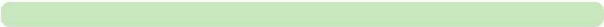


КОГОРТЫ В SQL

18 задач по удержанию

```
SELECT cohort_month,  
       SUM(retained) * 1.0 / COUNT(*)  
FROM cohort_flags  
GROUP BY cohort_month;
```



retention · activation · день 7

month age · revenue matrix

Глеб Зайцев
Когорты в SQL. 18
задач по удержанию

<https://litres.ru/74419932>

SelfPub; 2026

Аннотация

Практическая книга по когортному анализу в SQL. Внутри 18 задач на маленьких таблицах: месяц регистрации, размер когорты, активация, retention, возраст месяца, revenue matrix, повторная покупка, день 7, каналы и компактная таблица когорт. Все запросы запускаются в SQLite.

Содержание

Перед первой задачей	4
Зачем нужны когорты	4
Главный контроль	5
Маршрут по задачам	6
Практические задачи	7
Задача 1. Разнести клиентов по месяцу регистрации	7
Задача 2. Посчитать размер когорты	10
Задача 3. Найти первую покупку в когорте	12
Задача 4. Посчитать активацию когорты	15
Конец ознакомительного фрагмента.	16

Глеб Зайцев

Когорты в SQL. 18

задач по удержанию

Перед первой задачей

Зачем нужны когорты

Общий средний показатель часто смешивает старых и новых клиентов так, что становится непонятно, что реально улучшается.

Когорта фиксирует момент входа клиента и позволяет смотреть активацию, удержание и деньги во времени.

Эта книга тренирует SQL-запросы для когорт без внешних BI-инструментов и без личных данных.

Главный контроль

Всегда показывайте размер когорты рядом с процентом, иначе маленькая группа может выглядеть как большой успех.

Различайте календарный месяц, точное число дней и возраст когорты: это разные правила.

Все примеры запускаются в SQLite на маленьких встроенных таблицах.

Маршрут по задачам

1. Разнести клиентов по месяцу регистрации
2. Посчитать размер когорты
3. Найти первую покупку в когорте
4. Посчитать активацию когорты
5. Посчитать удержание на следующий месяц
6. Посчитать retention rate
7. Развернуть когорту по возрасту месяца
8. Построить матрицу revenue по двум месяцам
9. Сравнить когорты по прибыли
10. Найти когорты без повторной покупки
11. Проверить когорту по каналу
12. Посчитать активность на седьмой день
13. Найти клиентов, вернувшихся после паузы
14. Сравнить когорты по среднему чеку
15. Посчитать заказы на клиента в когорте
16. Найти когорты с маленьким размером
17. Собрать компактную таблицу когорт
18. Посчитать выручку первых 30 дней

Практические задачи

Задача 1. Разнести клиентов по месяцу регистрации

Рабочий вопрос

Нужно получить когорту каждого клиента по месяцу регистрации. Это нужно для отчёта, который можно проверить без ручной правки.

Данные

```
CREATE TABLE customers (client_id
INTEGER, registered_at TEXT);
INSERT INTO customers VALUES
(10, '2026-08-15'), (20, '2026-08-20'),
(30, '2026-09-01');
```

Запрос

```
SELECT                                client_id,  
substr(registered_at,1,7) AS cohort_month  
FROM customers  
ORDER BY client_id;
```

Ожидаемый результат

```
[[10, "2026-08"], [20, "2026-08"], [30,  
"2026-09"]]
```

Почему работает

substr берёт YYYY-MM из даты формата YYYY-MM-DD и создаёт простую месячную когорту.

Запрос сделан на маленькой синтетической таблице, поэтому ответ можно проверить глазами и повторить в SQLite.

Явный ORDER BY или отдельная агрегация фиксируют смысл результата, а не оставляют его на случайный порядок строк.

Где легко ошибиться

Если даты хранятся не в ISO-формате, такой приём может дать неверную группировку.

Самостоятельная проверка

Какая когорта у клиента 30? Объясните ответ по исходным строкам и условию запроса.

Ответ: 2026-09. Это следует из исходных строк и условия запроса.

Задача 2. Посчитать размер когорты

Рабочий вопрос

Нужно узнать, сколько клиентов вошло в каждую когорту регистрации. Это нужно для отчёта, который можно проверить без ручной правки.

Данные

```
CREATE TABLE customers(client_id
INTEGER, registered_at TEXT);
INSERT INTO customers VALUES
(10, '2026-08-15'), (20, '2026-08-20'),
(30, '2026-09-01');
```

Запрос

```
SELECT substr(registered_at,1,7) AS
cohort_month, COUNT(*) AS cohort_size
FROM customers
GROUP BY cohort_month
ORDER BY cohort_month;
```

Ожидаемый результат

```
[["2026-08", 2], ["2026-09", 1]]
```

Почему работает

GROUP BY по месяцу регистрации собирает клиентов в когорты и считает размер каждой.

Запрос сделан на маленькой синтетической таблице, поэтому ответ можно проверить глазами и повторить в SQLite.

Явный ORDER BY или отдельная агрегация фиксируют смысл результата, а не оставляют его на случайный порядок строк.

Где легко ошибиться

Размер когорты нужен как знаменатель; без него retention превращается в голое число вернувшихся.

Самостоятельная проверка

Сколько клиентов в августовской когорте?

Ответ: Два клиента. Это следует из исходных строк и условия запроса.

Задача 3. Найти первую покупку в когорте

Рабочий вопрос

Нужно связать регистрацию и первую покупку, чтобы понять, кто активировался.

Данные

```
CREATE TABLE customers(client_id
INTEGER, registered_at TEXT);
CREATE TABLE orders(client_id INTEGER,
paid_at TEXT);
INSERT INTO customers VALUES
(10, '2026-08-15'), (20, '2026-08-20'),
(30, '2026-09-01');
INSERT INTO orders VALUES
(10, '2026-08-16'), (10, '2026-09-02'),
(30, '2026-09-03');
```

Запрос

```
SELECT                                c.client_id,  
substr(c.registered_at,1,7)           AS  
cohort_month,                        MIN(o.paid_at)      AS  
first_paid_at  
FROM customers c  
LEFT JOIN orders o ON o.client_id =  
c.client_id  
GROUP BY c.client_id, cohort_month  
ORDER BY c.client_id;
```

Ожидаемый результат

```
[[10, "2026-08", "2026-08-16"], [20,  
"2026-08", null], [30, "2026-09",  
"2026-09-03"]]
```

Почему работает

LEFT JOIN сохраняет клиентов без покупок, а MIN показывает первую оплату там, где она есть.

Запрос сделан на маленькой синтетической таблице, поэтому ответ можно проверить глазами и повторить в SQLite.

Явный ORDER BY или отдельная агрегация фиксируют смысл результата, а не оставляют его на случайный порядок строк.

Где легко ошибиться

INNER JOIN удалил бы неактивировавшихся клиентов и зависил бы картину активации.

Самостоятельная проверка

У какого клиента нет first_paid_at? Объясните ответ по исходным строкам и условию запроса.

Ответ: У клиента 20. Это следует из исходных строк и условия запроса.

Задача 4. Посчитать активацию когорты

Рабочий вопрос

Нужно узнать долю клиентов когорты, у которых была хотя бы одна покупка.

Данные

```
CREATE TABLE customers(client_id
INTEGER, registered_at TEXT);
CREATE TABLE orders(client_id INTEGER,
paid_at TEXT);
INSERT INTO customers VALUES
(10, '2026-08-15'), (20, '2026-08-20'),
(30, '2026-09-01');
INSERT INTO orders VALUES
(10, '2026-08-16'), (30, '2026-09-03');
```

Запрос

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.