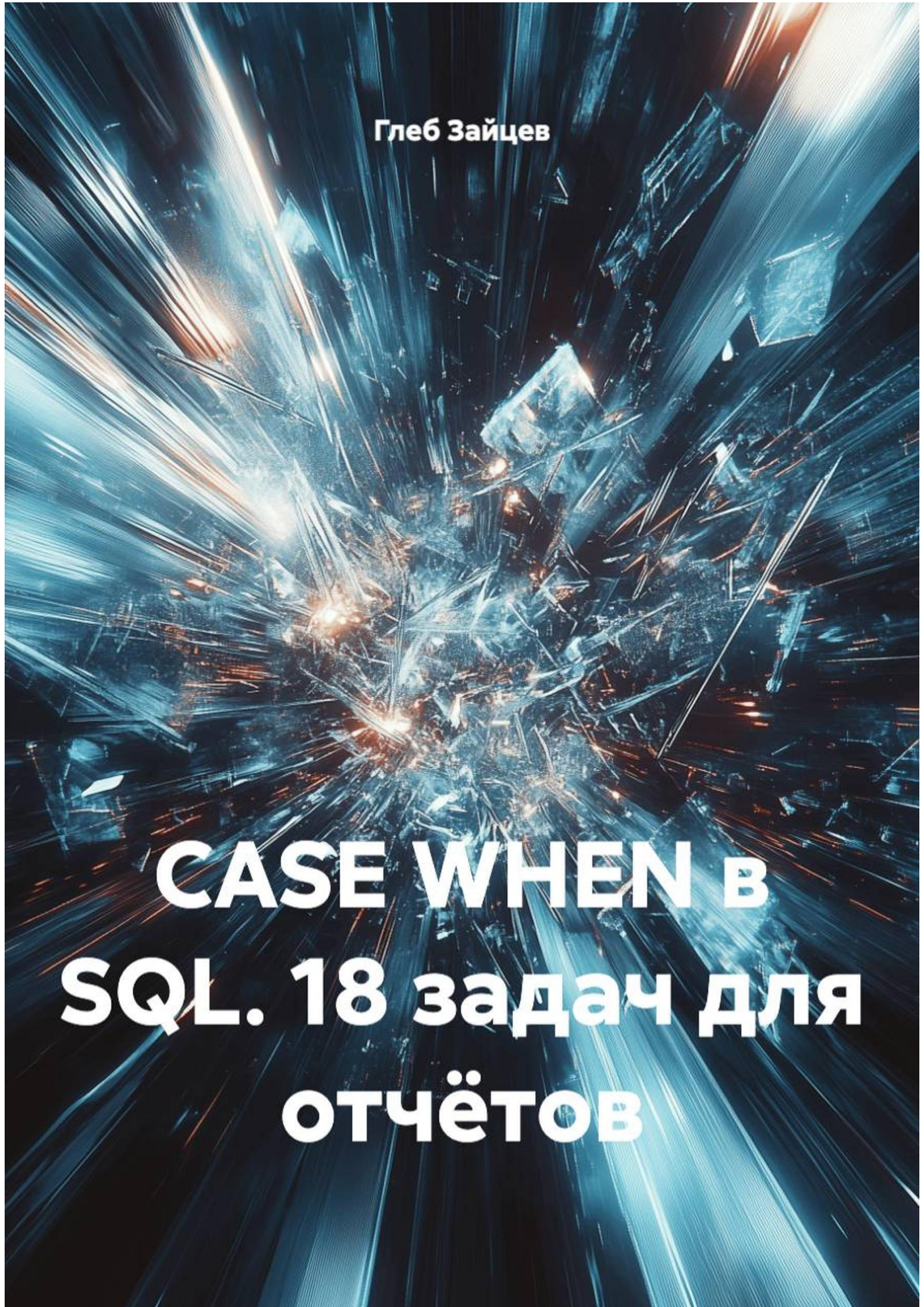




Глеб Зайцев

CASE WHEN в SQL. 18 задач для отчётов

Глеб Зайцев

**CASE WHEN в SQL.
18 задач для отчётов**

«Автор»

2026

Зайцев Г.

CASE WHEN в SQL. 18 задач для отчётов / Г. Зайцев —
«Автор», 2026

Практическая книга по CASE WHEN в SQL для аналитиков и начинающих разработчиков. Внутри 18 задач на маленьких таблицах: сегменты, статусы, корзины сумм, условные суммы, условные счётчики, приоритетная сортировка, флаги просрочки, скидки и итоговые отчёты. Все примеры запускаются в SQLite, данные встроены прямо в книгу.

© Зайцев Г., 2026

© Автор, 2026

Содержание

Перед первой задачей	5
Почему CASE нужен почти в каждом отчёте	5
Как читать задачи	6
Маршрут по задачам	7
Практические задачи	8
Задача 1. Разложить заказы по размеру чека	8
Задача 2. Нормализовать статусы заказа	10
Задача 3. Посчитать выручку по корзинам цены	11
Конец ознакомительного фрагмента.	12

Глеб Зайцев

CASE WHEN в SQL. 18 задач для отчётов

Перед первой задачей

Почему CASE нужен почти в каждом отчёте

Реальные отчёты редко показывают только сырые значения. Обычно нужно превратить суммы в корзины, статусы — в понятные подписи, а условия — в флаги.

CASE WHEN делает это прямо внутри SQL: он выбирает значение по условию и помогает строить отчёт без ручной правки в таблице после выгрузки.

В этой книге CASE рассматривается не как справочник синтаксиса, а как рабочий инструмент аналитика: от простых меток до условной агрегации.

Как читать задачи

В каждой задаче сначала сформулирована бизнес-потребность, затем идут маленькие таблицы, запрос, ожидаемый результат и объяснение.

Старайтесь перед запуском предсказать результат CASE: какое первое условие сработает и что попадёт в ELSE.

Все запросы рассчитаны на SQLite. Конструкция CASE стандартная, поэтому логика переносится в PostgreSQL, MySQL и другие SQL-системы.

Маршрут по задачам

- [1. Разложить заказы по размеру чека](#)
- [2. Нормализовать статусы заказа](#)
- [3. Посчитать выручку по корзинам цены](#)
- [4. Сложить только оплаченные и отменённые суммы](#)
- [5. Посчитать крупные заказы клиента](#)
- [6. Посчитать конверсию канала](#)
- [7. Отсортировать статусы по бизнес-приоритету](#)
- [8. Заменить пустой сегмент клиента](#)
- [9. Поставить флаг просрочки](#)
- [10. Собрать возрастные корзины покупателей](#)
- [11. Рассчитать скидку по правилу](#)
- [12. Посчитать средний оплаченный чек](#)
- [13. Сделать маленький pivot по статусам](#)
- [14. Пометить остатки на складе](#)
- [15. Выявить подозрительные суммы](#)
- [16. Группировать товары по маржинальности](#)
- [17. Выбрать действие по возврату](#)
- [18. Собрать финальный клиентский отчёт](#)

Практические задачи

Задача 1. Разложить заказы по размеру чека

Рабочий вопрос

Нужно подписать каждый заказ как маленький, средний или крупный, чтобы быстро увидеть структуру чеков.

Данные

```
CREATE TABLE orders(id INTEGER, total INTEGER);
INSERT INTO orders VALUES (1,150), (2,490), (3,900), (4,1200);
```

Запрос

```
SELECT id, total,
CASE
WHEN total < 300 THEN 'small'
WHEN total < 800 THEN 'medium'
ELSE 'large'
END AS check_size
FROM orders
ORDER BY id;
```

Ожидаемый результат

```
[[1, 150, "small"], [2, 490, "medium"], [3, 900, "large"], [4,
1200, "large"]]
```

Почему работает

CASE проверяет условия сверху вниз.
150 меньше 300, поэтому получает small.
900 не проходит первые два условия и попадает в ELSE large.

Где легко ошибиться

Порядок важен: если поставить total < 800 выше total < 300, маленькие чеки тоже станут medium.

Самостоятельная проверка

Добавьте заказ на 300. В какой сегмент он попадёт и почему?

Ответ: Он попадёт в medium, потому что условие small требует строго меньше 300.

Задача 2. Нормализовать статусы заказа

Рабочий вопрос

В базе статусы хранятся техническими словами, а в отчёте нужны короткие понятные русские подписи.

Данные

```
CREATE TABLE orders(id INTEGER, status TEXT);
INSERT INTO orders VALUES (1, 'paid'), (2, 'new'),
(3, 'cancelled'), (4, 'refunded');
```

Запрос

```
SELECT id, status,
CASE status
WHEN 'paid' THEN 'оплачен'
WHEN 'new' THEN 'новый'
WHEN 'cancelled' THEN 'отменён'
ELSE 'проверить'
END AS status_label
FROM orders
ORDER BY id;
```

Ожидаемый результат

```
[[1, "paid", "оплачен"], [2, "new", "новый"], [3, "cancelled",
"отменён"], [4, "refunded", "проверить"]]
```

Почему работает

Простая форма CASE сравнивает одно поле status с набором значений. Известные статусы получают человекочитаемые подписи. refunded не описан явно, поэтому попадает в ELSE.

Где легко ошибиться

Не оставляйте ELSE пустым в отчётах со статусами: новые значения должны быть заметны, а не превращаться в NULL.

Самостоятельная проверка

Добавьте явное правило для refunded. Какая подпись должна появиться?

Ответ: Можно вернуть подпись 'возврат', и строка 4 перестанет попадать в 'проверить'.

Задача 3. Посчитать выручку по корзинам цены

Рабочий вопрос

Нужно сгруппировать продажи не по точной цене, а по понятным ценовым корзинам.

Данные

```
CREATE TABLE sales(id INTEGER, price INTEGER, qty INTEGER);
INSERT INTO sales VALUES (1,99,2), (2,199,3), (3,499,1),
(4,1200,1);
```

Запрос

```
SELECT
CASE
WHEN price < 150 THEN 'до 150'
WHEN price < 500 THEN '150-499'
ELSE '500+'
END AS price_bucket,
SUM(price * qty) AS revenue
FROM sales
GROUP BY price_bucket
ORDER BY revenue DESC;
```

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.