

SQL

Существует три таблицы:

1. Orders - информация о заказах

date - время заказа в timestamp

clientId - id пользователя, который совершил заказ

id - id заказа

ClientOrderStateID - статус заказа, меняется во времени в зависимости от стадии заказа

Статусы заказа

1 - заказ оформлен

2 - заказ получен

3 - заказ отменен

2. AdditionalInfo - информация об источнике заказа

ClientOrderID - id заказа

code - параметр

value - значение параметра из столбца code

code принимает следующие значения: IsTestOrder - тестовый заказ или нет

platform - платформа, с которой был куплен заказ

3. ClientOrderItem - информация о товарах в заказе

ClientOrderID - id заказа

ItemId - id товара

category - категория товара

qty - количество

price - цена за одну единицу товара

Задания

1. Вывести сколько заказов было оформлено, и сколько в итоге получено. Не учитывать тестовые заказы.
2. Для каждой платформы и категории посчитать: сколько было куплено товаров, сколько было получено заказов, GMV. Не учитывать тестовые заказы.
3. Найти категорию, которая приносит наибольшую выручку
4. Какой товар чаще других встречается в отмененных заказах.
5. Найдите среднее время между первым и вторым заказом у пользователей. Для решения запроса не используйте джойны. Тестовые заказы фильтровать не нужно.
6. Для каждой категории найдите топ – 3 пользователей, у которых наименьшее количество дней между первой и последней покупкой в этой категории.

При выполнении использовала MySQL

1. Вывести сколько заказов было оформлено, и сколько в итоге получено. Не учитывать тестовые заказы.

Решение:

```
SELECT O.ClientOrderStateID, COUNT(O.ClientOrderStateID) as 'Кол-во заказов'  
FROM Orders as O, AdditionalInfo as A  
WHERE O.id = A.ClientOrderID AND (O.ClientOrderStateID=1 or O.ClientOrderStateID=2)  
AND A.code!='IsTestOrder'  
GROUP BY O.ClientOrderStateID ;
```

2. Для каждой платформы и категории посчитать: сколько было куплено товаров, сколько было получено заказов, GMV. Не учитывать тестовые заказы.

Решение:

```
SELECT COUNT(DISTINCT C.ItemId) as 'Кол-во товаров', COUNT(DISTINCT  
C.ClientOrderID) as 'Кол-во заказов'  
FROM ClientOrderItem as C, AdditionalInfo as A  
WHERE C.ClientOrderID = A.ClientOrderID AND A.code!='IsTestOrder'  
GROUP BY A.platform;
```

3. Найти категорию, которая приносит наибольшую выручку

Решение:

```
SELECT C.category, SUM(C.price)  
FROM ClientOrderItem as C  
GROUP BY C.category  
having SUM(C.price) = (SELECT MAX(m.sumpr)  
  FROM (SELECT C.category, SUM(C.price) as sumpr  
    FROM ClientOrderItem as C  
    GROUP BY C.category) AS m);
```

4. Какой товар чаще других встречается в отмененных заказах.

Решение:

```
SELECT C.category, COUNT(C.category)  
FROM ClientOrderItem as C, Orders as O  
WHERE O.id = C.ClientOrderID AND O.ClientOrderStateID=2  
GROUP BY C.category  
HAVING COUNT(C.category)=(SELECT MAX(m.sumpr)  
  FROM (SELECT C.category, COUNT(C.category) as sumpr  
    FROM ClientOrderItem as C  
    GROUP BY C.category) AS m);
```

Со следующими заданиями возникли трудности в связи с отсутствием времени и опыта в оконных функциях.

5. Найдите среднее время между первым и вторым заказом у пользователей. Для решения запроса не используйте джойны. Тестовые заказы фильтровать не нужно.

Решение:

```
With qu as (SELECT *, rank() over (partition by clientId order by date) as rnk FROM Orders)  
SELECT avg(datediff((SELECT date FROM qu WHERE rnk=2),(SELECT date FROM qu  
WHERE rnk=1)))  
FROM qu;
```

Это не является полным решением, т.к. выводит среднее значение только по одному клиенту.