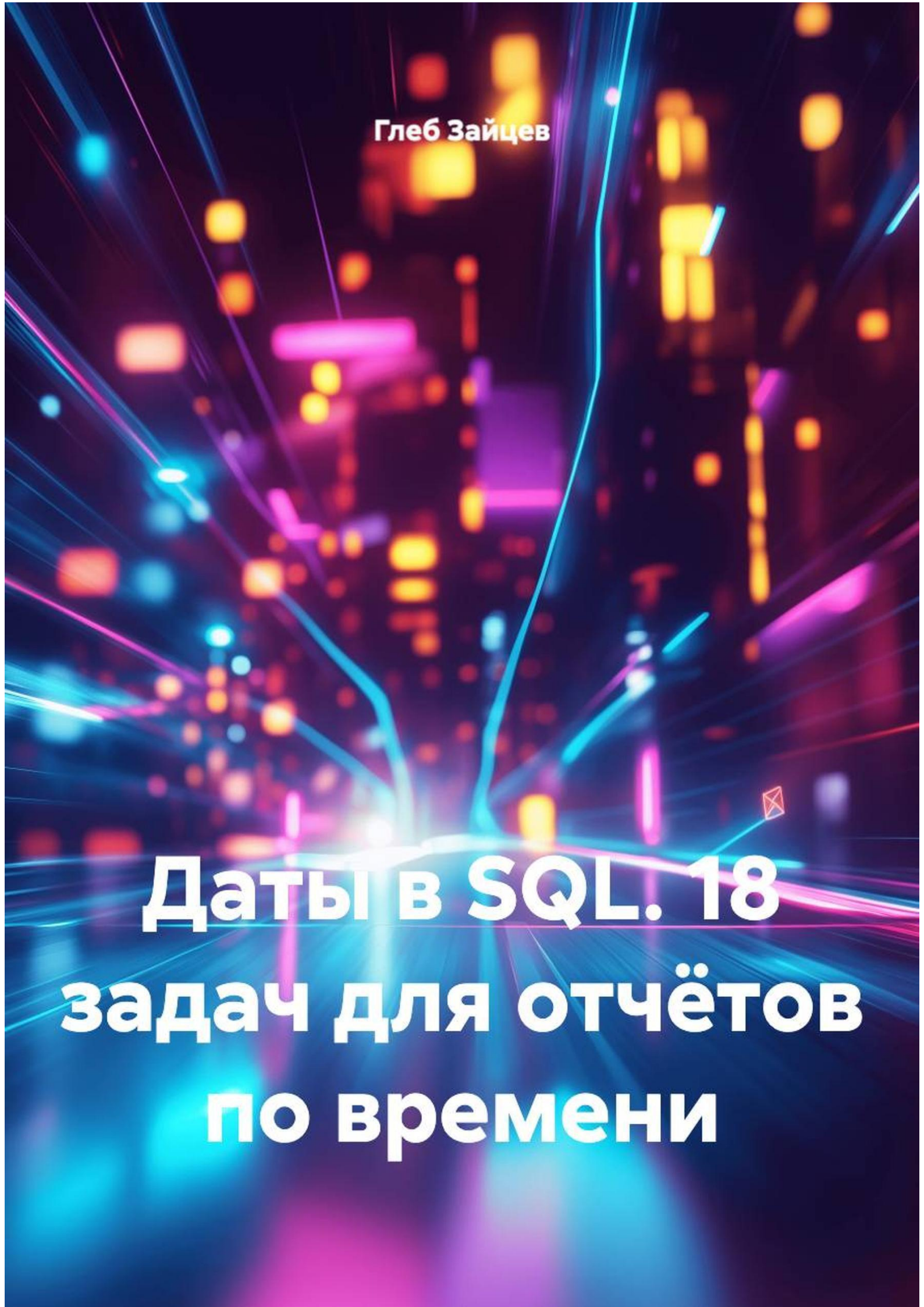




Глеб Зайцев

Даты в SQL. 18 задач для отчётов по времени

Глеб Зайцев

**Даты в SQL. 18 задач
для отчётов по времени**

«Автор»

2026

Зайцев Г.

Даты в SQL. 18 задач для отчётов по времени / Г. Зайцев —
«Автор», 2026

Практическая книга по работе с датами в SQL для аналитиков, менеджеров и начинающих разработчиков. Внутри 18 задач на маленьких таблицах: фильтры по дню и месяцу, группировка по дате, неделе и месяцу, первое и последнее событие, просрочки, длительность доставки, календарная таблица и проверки пустых дней. Все запросы запускаются в SQLite, данные встроены прямо в книгу.

© Зайцев Г., 2026

© Автор, 2026

Содержание

Перед первой задачей	5
Почему даты ломают отчёты	5
Как читать примеры	6
Маршрут по задачам	7
Практические задачи	8
Задача 1. Выбрать заказы за конкретный день	8
Задача 2. Отфильтровать месяц полуоткрытым интервалом	10
Задача 3. Сгруппировать продажи по дням	12
Конец ознакомительного фрагмента.	13

Глеб Зайцев

Даты в SQL. 18 задач для отчётов по времени

Перед первой задачей

Почему даты ломают отчёты

В отчётах по времени опасны не сами функции, а границы периода: включили ли последний день, потеряли ли вечерние заказы, смешали ли дату и timestamp.

Хороший SQL-отчёт по датам сначала отвечает на простой вопрос: что является событием, какой период выбран и какие строки должны попасть внутрь.

Эта книга тренирует практические ситуации: продажи за день, месяц, неделю, просрочки, длительность между событиями и пустые даты в календаре.

Как читать примеры

Все данные маленькие, чтобы результат можно было проверить глазами до запуска запроса.

В SQLite удобно использовать `date()`, `strftime()` и `julianday()`. В других СУБД названия функций могут отличаться, но логика периодов остаётся той же.

Особое внимание уделяйте полуоткрытым интервалам: начало включаем, конец следующего периода не включаем. Это безопаснее, чем пытаться угадать последний timestamp дня.

Маршрут по задачам

- [1. Выбрать заказы за конкретный день](#)
- [2. Отфильтровать месяц полуоткрытым интервалом](#)
- [3. Сгруппировать продажи по дням](#)
- [4. Собрать продажи по месяцам](#)
- [5. Посчитать заказы по дням недели](#)
- [6. Найти заказы за последние семь дней](#)
- [7. Найти первую и последнюю покупку клиента](#)
- [8. Посчитать дни от регистрации до первой покупки](#)
- [9. Найти просроченные счета](#)
- [10. Посчитать среднюю длительность доставки](#)
- [11. Разделить заказы на новые и старые](#)
- [12. Нормализовать timestamp до даты](#)
- [13. Построить когорты регистраций и покупок](#)
- [14. Посчитать пустые дни через календарь](#)
- [15. Собрать недельный отчёт](#)
- [16. Найти первый день месяца](#)
- [17. Проверить события после дедлайна](#)
- [18. Собрать финальный календарный KPI](#)

Практические задачи

Задача 1. Выбрать заказы за конкретный день

Рабочий вопрос

Нужно получить все заказы, которые были созданы 5 сентября, даже если внутри дня указано разное время.

Данные

```
CREATE TABLE orders(id INTEGER, created_at TEXT, total
INTEGER);
INSERT INTO orders VALUES
(1, '2026-09-05 09:10:00', 500),
(2, '2026-09-05 18:45:00', 700),
(3, '2026-09-06 01:20:00', 300);
```

Запрос

```
SELECT id, total
FROM orders
WHERE date(created_at) = '2026-09-05'
ORDER BY id;
```

Ожидаемый результат

```
[[1, 500], [2, 700]]
```

Почему работает

`date(created_at)` отбрасывает время и оставляет календарный день.

Первые две строки относятся к 5 сентября.

Заказ 3 создан уже 6 сентября и не попадает в результат.

Где легко ошибиться

Сравнение `created_at = '2026-09-05'` не найдёт строки с временем, потому что полное значение отличается от короткой даты.

Самостоятельная проверка

Почему заказ в 18:45 попал в выборку, хотя время явно не равно полуночи?

Ответ: Потому что `date(created_at)` преобразует timestamp к дате 2026-09-05.

Задача 2. Отфильтровать месяц полуоткрытым интервалом

Рабочий вопрос

Нужно посчитать сентябрьскую выручку так, чтобы не потерять вечерние заказы последнего дня месяца.

Данные

```
CREATE TABLE orders(id INTEGER, created_at TEXT, total
INTEGER);
INSERT INTO orders VALUES
(1, '2026-08-31 23:50:00', 100),
(2, '2026-09-01 00:05:00', 400),
(3, '2026-09-30 22:10:00', 600),
(4, '2026-10-01 00:01:00', 900);
```

Запрос

```
SELECT SUM(total) AS september_revenue
FROM orders
WHERE created_at >= '2026-09-01'
AND created_at < '2026-10-01';
```

Ожидаемый результат

```
[[1000]]
```

Почему работает

Начало сентября включено условием `>=`.
Начало октября не включено условием `<`.
Внутри остаются заказы на 400 и 600.

Где легко ошибиться

Условие `created_at <= '2026-09-30'` может потерять строки 30 сентября с указанным временем.

Самостоятельная проверка

Какой заказ специально проверяет правую границу периода?

Ответ: Заказ 4 на 2026-10-01 00:01:00 показывает, что октябрь не попадает в сентябрьский отчёт.

Задача 3. Сгруппировать продажи по дням

Рабочий вопрос

Нужно получить дневной отчёт: сколько заказов и какая выручка были в каждый день.

Данные

```
CREATE TABLE orders(id INTEGER, created_at TEXT, total
INTEGER);
INSERT INTO orders VALUES
(1, '2026-09-01 10:00:00', 300),
(2, '2026-09-01 19:30:00', 200),
(3, '2026-09-02 11:15:00', 800);
```

Запрос

```
SELECT date(created_at) AS day,
COUNT(*) AS orders_count,
SUM(total) AS revenue
FROM orders
```

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.