

ДЗ

Первая часть:

В качестве ДЗ делаем прогноз ТО на 05.2017. В качестве метода прогноза - считаем сколько денег тратят группы клиентов в день:

1. Группа часто покупающих (3 и более покупок) и которые последний раз покупали не так давно. Считаем сколько денег оформленного заказа приходится на 1 день. Умножаем на 30.
2. Группа часто покупающих, но которые не покупали уже значительное время. Так же можем сделать вывод, из такой группы за след месяц сколько купят и на какую сумму.
3. Отдельно разобрать пользователей с 1 и 2 покупками за все время, прогнозируем их.
4. В итоге у вас будет прогноз ТО и вы сможете его сравнить с фактом и оценить грубо разлет по данным.

Как источник данных используем данные по продажам за 2 года.

Вторая часть:

В GA Demo Account найти за последний год день недели с самым большим трафиком (кол-во сеансов) и день недели с самым большим товарооборотом. (к примеру, может получиться, что в среду самый большой трафик по данным года, а товарооборот - в четверг)

1.1. Можно сделать временную таблицу для заказов до 05.17 для упрощения запросов и потом строить из нее таблицы для групп.

```
USE orders;
-- таблица заказов до 05.17
CREATE TABLE orders_before_0517
SELECT *
FROM orders.orders_
WHERE date(o_date) < date('2017-05-01')
```

Находим пользователей, которые покупали три и более раз, с последней покупки которых прошло не более 30 дней. Выводим их средний чек, общий чек, количество дней между первой и последней покупкой, количество покупок и количество денег оформленного заказа на 1 день.

```
-- табличка для первой группы
SELECT user_id, CEIL(AVG(price)) a,
SUM(price) s,
TIMESTAMPDIFF(DAY, MIN(o_date), MAX(o_date)) days,
30/TIMESTAMPDIFF(DAY, MIN(o_date),MAX(o_date))/(COUNT(id_o)-1) a_d,
COUNT(id_o) c,
MAX(o_date) max_date,
30 * CEIL(30/TIMESTAMPDIFF(DAY, MIN(o_date),MAX(o_date))/(COUNT(id_o)-1)* AVG(price)) sum_per_month
FROM orders.orders_before_0517
GROUP BY user_id
HAVING c >= 3 AND max_date > date('2017-04-01') AND days > 3
ORDER BY sum_per_month DESC
```

user_id	a	s	days	a_d	c	max_date	sum_per_month
INT(11)	DOUBLE(17, 0)	DOUBLE	BIGINT(21)	DECIMAL(10, 8)	BIGINT(21)	DATE	DOUBLE(17, 0)
3556940	37425	187124	4	1,87500000	5	16.04.2017	2105160
3570245	33806	101417,4	13	1,15384615	3	27.04.2017	1170210
3606130	12465	37394	5	3,00000000	3	25.04.2017	1121820
2857524	15750	63000	5	2,00000000	4	26.04.2017	945000
3475576	9660	28977,9	5	3,00000000	3	06.04.2017	869340
3536837	11441	34322,4	6	2.50000000	3	17.04.2017	858090

Тут возможно имеет смысл выкинуть пользователей с разницей в днях между покупками менее 5. Потому что более 3 дней получаются достаточно большие значения. Занесем в таблицу и те и другие значения.

```
-- сумма пользователей 1 группы за месяц
SELECT SUM(t.sum_per_month)
FROM
(
  SELECT user_id, CEIL(AVG(price)) a,
         SUM(price) s,
         TIMESTAMPDIFF(DAY, MIN(o_date), MAX(o_date)) days,
         30/TIMESTAMPDIFF(DAY, MIN(o_date),MAX(o_date))/(COUNT(id_o)-1) a_d,
         COUNT(id_o) c,
         MAX(o_date) max_date,
         30 * CEIL(30/TIMESTAMPDIFF(DAY, MIN(o_date),MAX(o_date))/(COUNT(id_o)-1)* AVG(price)) sum_per_month
  FROM orders.orders_before_0517
  GROUP BY user_id
  HAVING c >= 3 AND max_date > date('2017-04-01') AND days > 5
) t
```

1.2. Проделаем тоже самое для 2 группы. Тут возьмем всех оставшихся пользователей, у которых 3 и более покупок.

```
-- сумма пользователей 2 группы за месяц
SELECT SUM(t.sum_per_month)
FROM
(
  SELECT user_id, CEIL(AVG(price)) a,
         SUM(price) s,
         TIMESTAMPDIFF(DAY, MIN(o_date), MAX(o_date)) days,
         30/TIMESTAMPDIFF(DAY, MIN(o_date),MAX(o_date))/(COUNT(id_o)-1) a_d,
         COUNT(id_o) c,
         MAX(o_date) max_date,
         30 * CEIL(30/TIMESTAMPDIFF(DAY, MIN(o_date),MAX(o_date))/(COUNT(id_o)-1)* AVG(price)) sum_per_month
  FROM orders.orders_before_0517
  GROUP BY user_id
  HAVING c >= 3 AND max_date <= date('2017-04-01') AND days > 3
) t
```

1 группа days > 3	2 группа days > 3
147 541 950	717 910 410
1 группа days > 5	2 группа days > 5
128 057 970	586 633 620

Как видно из таблички показатели для групп достаточно высокие. Чтобы приблизить к реальности прогноз необходимо брать пользователей с большим количеством покупок или увеличивать промежутки между покупками при составлении прогноза. Для второй группы такие завышенные показатели, потому что стоит брать меньший период времени, то есть по сути учитывать группу Lost пользователей.

```
-- сумма пользователей 2 группы за месяц
SELECT SUM(t.sum_per_month)
FROM
(
  SELECT user_id, CEIL(AVG(price)) a,
         SUM(price) s,
         TIMESTAMPDIFF(DAY, MIN(o_date), MAX(o_date)) days,
         30/TIMESTAMPDIFF(DAY, MIN(o_date),MAX(o_date))/(COUNT(id_o)-1) a_d,
         COUNT(id_o) c,
         MAX(o_date) max_date,
         30 * CEIL(30/TIMESTAMPDIFF(DAY, MIN(o_date),MAX(o_date))/(COUNT(id_o)-1)* AVG(price)) sum_per_month
  FROM orders.orders_before_0517
  GROUP BY user_id
  HAVING c >= 3 AND max_date <= date('2017-04-01') AND max_date > date('2017-02-01') AND days > 5
) t
```

Запрос может выглядеть следующим образом если считать тех кто не покупал более 90 дней назад в группе Lost.

2 группа days >	
3 с учетом lost	
	177 686 850
2 группа days >	
5 с учетом lost	
	150 956 250

Результаты значительно ближе к реальности.

1.3. Чтобы оценить процент пользователей покупающих второй раз и третий раз можно использовать подобные запросы.

```
-- количество пользователей купивших в 3 месяце первый раз и купивших в 4 месяце второй раз
SELECT COUNT(user_id)
FROM
(
  SELECT user_id, COUNT(id_o) c,
         MAX(o_date) max_date,
         MIN(o_date) min_date
  FROM orders.orders_before_0517
  GROUP BY user_id
  HAVING c = 2 AND max_date >= date('2017-04-01') AND min_date >= date('2017-03-01') AND min_date < date('2017-04-01')
) t

-- количество пользователей купивших в 3 месяце первый раз и не купивших более
SELECT COUNT(user_id)
FROM
(
  SELECT user_id, o_date, COUNT(id_o) c
  FROM orders.orders_before_0517
  GROUP BY user_id
  HAVING c = 1 AND date(o_date) > date('2017-02-01') AND date(o_date) < date('2017-03-01')
) t

-- количество пользователей купивших в 3 месяце первый раз и купивших до 5 месяца еще 2 раза
SELECT COUNT(user_id)
FROM
(
  SELECT user_id, COUNT(id_o) c,
         MAX(o_date) max_date,
         MIN(o_date) min_date
  FROM orders.orders_before_0517
  GROUP BY user_id
  HAVING c = 3 AND max_date >= date('2017-04-01') AND min_date >= date('2017-03-01') AND min_date < date('2017-04-01')
) t

-- количество пользователей купивших со 2 месяца по 3 два раза и не купивших более
SELECT COUNT(user_id)
FROM
(
  SELECT user_id, COUNT(id_o) c,
         MAX(o_date) max_date,
         MIN(o_date) min_date
  FROM orders.orders_before_0517
  GROUP BY user_id
  HAVING c = 2 AND max_date > date('2017-02-01') AND min_date > date('2017-02-01')
         AND max_date < date('2017-03-01') AND min_date < date('2017-03-01')
) t
```

количество пользователей купивших в 3 месяце первый раз и купивших в 4 месяце второй раз	количество пользователей купивших в 3 месяце первый раз и не купивших более	%	
1301	27184	4,79%	
количество пользователей купивших в 3 месяце первый раз и купивших до 5 месяца еще 2 раза	количество пользователей купивших со 2 месяца по 3 два раза и не купивших более	%	
666	1733	38,43%	

Прогнозируем их с помощью подобных запросов.

-- количество и сумма чека пользователей купивших в 4 месяце 1 раз

```
SELECT COUNT(user_id) c_u, SUM(s) s_s
FROM
(
  SELECT o_date,user_id,
         COUNT(id_o) c, SUM(price) s
    FROM orders.orders_before_0517
   GROUP BY user_id
   HAVING c = 1 AND date(o_date) >= date('2017-04-01')
) t
```

-- количество и сумма чека пользователей купивших в 4 месяце 2 раза

```
SELECT COUNT(user_id) c_u, SUM(s) s_s
FROM
(
  SELECT MIN(o_date) min_date,
         MAX(o_date) max_date,
         user_id,
         COUNT(id_o) c, SUM(price) s
    FROM orders.orders_before_0517
   GROUP BY user_id
   HAVING c = 2 AND min_date >= date('2017-04-01') AND max_date >= date('2017-04-01')
) t
```

сумма чека пользователей купивших в 4 месяце 1 раз	Прогноз на 5 месяц	Общий прогноз на 5 месяц
84 261 234	4 032 661	288 695 928
сумма чека пользователей купивших в 4 месяце 2 раза	Прогноз на 5 месяц	
14 699 397	5 649 047	

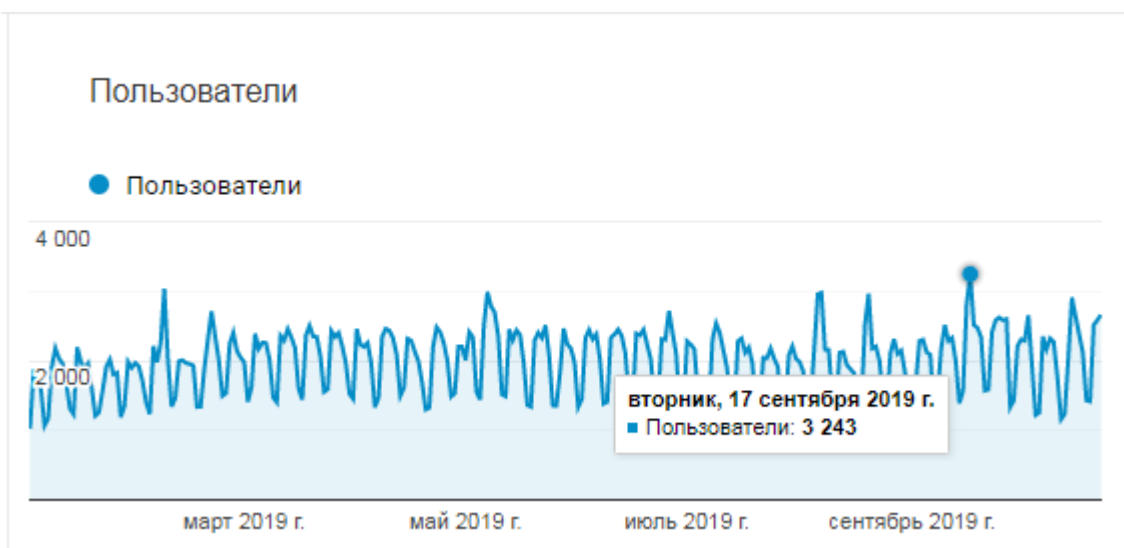
1.4. Сравним результаты.

```
-- Сумма за 5 месяц 17 года  
SELECT SUM(price)  
FROM orders.orders_  
WHERE date(o_date) >= date('2017-05-01') AND date(o_date) < date('2017-06-01')
```

Общий прогноз на 5 месяц	Реальные значения	%
288 695 928	217 075 552	75,19%

Для улучшения прогноза имеет смысл для 1 и 2 групп брать клиентов у которых более 3 покупок, а тех у кого 3 покупки прогнозировать подобно тем у кого 1 и 2 покупки. Также можно посчитать различные вероятности для группы Lost и разнице в днях между покупками.

2.1 Самый большой трафик



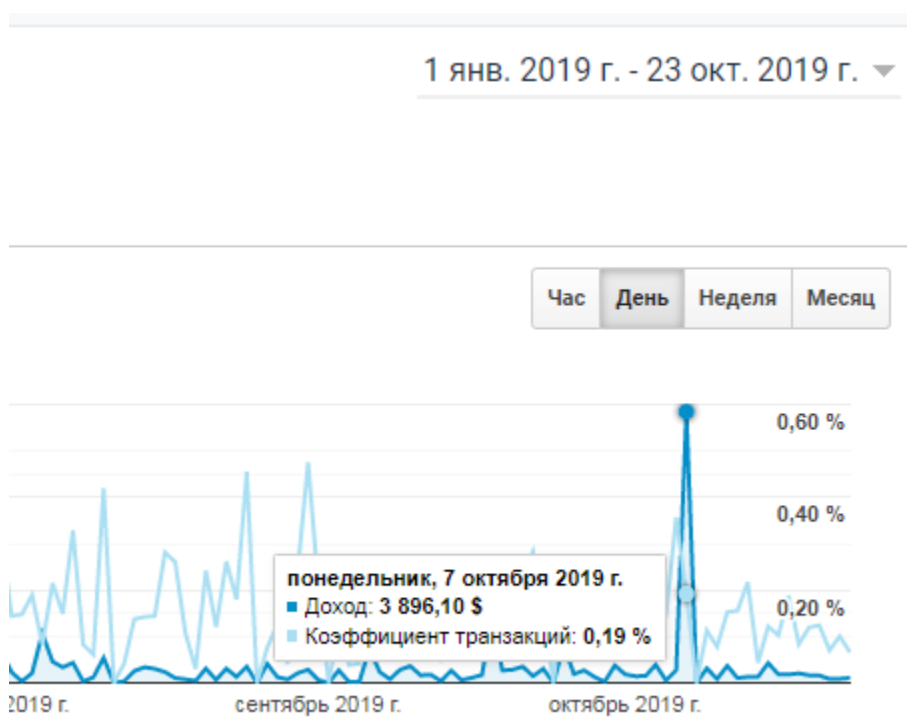
Самое большое количество транзакций



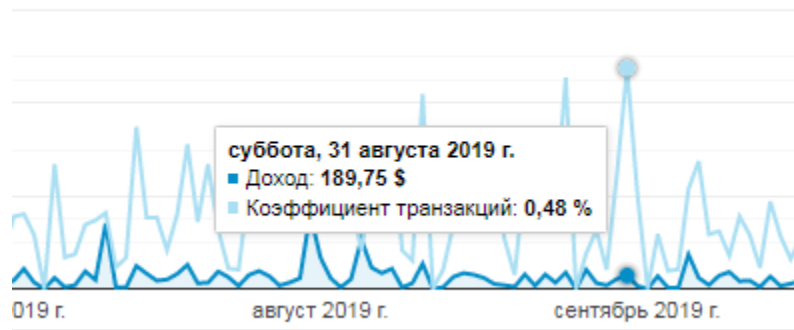
Самое большое количество совершенных покупок



Самый большой доход



На графиках видно, что самый большой доход не соответствует дням с самым большим количеством транзакций, пользователей, покупок.



Тут сравнение дня с самым большим количеством транзакций и дохода в этот день.