

В следующих задачах продолжим работать с данными мобильного приложения крупной розничной сети. В этих данных отражено то, как наше приложение устанавливают (**installs**), как активно в нем просматривают товары (**events**), как активно в нем покупают (**checks**), и данные о склейке устройств с логинами (**devices**).

Теперь описание данных:

**installs** — содержит данные об установках приложения по дням.

- **DeviceID** — идентификатор устройства, на которое было установлено приложение;
- **InstallationDate** — дата установки приложения;
- **InstallCost** — цена установки приложения в рублях;
- **Platform** — платформа, на которой было установлено приложение (iOS/ Android);
- **Source** — источник установки приложения (магазин приложения/ рекламная система/ переход с сайта).

**events** — содержит данные о том, как активно пользователи просматривают товары в приложении по дням.

- **DeviceID** — идентификатор устройства, на котором используется приложение;
- **AppPlatform** — платформа, на которой используется приложение (iOS/ Android);
- **EventDate** — дата, за которую собрана статистика;
- **events** — количество просмотров всех товаров за этот день у этого DeviceID.

**checks** — содержит данные о покупках пользователей в приложении по дням

- **UserID** — идентификатор пользователя;
- **Rub** — суммарный чек пользователя на дату;
- **BuyDate** — дата, за которую собрана статистика.

Особенность приложения заключается в том, что для просмотра товаров не нужна авторизация. До момента авторизации про пользователя известен только его DeviceID — идентификатор устройства. При этом для совершения покупки логин обязателен. На моменте авторизации пользователю присваивается UserID, и тогда мы уже знаем два его идентификатора: DeviceID (устройство) и UserID (логин). Так как на этапах установки приложения и просмотра каталога пользователь еще может быть не авторизован, там мы сохраняем только DeviceID. Но так как покупки нельзя совершить без авторизации, то покупки сохраняются только с UserID. Для того чтобы просмотры и установки можно было объединить с

покупками, нам нужна таблица соответствия DeviceID к UserID, то есть таблица **devices**:

- **DeviceID** — идентификатор устройства;
- **UserID** — идентификатор пользователя.

Задания:

1. Продолжим работать с таблицей checks!

Посчитайте, сколько покупок приходится на каждого клиента, указав имя столбца как NumChecks. Результирующую таблицу с UserID, NumChecks отсортируйте по убыванию (DESC) нового столбца. Ограничение вывода – 10.

```
SELECT UserID, COUNT(*) AS NumChecks
FROM checks
GROUP BY UserID
ORDER BY NumChecks DESC
LIMIT 10
```

2. Давайте немного усложним задачу, и посчитаем ещё и сумму сделанных покупок в рублях (Rub).

Выведите в результирующую таблицу UserID клиентов, количество сделанных покупок (NumChecks) и сумму потраченных денег (Revenue). Результат отсортируйте по убыванию Revenue, оставив 10 записей.

```
SELECT UserID, COUNT(*) AS NumChecks, SUM(Rub) AS Revenue
FROM checks
GROUP BY UserID
ORDER BY Revenue DESC
LIMIT 10
```

3. Теперь давайте попробуем с помощью группировки и агрегатных функций посмотреть динамику трат клиентов. Для этого нужно посчитать по дням минимальный, средний, и максимальный чек. Иными словами, необходимо сгруппироваться по датам, и использовать агрегатные функции MIN, AVG, MAX.

Выведите минимальный (MinCheck), максимальный (MaxCheck) и средний (AvgCheck) чек по дням. В результирующей таблице используйте сортировку по убыванию (DESC) столбца BuyDate. Как и на предыдущем шаге, ограничение на вывод равно 10.

```
SELECT BuyDate, MIN(Rub) AS MinCheck, MAX(Rub) AS MaxCheck, AVG(Rub)
AS AvgCheck
```

```
FROM checks
GROUP BY BuyDate
ORDER BY BuyDate DESC
LIMIT 10
```

4. Представим, что для проведения новой акции нам потребовалось выгрузить сегмент пользователей, которые приносят наибольшую выручку.

Найдите покупателей, сумма покупок которых превышает 10 000 рублей (Rub). Для этого сгруппируйте пользователей по UserID, затем примените нужную агрегатную функцию и назовите новую колонку Revenue. Отфильтруйте нужные значения, результат отсортируйте по убыванию UserID и выведите 10 строк.

```
SELECT UserID, SUM(Rub) AS Revenue
FROM checks
GROUP BY UserID
HAVING SUM(Rub) > 10000
ORDER BY UserID DESC
LIMIT 10
```

В этой и последующих задачах мы будем работать с таблицей retail в ClickHouse, которая содержит данные о покупках.

Таблица включает следующие поля:

- InvoiceNo — идентификатор покупки
- StockCode — идентификатор товара на складе
- Description — текстовое описание товара
- Quantity — количество товара
- InvoiceDate — дата покупки
- UnitPrice — цена товара
- CustomerID — идентификатор покупателя
- Country — страна

5. Представим, что к вам пришел менеджер с проблемой: за последний месяц он наблюдает снижение выручки в некоторых регионах и хочет определить возможную причину этого снижения. Для начала можно сгруппировать данные о выручке по странам и найти среди них топ-5 стран по величине выручки, так как интереснее всего нам будет смотреть именно на данные этих стран.

Посчитайте выручку (Revenue) по странам и отсортируйте таблицу в порядке убывания выручки. В качестве ответа скопируйте название страны с самой высокой выручкой. Обратите внимание, что в изначальной таблице указана цена одной единицы товара (UnitPrice) и количество купленного товара

(Quantity), а не выручка, и для получения Revenue нужно будет использовать агрегирующую функцию Sum и оператор умножения \*.

```
SELECT Country, SUM(UnitPrice*Quantity) AS Revenue
FROM retail
GROUP BY Country
ORDER BY Revenue DESC
```

Ответ: United Kingdom

6. В качестве более подробного описания выручки по странам можно посмотреть, каково среднее количество купленных товаров и средняя цена товара в покупках, совершенных в определенной стране.

Посчитайте среднее число купленных товаров по стране и среднюю цену товара, сгруппировав данные по странам и используя агрегирующую функцию Avg, и отсортируйте по убыванию средней цены товара (DESC).

Обратите внимание, что в данных встречаются строки с Description 'Manual', которые включают данные об удаленных из чека позициях. Для получения правильного ответа такие строки необходимо отфильтровать с использованием операторов WHERE и неравенства !=.

В качестве ответа впишите название страны, в которой самая высокая средняя цена за купленный товар.

```
SELECT Country, AVG(Quantity), AVG(UnitPrice)
FROM retail
WHERE Description != 'Manual'
GROUP BY Country
ORDER BY AVG(UnitPrice) DESC
```

Ответ: Canada

7. Теперь посмотрим на динамику общей суммы выручки по месяцам.

Посчитайте выручку (Revenue) по месяцам (обратите внимание, что в таблице указана цена одной единицы товара и количество купленного товара), округлив InvoiceDate к началу месяца с помощью toStartOfMonth().

В качестве ответа укажите сумму выручки (Revenue) из первой строки результирующей таблицы, используя сортировку по убыванию (DESC) по столбцу Revenue. Не забывайте отфильтровать строки по условию Description != 'Manual'

```
SELECT toStartOfMonth(InvoiceDate) AS month, SUM(Quantity*UnitPrice)
AS Revenue
FROM retail
WHERE Description != 'Manual'
GROUP BY toStartOfMonth(InvoiceDate)
ORDER BY Revenue DESC
```

Ответ: 1130664.25

8. Давайте посмотрим на динамику выручки от покупателей, которые в среднем покупают самые дорогие товары. В качестве целевой метрики будем использовать среднюю цену купленного товара (UnitPrice), данные посмотрим за март 2011 года.

Посчитайте среднюю цену товара клиента (UnitPrice) с группировкой по id покупателя, используя агрегирующую функцию Avg, затем выведите строки с покупками тех покупателей, у которых самая высокая средняя цена купленного товара в марте. Не забывайте отфильтровать строки по условию Description != 'Manual'. В качестве ответа впишите CustomerID покупателя с самой высокой средней ценой купленного товара.

```
SELECT CustomerID, AVG(UnitPrice) AS Revenue
FROM retail
WHERE toStartOfMonth(InvoiceDate) = '2011-03-01' AND Description != 'Manual'
GROUP BY CustomerID
ORDER BY Revenue DESC
```

Ответ: 13452

9. Давайте попробуем ответить на следующий вопрос: "Как изменилось среднее, минимальное и максимальное количество купленного товара в стране с наибольшей выручкой в течение последних месяцев?"

Для этого нужно сгруппировать данные по странам и месяцам, посчитать среднее (Avg), минимальное (Min) и максимальное (Max) количество купленных товаров (Quantity), и отсортировать данные по месяцу, применить фильтр по стране (нам нужны данные из страны с наибольшей выручкой из задания 8 (step 8)). Если обратить внимание на полученное значение минимального количества товаров, то станет понятно, что там есть отрицательные значения, которые следует отфильтровать (также не забывайте отфильтровывать значения с Description != 'Manual').

В качестве ответа скопируйте наибольшее историческое значение среднего числа купленных товаров для этой страны.

```
SELECT toStartOfMonth(InvoiceDate) AS month, AVG(Quantity),
MIN(Quantity), MAX(Quantity)
FROM retail
WHERE Country = 'United Kingdom' AND Description != 'Manual' AND
Quantity > 0
GROUP BY toStartOfMonth(InvoiceDate)
ORDER BY AVG(Quantity) DESC
```

Ответ: 16.45

