



Глеб Зайцев

Агрегаты SQL. 18 задач на COUNT, SUM и AVG

Глеб Зайцев

Агрегаты SQL. 18 задач на COUNT, SUM и AVG

<https://litres.ru/74419334>

SelfPub; 2026

Аннотация

Практическая книга по агрегатным функциям SQL для аналитиков и начинающих разработчиков. Внутри 18 задач на маленьких таблицах: COUNT, SUM, AVG, MIN, MAX, COUNT DISTINCT, HAVING, условная агрегация, средний чек, weighted average, NULL в расчётах и финальный KPI-отчёт. Все примеры запускаются в SQLite, данные встроены прямо в книгу.

Содержание

Перед первой задачей	4
Почему агрегаты опаснее, чем выглядят	4
Как проверять себя	5
Маршрут по задачам	6
Практические задачи	7
Задача 1. Посчитать все заказы	7
Задача 2. Посчитать выручку оплаченных заказов	9
Задача 3. Найти средний оплаченный чек	11
Конец ознакомительного фрагмента.	13

Глеб Зайцев

Агрегаты SQL. 18 задач на COUNT, SUM и AVG

Перед первой задачей

**Почему агрегаты
опаснее, чем выглядят**

COUNT, SUM и AVG выглядят простыми, пока отчёт не начинает считать не тех клиентов, не те строки или не тот уровень детализации.

Одна и та же таблица может отвечать на разные вопросы: сколько заказов, сколько клиентов, сколько товарных строк, какая выручка и какой средний чек.

Эта книга тренирует именно смысл метрики: какая строка считается фактом, где нужен фильтр, когда нужен DISTINCT и почему среднее иногда нужно взвешивать.

Как проверять себя

Перед запуском запроса посчитайте руками две-три контрольные строки. Если итог неожиданен, проблема обычно не в SQL-движке, а в выбранном зерне отчёта.

Отдельно проверяйте NULL. COUNT(*), COUNT(column), SUM и AVG относятся к отсутствующим значениям по-разному, и это часто меняет выводы.

Все запросы рассчитаны на SQLite. Базовые агрегаты SQL переносимы почти в любую СУБД, поэтому задачи полезны и за пределами SQLite.

Маршрут по задачам

1. Посчитать все заказы
2. Посчитать выручку оплаченных заказов
3. Найти средний оплаченный чек
4. Найти минимальный и максимальный чек
5. Отличить COUNT(*) от COUNT(email)
6. Посчитать уникальных покупателей
7. Сложить сумму строк корзины
8. Собрать выручку по категориям
9. Посчитать заказы и выручку клиента
10. Оставить клиентов с выручкой от 500
11. Средний чек по категории
12. Условная агрегация по статусам
13. Посчитать товары без продаж
14. Понять SUM и NULL
15. Посчитать взвешенную среднюю цену
16. Собрать выручку по дням
17. Найти категории выше среднего итога
18. Собрать финальный KPI-отчёт

Практические задачи

Задача 1. Посчитать все заказы

Рабочий вопрос

Нужно получить базовое число заказов в таблице, не считая суммы и не группируя строки.

Данные

```
CREATE TABLE orders(id INTEGER, status  
TEXT, total INTEGER);  
INSERT INTO orders VALUES (1, 'paid', 500),  
(2, 'new', 700), (3, 'paid', 300),  
(4, 'cancelled', 900);
```

Запрос

```
SELECT COUNT(*) AS orders_count  
FROM orders;
```

Ожидаемый результат

`[[4]]`

Почему работает

`COUNT(*)` считает строки результата.

В таблице `orders` четыре строки.

Статус и сумма не влияют на число строк без `WHERE`.

Где легко ошибиться

`COUNT(*)` не означает число оплаченных заказов или клиентов. Это только число строк после всех фильтров.

Самостоятельная проверка

Как изменится результат, если добавить пятую строку со `status = test`?

Ответ: `COUNT(*)` станет 5, потому что без `WHERE` считаются все строки таблицы.

Задача 2. Посчитать выручку оплаченных заказов

Рабочий вопрос

Финансисту нужна сумма только по оплаченным заказам, без новых и отменённых строк.

Данные

```
CREATE TABLE orders(id INTEGER, status  
TEXT, total INTEGER);  
INSERT INTO orders VALUES (1, 'paid', 500),  
(2, 'new', 700), (3, 'paid', 300),  
(4, 'cancelled', 900);
```

Запрос

```
SELECT SUM(total) AS paid_revenue  
FROM orders  
WHERE status = 'paid';
```

Ожидаемый результат

[[800]]

Почему работает

WHERE оставляет только строки paid.

SUM(total) складывает 500 и 300.

new и cancelled не участвуют в расчёте выручки.

Где легко ошибиться

Если забыть WHERE, сумма станет оборотом всех строк, а не подтверждённой оплаченной выручкой.

Самостоятельная проверка

Добавьте paid-заказ на 200. Каким станет paid_revenue?

Ответ: paid_revenue станет 1000, потому что к 800 добавится 200.

Задача 3. Найти средний оплаченный чек

Рабочий вопрос

Нужно посчитать среднюю сумму оплаченного заказа, не включая неоплаченные и отменённые заказы.

Данные

```
CREATE TABLE orders(id INTEGER, status  
TEXT, total INTEGER);  
INSERT INTO orders VALUES (1, 'paid', 500),  
(2, 'new', 900), (3, 'paid', 700),  
(4, 'cancelled', 300);
```

Запрос

```
SELECT      ROUND(AVG(total),      1)      AS  
avg_paid_total  
FROM orders  
WHERE status = 'paid';
```

Ожидаемый результат

[[600.0]]

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.