

The background of the entire image is an abstract composition of vibrant green light trails. These trails are composed of numerous thin, curved lines that sweep across the frame, creating a sense of dynamic movement and energy. Interspersed among these lines are small, bright green dots and larger, soft, out-of-focus circular glows, reminiscent of light painting or a long-exposure photograph of light sources. The overall color palette is a range of greens, from deep forest greens to bright, almost white highlights where the light trails are most intense.

Глеб Зайцев

**SQL на практике. 60 задач
с решениями на данных
магазина**

Глеб Зайцев

**SQL на практике. 60 задач с
решениями на данных магазина**

«Автор»

2026

Зайцев Г.

SQL на практике. 60 задач с решениями на данных магазина /
Г. Зайцев — «Автор», 2026

Практикум для тех, кто знает основы SELECT и хочет научиться собирать запрос из условия задачи. 60 упражнений на одной учебной SQLite-базе магазина: покупатели, товары, заказы, позиции заказов и платежи. Шесть глав: фильтрация, агрегирование, JOIN, подзапросы, CTE и оконные функции. Сначала — задания; затем — подсказки, решения, ожидаемое число строк и пример результата. Отдельно разобраны регистр кириллицы, расхождения между заказом и оплатой, пропущенные даты и отличие трёх строк от трёх календарных дней. Полный код создания базы и инструмент запуска запросов включены в книгу. Для практики нужен компьютер с Python 3 и встроенным модулем sqlite3. Отдельный архив, платный сервис или аккаунт не требуются. Данные синтетические; это учебные упражнения, а не готовая финансовая отчётность.

© Зайцев Г., 2026

© Автор, 2026

Содержание

Как работать с практикумом	6
Подготовка учебной базы	7
Схема и правила расчётов	8
Глава 1 Базовые выборки	9
Глава 2 Агрегации	11
Глава 3 Связи таблиц	13
Конец ознакомительного фрагмента.	15

Глеб Зайцев

SQL на практике. 60 задач с решениями на данных магазина

SQL на практике

60 задач с решениями на данных магазина

Глеб Зайцев

SQLite · самостоятельная практика

Первое книжное издание 2026

Для выполнения заданий нужен компьютер с Python 3 и модулем sqlite3. Весь код создания учебной базы включён в книгу. Отдельный архив, платный сервис и доступ к чужому серверу не требуются.

Как работать с практикумом

Эта книга поможет потренировать SQL на одной связанной базе интернет-магазина. В ней 60 задач: от отбора товаров до рейтингов и накопительных сумм. Сначала решайте самостоятельно, затем сверяйтесь с подсказкой и эталонным запросом во второй части.

Практикум предназначен для тех, кто уже знаком с таблицами и базовым SELECT. Он не заменяет полный учебник по проектированию баз. Краткие введения объясняют подход к каждой группе задач, а разборы обращают внимание на ошибки в определении метрик.

Все имена, товары, заказы и платежи в наборе учебные. Это не данные реальных покупателей и не выгрузка продавца. Распределения намеренно простые и не отражают рынок. Регистрации и заказы создаются независимо; встречаются заказы раньше регистрации, что отдельно учтено в задании об интервале до первой покупки.

Задачи не требуют менять исходные строки. Сохраните свой запрос в отдельный файл `answer.sql` и выполняйте его только на учебной базе. Приведённая в приложении команда открывает базу в режиме чтения.

Примеры рассчитаны на диалект SQLite. Перенос на PostgreSQL, MySQL или другую СУБД потребует адаптации функций дат и некоторых правил группировки. Все 60 эталонов проверены на SQLite 3.53.1. Это техническая проверка примеров, а не обещание трудоустройства или уровня квалификации.

Содержание

- Подготовка учебной базы
- Схема и правила расчётов
- Часть первая Задачи
 - Базовые выборки
 - Агрегации
 - Связи таблиц
 - Даты и периоды
 - Подзапросы и CTE
 - Оконные функции
- Часть вторая Подсказки и решения
- Приложение А Создание базы
- Приложение Б Выполнение запросов
- Приложение В Проверки спорных случаев
- Документация

Подготовка учебной базы

Создайте отдельную пустую папку для практикума. В текстовом редакторе создайте в ней файл `create_shoplab.py` в кодировке UTF-8. Скопируйте в него весь код из приложения А: от первой строки импорта до последнего `print`. Не включайте заголовки книги и номера страниц. Сохраняйте обычные прямые кавычки и переносы строк. В этой электронной версии каждая строка Python начинается с левого края: код не зависит от абзацных отступов читалки.

В терминале, открытом в этой папке, выполните команду ниже. Если на вашем компьютере Python запускается как `python3` или `py`, замените только первое слово команды.

```
python create_shoplab.py
```

Программа создаст `shoplab.sqlite3` рядом с файлом и выведет контрольные количества строк. Если база уже существует, она остановится и не перезапишет её. Для повторного чистого запуска используйте новую пустую папку.

customers — 30 строк.

products — 18 строк.

orders — 72 строк.

order_items — 144 строк.

payments — 60 строк.

Python и встроенная в него SQLite могут иметь разные версии. Для проверки поддержки оконных функций выполните следующий короткий пример в отдельном файле `check_sqlite.py`. Он не создаёт файлов базы и должен напечатать `[(1,)]`.

```
import sqlite3
db = sqlite3.connect(":memory:")
print(db.execute("SELECT row_number() OVER ()").fetchall())
db.close()
```

Если Python не найден, установите актуальную стабильную версию Python 3 с официального сайта python.org. Если проверка выдаёт ошибку синтаксиса у `OVER`, используемая библиотека SQLite слишком старая для шестой главы. Текст книги можно читать с телефона, но выполнять практические задания удобнее на компьютере.

Первый запрос

Создайте `query.py` из приложения Б. В файле `answer.sql` сохраните один запрос без обрамляющих кавычек:

```
SELECT COUNT(*) AS customers_count
FROM customers;
python query.py answer.sql
```

Ожидается заголовок `customers_count`, значение 30 и строка Строк: 1. Затем заменяйте содержимое `answer.sql` своим решением. Команда показывает весь результат и число строк; автоматическую оценку она не выставляет.

В некоторых читалках копирование длинного кода неудобно. Откройте приобретённую книгу на компьютере в доступном текстовом формате. При переносе проверяйте переносы строк и прямые кавычки. Если читалка добавила отступы абзацев, уберите пробелы только в начале строк кода; внутри строк пробелы сохраняйте. Код нельзя копировать как картинку.

Схема и правила расчётов

customers

Покупатели. `customer_id` — первичный ключ, `customer_name` — учебное имя, `city` — город, `segment` — retail, pro или premium, `signup_date` — дата регистрации. `email_opt_in` равен 1 при согласии на рассылку и 0 иначе.

products

Товары. `product_id` — ключ, `product_name` — название, `category` — office, home, electronics или sport. `price_rub` — текущая цена в целых рублях, `stock_qty` — учебный остаток, `is_active` — признак активности 0 или 1.

orders

Заказы. `order_id` — ключ, `customer_id` ссылается на покупателя, `order_date` — дата. `status` принимает paid, shipped, delivered или cancelled. `channel` принимает market, site или partner.

order_items

Строки заказов. Составной ключ `order_id` и `product_id` допускает товар в заказе один раз. `quantity` — количество, `unit_price_rub` — цена за единицу именно в этом заказе. Она может отличаться от текущей цены товара.

payments

Платежи. `payment_id` — ключ; уникальный `order_id` означает не более одного платежа на заказ. `paid_at` — дата, `amount_rub` — сумма в целых рублях, `method` — card, sbr или invoice.

Связи: один покупатель имеет несколько заказов; один заказ имеет несколько товарных строк и ноль или один платёж. Одна строка заказа относится к одному товару. Внешние ключи включены при создании базы.

В этом наборе отменённые заказы не имеют платежей; остальные оплачены целиком. Возвраты, налоги, комиссии и себестоимость не моделируются. Слово «выручка» в заданиях означает только сумму учебных платежей либо равную ей стоимость строк неотменённых заказов. Чистая прибыль здесь не рассчитывается.

Если условие не исключает отменённые заказы, учитывайте их. В задачах на заказы и товарные строки это специально оговорено. Платёж не следует суммировать после размножающего его соединения с товарными строками.

Платежи охватывают январь — август 2026 года; август неполный. Дни и месяцы без строк не добавляются автоматически. Временем отсчёта для «последних 30 дней» служит максимальная дата набора, а не сегодняшний день.

Глава 1 Базовые выборки

Начните с набора строк. WHERE оставляет только строки, удовлетворяющие условию, а SELECT задаёт столбцы результата. ORDER BY нужен, если важна последовательность; без него порядок не обещан.

В LIMIT сначала определите сортировку. Для устойчивого результата добавляйте идентификатор после основной метрики, если условие не требует сохранять ничьи. NULL проверяют через IS NULL, а не через равенство.

Подсказки и эталоны находятся во второй части под теми же номерами. Поля в строке «Результат» задают форму ответа; выводите их в указанном порядке. Если в условии не указан порядок строк, используйте сортировку из эталона для удобства сравнения.

Задача 1 1

Выведите все активные товары: идентификатор, название, категорию и цену. Отсортируйте сначала по категории, затем по цене по возрастанию.

Результат: product_id, product_name, category, price_rub.

Задача 1 2

Найдите покупателей из Москвы и Казани. Покажите имя, город и сегмент, отсортируйте по городу и имени.

Результат: customer_name, city, segment.

Задача 1 3

Покажите идентификатор заказа, покупателя, дату и статус для заказов за февраль 2026 года. Включите весь месяц. Отсортируйте по дате и идентификатору заказа.

Результат: order_id, customer_id, order_date, status.

Задача 1 4

Получите список уникальных городов покупателей в алфавитном порядке.

Результат: city.

Задача 1 5

Найдите пять самых дорогих активных товаров.

Результат: product_name, category, price_rub.

Задача 1 6

Выведите товары с остатком меньше 10 единиц, но больше нуля.

Результат: product_id, product_name, stock_qty.

Задача 1 7

Найдите товары, в названии которых содержится подстрока «набор» точно в указанном регистре. Выведите `product_id`, `product_name` и `price_rub`, отсортировав по `product_id`.

Результат: `product_id`, `product_name`, `price_rub`.

Задача 1 8

Покажите все отменённые заказы из канала `site`.

Результат: `order_id`, `customer_id`, `order_date`.

Задача 1 9

Разделите товары на ценовые группы: цена меньше 500 рублей — `low`; от 500 включительно до 1000 не включительно — `mid`; от 1000 включительно — `high`. Выведите название, цену и группу, отсортировав по цене и `product_id`.

Результат: `product_name`, `price_rub`, `price_band`.

Задача 1 10

Выведите покупателей, которые согласились на email-рассылку и зарегистрировались не раньше 1 марта 2026 года.

Результат: `customer_id`, `customer_name`, `signup_date`.

Глава 2 Агрегации

GROUP BY меняет единицу наблюдения: вместо отдельных строк вы получаете одну строку на группу. WHERE фильтрует исходные строки, HAVING — уже посчитанные группы.

Перед SUM определите, что именно складывается: стоимость строк заказа, полученные платежи или количество единиц. COUNT(*) считает строки, COUNT(поле) пропускает NULL. Для долей используйте 100.0, чтобы избежать целочисленного деления.

Подсказки и эталоны находятся во второй части под теми же номерами. Поля в строке «Результат» задают форму ответа; выводите их в указанном порядке. Если в условии не указан порядок строк, используйте сортировку из эталона для удобства сравнения.

Задача 2 1

Посчитайте количество покупателей в каждом городе.

Результат: city, customers_count.

Задача 2 2

Найдите среднюю цену активных товаров в каждой категории, округлив до двух знаков.

Результат: category, avg_price_rub.

Задача 2 3

Посчитайте число заказов каждого статуса.

Результат: status, orders_count.

Задача 2 4

Посчитайте сумму полученных платежей по каждому способу оплаты.

Результат: method, payments_count, revenue_rub.

Задача 2 5

Найдите товары, для которых сумма quantity во всех строках заказов не меньше 8. В этой задаче учитывайте и отменённые заказы: измеряем объём заказанных, а не оплаченных единиц.

Результат: product_id, product_name, units_sold.

Задача 2 6

Рассчитайте стоимость строк каждого заказа как сумму quantity × unit_price_rub. Учитывайте все статусы. Отсортируйте по сумме по убыванию, затем по order_id.

Результат: order_id, order_total_rub.

Задача 2 7

Покажите покупателей, у которых не меньше трёх заказов.

Результат: customer_id, customer_name, orders_count.

Задача 2 8

Найдите минимальную и максимальную цену внутри каждой категории.

Результат: category, min_price_rub, max_price_rub.

Задача 2 9

Рассчитайте долю отменённых заказов в процентах с точностью до одного знака.

Результат: cancellation_rate_pct.

Задача 2 10

Рассчитайте средний чек по фактически полученным платежам.

Результат: average_payment_rub.

Глава 3 Связи таблиц

Соединение связывает строки по ключу. INNER JOIN оставляет совпадения, LEFT JOIN сохраняет все строки слева и подставляет NULL там, где пары нет.

Проверяйте кратность соединения. Заказ имеет несколько товарных строк, но не больше одного платежа в этой модели. Если присоединить обе таблицы к заказу, сумма платежа повторится. Сначала агрегируйте нужную сторону или разделите расчёты.

Подсказки и эталоны находятся во второй части под теми же номерами. Поля в строке «Результат» задают форму ответа; выводите их в указанном порядке. Если в условии не указан порядок строк, используйте сортировку из эталона для удобства сравнения.

Задача 3 1

Покажите каждый заказ вместе с именем покупателя и городом.
Результат: order_id, order_date, status, customer_name, city.

Задача 3 2

Соберите детализацию строк заказа: заказ, товар, количество, цена и сумма строки.
Результат: order_id, product_name, quantity, unit_price_rub, line_total_rub.

Задача 3 3

Найдите покупателей, у которых ещё нет ни одного заказа.
Результат: customer_id, customer_name, city.

Задача 3 4

Найдите товары, которые ни разу не попадали в заказ.
Результат: product_id, product_name, category.

Задача 3 5

Покажите заказы и способ оплаты. Сохраните в результате отменённые заказы без платежа.
Результат: order_id, status, method, amount_rub.

Задача 3 6

Рассчитайте стоимость строк неотменённых заказов по категориям. В учебной базе каждому такому заказу соответствует полный платёж; это допущение позволяет назвать сумму выручкой. Отсортируйте по выручке по убыванию и категории.

Результат: category, revenue_rub.

Задача 3 7

Покажите покупателей и их общую подтверждённую оплату, включая покупателей с нулевой суммой.

Результат: customer_id, customer_name, paid_rub.

Задача 3 8

Найдите заказы с существующим платежом, у которых сумма строк отличается от amount_rub. Заказы без платежа не включайте. Отсортируйте по order_id.

Результат: order_id, items_total_rub, amount_rub.

Задача 3 9

Для каждого канала посчитайте число заказов и сумму полученных платежей.

Результат: channel, orders_count, paid_rub.

Задача 3 10

Выведите топ-10 пар «покупатель — товар» по потраченной сумме.

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.