --- |
| **owd\_id** |
| 125 |

Задача 3.

Логика решения: К таблице с покупками с использованием оконной функции добавляется вычисляемая колонка - нарастающая сумма покупок для каждого отдельного клиента. От этой суммы берется остаток после выделения целой части в тысячах. Таким образом получается скользящий остаток в рублях, с учетом возможного предыдущего остатка от предшествующей покупки. Далее из этой таблицы считается количество бонусов начисленных в каждый день, так же с использованием OVER() - от суммы покупки в день + остаток от начисления бонусов в предшествующий день.

Запрос:

SELECT owd\_id, dday, POS,

TRUNC((POS + LAG(roll\_rest,1,0::double precision) OVER(PARTITION BY owd\_id))/1000) AS bonus

FROM (SELECT \*,

(SUM(POS) OVER(PARTITION BY owd\_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) -

TRUNC(SUM(POS) OVER(PARTITION BY owd\_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)/

1000)\*1000) AS roll\_rest FROM bonuses) b

ORDER BY owd\_id, dday;

, где bonuses - таблица с оборотом по картам клиентов,

dday - дата покупки,

roll\_rest - скользящий остаток в рублях от целых тысяч.

Пример вывода результата:

|  |  |  |  |
| --- | --- | --- | --- |
| **owd\_id** | **dday** | **pos** | **bonus** |
| 1 | 2016-02-12T00:00:00.000Z | 2500 | 2 |
| 1 | 2016-02-13T00:00:00.000Z | 1000 | 1 |
| 1 | 2016-02-14T00:00:00.000Z | 700 | 1 |
| 2 | 2016-02-12T00:00:00.000Z | 3000 | 3 |
| 2 | 2016-02-13T00:00:00.000Z | 4500 | 4 |
| 2 | 2016-02-13T00:00:00.000Z | 300 | 0 |