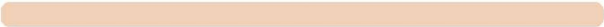


МАРЖА В SQL

18 задач для продавца и аналитика

```
SELECT sku,  
       SUM(revenue - cost) AS profit  
FROM order_items  
GROUP BY sku  
HAVING profit > 0;
```



прибыль · комиссия · скидки

возвраты · резервы · цена

Глеб Зайцев
Маржа в SQL. 18 задач
для продавца и аналитика

<https://litres.ru/74419897>

SelfPub; 2026

Аннотация

Практическая книга по расчёту маржи и прибыли в SQL. Внутри 18 задач на маленьких таблицах: прибыль заказа, прибыль SKU, комиссии, скидки, возвраты, каналы, резервы, минимальная цена и короткий маржинальный отчёт. Все запросы запускаются в SQLite.

Содержание

Перед первой задачей	4
Почему оборот не равен прибыли	4
Как читать задачи	5
Маршрут по задачам	6
Практические задачи	7
Задача 1. Посчитать прибыль по каждому заказу	7
Задача 2. Сложить прибыль по товарам	10
Задача 3. Найти убыточные товары	12
Конец ознакомительного фрагмента.	14

Глеб Зайцев

Маржа в SQL. 18 задач для продавца и аналитика

Перед первой задачей

Почему оборот не равен прибыли

Продажи могут расти, а деньги исчезать из-за себестоимости, комиссии, скидки, возврата или расходов на привлечение.

SQL помогает быстро отделить товары, которые дают оборот, от товаров, которые дают реальную маржу.

Эта книга не обещает универсальную финансовую модель, а тренирует аккуратные запросы для первичной проверки экономики.

Как читать задачи

Сначала определите, что считается строкой экономики: заказ, товар, клиент, день или канал.

Затем проверьте, какие расходы нужно вычесть: себестоимость, комиссия, скидка, резерв или привлечение.

Все примеры маленькие и исполняются в SQLite, поэтому формулу можно проверить без внешних файлов.

Маршрут по задачам

1. Посчитать прибыль по каждому заказу
2. Сложить прибыль по товарам
3. Найти убыточные товары
4. Посчитать долю маржи
5. Учесть комиссию площадки
6. Сравнить каналы по прибыли
7. Проверить эффект скидки
8. Учесть возвраты
9. Посчитать прибыль по категориям
10. Найти заказы ниже минимальной прибыли
11. Сравнить план и факт прибыли
12. Проверить нулевую себестоимость
13. Посчитать прибыль на клиента
14. Найти минимальную цену для цели по прибыли
15. Собрать прибыльный топ товаров
16. Посчитать прибыль после резерва
17. Разложить прибыль по дням
18. Собрать короткий маржинальный отчёт

Практические задачи

Задача 1. Посчитать прибыль по каждому заказу

Рабочий вопрос

Нужно увидеть не только выручку, но и деньги после себестоимости по каждому заказу.

Данные

```
CREATE TABLE orders(order_id INTEGER,  
revenue INTEGER, cost INTEGER);  
INSERT INTO orders VALUES (1, 900, 420),  
(2, 500, 250), (3, 300, 360);
```

Запрос

```
SELECT order_id, revenue - cost AS  
gross_profit
```

```
FROM orders  
ORDER BY order_id;
```

Ожидаемый результат

```
[[1, 480], [2, 250], [3, -60]]
```

Почему работает

Прибыль строки считается как revenue минус cost, без группировки.

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

Выручка может быть положительной даже у убыточного заказа, если себестоимость выше цены.

Самостоятельная проверка

Какой заказ убыточный и почему? Объясните ответ по ис-

ходным строкам и условию запроса.

Ответ: Заказ 3: 300 минус 360 даёт минус 60.

Задача 2. Сложить прибыль по товарам

Рабочий вопрос

Нужно понять, какие SKU дают суммарную прибыль, если один товар встречается в нескольких заказах.

Данные

```
CREATE TABLE order_items (sku TEXT,  
revenue INTEGER, cost INTEGER);  
INSERT INTO order_items  
VALUES ('A1', 500, 200), ('A1', 700, 300),  
('B2', 400, 380), ('C3', 300, 350);
```

Запрос

```
SELECT sku, SUM(revenue - cost) AS  
gross_profit  
FROM order_items  
GROUP BY sku  
ORDER BY gross_profit DESC, sku;
```

Ожидаемый результат

```
[["A1", 700], ["B2", 20], ["C3", -50]]
```

Почему работает

SUM(revenue - cost) складывает прибыль всех строк одного SKU.

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

Сортировка по выручке могла бы поставить выше товар с большой ценой, но слабой прибылью.

Самостоятельная проверка

Какой SKU самый прибыльный? Объясните ответ по исходным строкам и условию запроса.

Ответ: A1, потому что его прибыль равна 700.

Задача 3. Найти убыточные товары

Рабочий вопрос

Нужно выделить товары, у которых суммарная прибыль ниже нуля, чтобы не масштабировать минус.

Данные

```
CREATE TABLE order_items(sku TEXT,
revenue INTEGER, cost INTEGER);
INSERT INTO order_items
VALUES ('A1',500,200), ('B2',400,450),
('B2',100,120), ('C3',300,260);
```

Запрос

```
SELECT sku, SUM(revenue - cost) AS
gross_profit
FROM order_items
GROUP BY sku
HAVING SUM(revenue - cost) < 0
ORDER BY sku;
```

Ожидаемый результат

```
[[ "B2", -70 ]]
```

Почему работает

HAVING фильтрует группы товаров уже после расчёта суммарной прибыли.

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

WHERE revenue - cost < 0 нашёл бы отдельные минусовые строки, но не итог по SKU.

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.