

A/B-МЕТРИКИ В SQL

18 задач без магии

```
SELECT variant,  
       SUM(paid) payers,  
       ROUND(1.0*SUM(paid)/COUNT(*),2)  
FROM experiment  
GROUP BY variant;
```

конверсия · выручка · прибыль

дубли · группы · возвраты

Глеб Зайцев

**A/B-метрики в SQL.
18 задач без магии**

«Автор»

2026

Зайцев Г.

А/В-метрики в SQL. 18 задач без магии / Г. Зайцев — «Автор»,
2026

Практическая книга по расчёту А/В-метрик в SQL без обещаний статистических чудес. Внутри 18 задач на маленьких таблицах: конверсия, размер групп, дубли назначений, пользователи в двух вариантах, средний чек, выручка и прибыль на пользователя, возвраты и сегменты. Все примеры запускаются в SQLite.

© Зайцев Г., 2026

© Автор, 2026

Содержание

Перед первой задачей	5
Что SQL может дать А/В-тесту	5
Граница ответственности	6
Маршрут по задачам	7
Практические задачи	8
Задача 1. Посчитать конверсию двух вариантов	8
Задача 2. Проверить равномерность распределения	10
Задача 3. Найти пользователей в двух вариантах	11
Конец ознакомительного фрагмента.	12

Глеб Зайцев

А/В-метрики в SQL. 18 задач без магии

Перед первой задачей

Что SQL может дать А/В-тесту

SQL не заменяет статистическое решение, но помогает аккуратно собрать базовые метрики эксперимента.

Перед выводами нужно проверить назначение пользователей, дубли, размер групп и то, что знаменатель выбран правильно.

Эта книга фокусируется на практической подготовке данных: конверсия, выручка, прибыль, возвраты и сегменты.

Граница ответственности

Задачи не обещают доказать победителя эксперимента автоматически.

Они учат собрать чистые таблицы и увидеть типовые ошибки, из-за которых красивый процент может быть неверным.

Все данные маленькие и синтетические, без внешних файлов и без привязки к реальным пользователям.

Маршрут по задачам

- [1. Посчитать конверсию двух вариантов](#)
- [2. Проверить равномерность распределения](#)
- [3. Найти пользователей в двух вариантах](#)
- [4. Сравнить средний чек по вариантам](#)
- [5. Посчитать выручку на пользователя](#)
- [6. Разделить конверсию по устройствам](#)
- [7. Проверить дубликаты назначений](#)
- [8. Собрать чистое назначение пользователя](#)
- [9. Посчитать клики по вариантам](#)
- [10. Сравнить оплату после клика](#)
- [11. Проверить минимальный размер выборки](#)
- [12. Посчитать разницу конверсий](#)
- [13. Сравнить возвраты по вариантам](#)
- [14. Посчитать валовую прибыль по вариантам](#)
- [15. Сравнить среднюю прибыль на пользователя](#)
- [16. Проверить платящих пользователей с несколькими заказами](#)
- [17. Отделить тестовых пользователей](#)
- [18. Собрать компактный итог эксперимента](#)

Практические задачи

Задача 1. Посчитать конверсию двух вариантов

Рабочий вопрос

Нужно сравнить варианты А и В по доле оплат, не смешивая число пользователей и число событий.

Данные

```
CREATE TABLE experiment (user_id INTEGER, variant TEXT, paid
INTEGER);
INSERT INTO experiment VALUES (1, 'A', 1), (2, 'A', 0), (3, 'A', 0),
(4, 'B', 1), (5, 'B', 1), (6, 'B', 0);
```

Запрос

```
SELECT variant,
COUNT(*) AS users,
SUM(paid) AS payers,
ROUND(1.0 * SUM(paid) / COUNT(*), 2) AS conversion
FROM experiment
GROUP BY variant
ORDER BY variant;
```

Ожидаемый результат

```
[["A", 3, 1, 0.33], ["B", 3, 2, 0.67]]
```

Почему работает

SUM(paid) считает оплаты, COUNT(*) считает пользователей варианта, а дробное деление даёт конверсию.

Запрос намеренно работает на маленькой таблице, поэтому результат можно проверить глазами до переноса приёма в рабочий отчёт.

ORDER BY или явная агрегация делают вывод устойчивым: строки не зависят от случайного порядка хранения данных.

Где легко ошибиться

Такая таблица показывает метрику, но сама по себе не доказывает статистическую значимость различия.

Самостоятельная проверка

У какого варианта выше конверсия? Объясните ответ, опираясь на строки исходных данных.

Ответ: У В: две оплаты из трёх против одной из трёх у А.

Задача 2. Проверить равномерность распределения

Рабочий вопрос

Нужно увидеть, сколько пользователей попало в каждый вариант, потому что сильный перекос ломает доверие к сравнению.

Данные

```
CREATE TABLE assignments(user_id INTEGER, variant TEXT);
INSERT INTO assignments VALUES (1, 'A'), (2, 'A'), (3, 'A'),
(4, 'B'), (5, 'B');
```

Запрос

```
SELECT variant, COUNT(*) AS users
FROM assignments
GROUP BY variant
ORDER BY variant;
```

Ожидаемый результат

```
[["A", 3], ["B", 2]]
```

Почему работает

GROUP BY variant показывает размер каждой группы эксперимента.

Запрос намеренно работает на маленькой таблице, поэтому результат можно проверить глазами до переноса приёма в рабочий отчёт.

ORDER BY или явная агрегация делают вывод устойчивым: строки не зависят от случайного порядка хранения данных.

Где легко ошибиться

Даже равные размеры групп не гарантируют чистый эксперимент, но сильный перекос сразу требует внимания.

Самостоятельная проверка

Какой вариант получил меньше пользователей?

Ответ: В получил двух пользователей, А получил трёх.

Задача 3. Найти пользователей в двух вариантах

Рабочий вопрос

Нужно поймать нарушение эксперимента, когда один пользователь оказался сразу в А и В.

Данные

```
CREATE TABLE assignments(user_id INTEGER, variant TEXT);
INSERT INTO assignments VALUES (1, 'A'), (2, 'A'), (2, 'B'),
(3, 'B');
```

Запрос

```
SELECT user_id, COUNT(DISTINCT variant) AS variants_seen
FROM assignments
GROUP BY user_id
HAVING COUNT(DISTINCT variant) > 1
ORDER BY user_id;
```

Ожидаемый результат

```
[[2, 2]]
```

Почему работает

COUNT(DISTINCT variant) показывает, сколько разных вариантов видел пользователь.

Запрос намеренно работает на маленькой таблице, поэтому результат можно проверить глазами до переноса приёма в рабочий отчёт.

ORDER BY или явная агрегация делают вывод устойчивым: строки не зависят от случайного порядка хранения данных.

Где легко ошибиться

Если такие пользователи есть, их нельзя молча оставлять в простом сравнении конверсии.

Самостоятельная проверка

Какой пользователь нарушил правило? Объясните ответ, опираясь на строки исходных данных.

Ответ: Пользователь 2 видел оба варианта.

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.