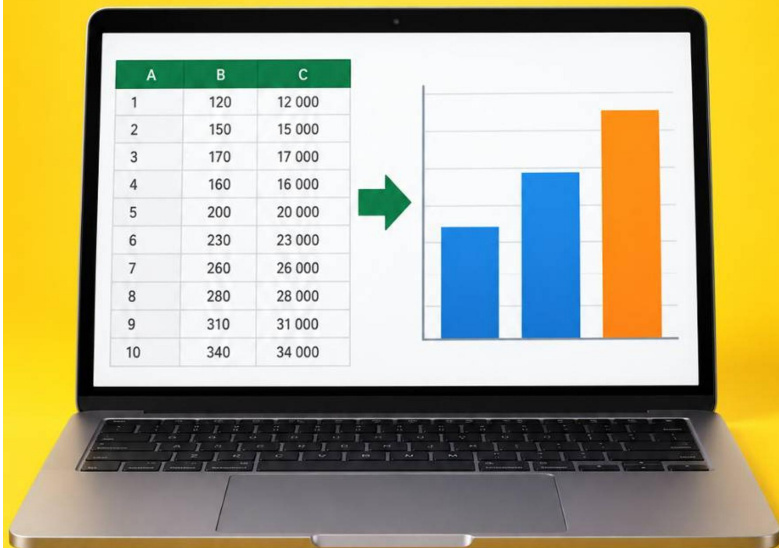


ПРОФЕССИЯ: АНАЛИТИК ДАННЫХ



АНТОН ЗОЛОТАВИН

Антон Золотавин

Профессия: Аналитик данных

<https://litres.ru/74621103>

SelfPub; 2026

Аннотация

Хотите освоить востребованную профессию и научиться находить в цифрах ответы на вопросы бизнеса? «Профессия: Аналитик данных» поможет пройти путь от первых таблиц до самостоятельного проекта для портфолио.

Вы освоите обработку данных в Excel, научитесь писать SQL-запросы, выполнять расчёты на Python и строить понятные дашборды. На примере учебного интернет-магазина разберётесь, почему заказы растут, а доход снижается, как сравнивать рекламные каналы и какие ошибки искажают отчёты.

Внутри — пошаговые объяснения, практические задания с решениями, готовые данные и проверочные результаты. Отдельные главы посвящены общению с заказчиком, оформлению портфолио, резюме и поиску первой работы.

Содержание

Введение. От таблицы к решению	6
Как проходить практикум	8
Что считать законченной работой	10
Глава 1. Работа аналитика и её результат	11
За что платят специалисту	11
Как выглядит рабочая неделя	13
Доход и условия работы	14
Практическая работа 1. Выберите рабочую роль	16
Глава 2. Постановка задачи и язык показателей	17
Уточняем запрос руководителя	17
Единица наблюдения	19
Договоримся о деньгах	20
Паспорт метрики	22
Практическая работа 2. Согласуйте результат	23
Глава 3. Получение данных и первая проверка	24
Что находится в комплекте	24
Словарь исходных полей	26
Принять файл — отдельная операция	28
Пять проверок до анализа	29
Дубликат записи и повторное событие	30
Практическая работа 3. Протокол приёмки	31
Организуем рабочую книгу	32

Формулы на уровне заказа	33
Проверяем формулы на нескольких строках	35
Сводная таблица без ловушек	36
Поиск по справочнику	38
Практическая работа 4. Первая записка	39
Записываем последовательность преобразований	41
Типы и локаль	42
Объединение и добавление	43
Обновление и проверка результата	44
Практическая работа 5. Испытайте обновление	45
Подключаем учебную базу	46
Выбираем нужные поля	48
Фильтруем даты и статусы	50
Считаем итоги	52
Практическая работа 6. Напишите первые запросы	55
Глава 7. Объединения и модель данных	56
Как соединяются таблицы	56
Сначала агрегируем, затем соединяем расходы	59
Заказы и позиции	61
Конец ознакомительного фрагмента.	62

Антон Золотавин

Профессия:

Аналитик данных

Профессия:

Аналитик данных

Антон Золотавин

Введение. От таблицы к решению

В понедельник руководитель интернет-магазина открывает отчёт: заказов стало больше, а денег после расходов осталось меньше. В отделе рекламы говорят, что привели новую аудиторию. В отделе продаж предлагают увеличить скидку. Финансист просит остановить кампании. У всех есть числа, но они описывают разные части происходящего. Аналитику предстоит собрать их в одну проверяемую картину.

Этой работе посвящена книга. Вы научитесь получать данные, приводить их в порядок, рассчитывать показатели и объяснять результат людям, которым нужно принять решение. Excel поможет быстро разобраться в таблицах. SQL позволит извлекать и объединять записи. Python пригодится для повторяемой обработки и проверки расчётов. На дашборде вы покажете главное так, чтобы пользователь понимал смысл показателя, период и ограничения.

Учебный проект проходит через всю книгу. В учебном магазине «Маяк» продаются товары для дома, кухни и спорта. Мы исследуем первое полугодие 2025 года. Все записи созданы специально для практикума и не описывают настоящую компанию. В комплекте есть CSV, готовая база SQLite, скрипт создания данных, расчётный скрипт и контрольные результаты. Никаких сведений о реальных покупателях в них нет.

Основной вопрос проекта: почему увеличение количества заказов не привело к сопоставимому росту маржинального дохода после маркетинга? Чтобы ответить, недостаточно найти падающий график. Придётся договориться о смысле выручки, исключить повторные записи, проверить статусы, сопоставить каналы и разобрать составляющие расходов. Итогом станет небольшой аналитический продукт: подготовленные данные, запросы, воспроизводимый расчёт, дашборд и записка для руководителя.

Как проходить практикум

Читайте главы по порядку, а код вводите в собственном проекте. Сначала повторите небольшой пример и разберите каждую строку. После этого выполните задание без подсказки. Сравнивайте с решением только после собственной попытки. Умение узнавать правильный код на странице ещё не означает, что вы сможете написать его на работе.

В книге встречаются три вида чисел. Контрольные результаты относятся к приложенным файлам и должны совпасть при соблюдении условий. Малые условные примеры объясняют отдельный приём и не входят в итоги магазина. Сведения о рынке труда привязаны к источнику и дате проверки; они не обещают определённую зарплату. Это различие помогает не переносить учебное допущение в реальную задачу.

Для работы достаточно компьютера, табличного редактора, Python и редактора SQL. Для SQL выбран SQLite: база хранится в одном файле, сервер устанавливать не требуется. Этот выбор сокращает подготовку первого проекта. В компаниях вы можете встретить PostgreSQL, ClickHouse и другие системы; их различия разобраны отдельно. Визуализацию можно собрать в DataLens или в другом доступном BI-инструменте, следуя спецификации из книги. Подписка на конкретный сервис не является условием выполнения расчётов.

Рассчитывайте время по своему темпу. Одному читателю понадобится больше занятий на формулы, другому — на программирование. Полезнее закончить один проверяемый этап, чем провести условный час за экраном. После занятия сохраняйте рабочий файл, кратко записывайте, что получилось, и формулируйте следующий вопрос. Если код не работает, сохраните полный текст ошибки и минимальный пример. Возвращаться к задаче с этими записями гораздо легче.

Не пытайтесь сразу освоить все инструменты отрасли. На первом круге важнее понимать структуру таблицы, способ расчёта и смысл вывода. Часть задач можно решить несколькими средствами. Мы намеренно пересчитаем одни и те же итоги в SQL и Python: независимая сверка обнаруживает ошибки, которые остаются незаметными внутри одного красивого отчёта.

Что считать законченной работой

К концу практикума у вас должна быть папка, которую другой человек сможет открыть и понять без устных пояснений. В ней лежат описание задачи, словарь полей, исходные данные, правила очистки, SQL-запросы, программа расчёта, итоговая витрина и аналитическая записка. Дашборд сопровождается инструкцией: какие фильтры работают, как обновляются данные и что означают показатели.

Отдельно сохраните перечень ограничений. У магазина нет всех данных настоящего бизнеса: отсутствуют зарплаты сотрудников, аренда, налоги, история фактических платежей и отдельный журнал возвратов. Поэтому наш итог после маркетинга нельзя называть чистой прибылью. В упражнении все завершённые заказы имеют одну товарную позицию. Позже мы разберём переход к нескольким позициям и покажем, почему он меняет расчёты.

Эти ограничения не делают проект бесполезным. Они задают границы ответа. Вы сможете показать, какие каналы и расходы связаны с изменением результата, а затем предложить дополнительную проверку. Хороший аналитик умеет остановить вывод там, где заканчиваются данные. На следующем шаге он запрашивает недостающую информацию или проектирует эксперимент.

Глава 1. Работа аналитика и её результат

За что платят специалисту

Аналитик данных помогает организации уменьшать неопределённость. Например, руководителю нужно выбрать товары для закупки, службе поддержки — оценить очередь обращений, а продуктовой команде — понять, где пользователи прекращают оформление заказа. Число строк в выгрузке само по себе никому не помогает. Пользу даёт связь между вопросом, надёжным расчётом и действием.

Обычная задача состоит из нескольких договорённостей. Кто будет использовать результат? Какое решение он принимает? Когда оно должно быть принято? Какие данные доступны к этому сроку? Что произойдёт, если расчёт ошибочен? Ответы определяют и глубину анализа, и формат выдачи. Для утреннего совещания может хватить проверенной таблицы. Для изменения кредитной политики потребуется совсем другой уровень проверки и согласования.

Разделение профессий зависит от компании. ВІ-аналитик обычно уделяет больше внимания модели данных, витринам и отчётности. Продуктовый аналитик исследует поведение

пользователей, метрики и эксперименты. Бизнес-аналитик часто описывает требования и процессы. Data Scientist может строить прогнозные модели. Эти роли пересекаются, поэтому название вакансии нужно сопоставлять с задачами и требованиями, а не воспринимать как точное определение.

Начинающему специалисту полезно выбрать ограниченную область ответственности: регулярный отчёт продаж, качество данных по заявкам, анализ рекламных каналов. Такая область позволяет увидеть весь цикл работы. Если сразу обещать финансовую стратегию, машинное обучение и автоматизацию хранилища, будет трудно проверить, какая часть результата действительно ваша и насколько она надёжна.

Как выглядит рабочая неделя

Представим условную неделю аналитика в магазине. В понедельник он проверяет обновление отчёта и замечает снижение доли завершённых заказов. Во вторник уточняет, не изменились ли статусы в учётной системе. В среду собирает разрезы по каналам и городам. В четверг обсуждает результат с операционным руководителем. В пятницу добавляет проверку в обработку, чтобы похожая проблема не прошла незамеченной в будущем.

Большая часть этой работы не похожа на эффектную демонстрацию нейросети. Нужно читать названия полей, искать ответственного за выгрузку, сверять итоги и возвращаться к определениям. Это нормальная часть профессии. Автоматизация полезна, когда процесс уже понятен; автоматизированная ошибка быстрее распространяется по отчётам.

Есть и исследовательские задачи без заранее известного ответа. Например, вы проверяете, отличаются ли возвраты у новых и повторных покупателей. Исследование может закончиться выводом, что имеющихся данных недостаточно. Такой результат полезен, если вы объяснили, чего именно не хватает, сколько стоит дополнительный сбор и какое решение пока следует отложить.

Доход и условия работы

На странице профессии hh.ru при проверке 14 октября 2026 года в московском блоке были показаны 95 700–113 100 рублей, а среди отдельных стартовых вакансий встречались предложения от 37 500 и от 65 000 рублей [1]. Это снимок конкретной выдачи: он не задаёт общероссийскую ставку, не описывает все вакансии и не гарантирует доход выпускнику. Условия налогообложения и состав вознаграждения нужно читать в конкретном предложении.

Для собственного ориентира соберите таблицу из двадцати подходящих вакансий. Зафиксируйте дату, город, формат работы, требуемый опыт, обязанности, инструменты и указанную зарплату. Разделяйте суммы до и после налогов, фиксированную часть и бонусы. Не присваивайте вакансиям без зарплаты ноль: это отсутствие наблюдения, а не бесплатная работа. Медиану рассчитывайте только по сопоставимым значениям.

В штате обычно есть доступ к внутренним данным и коллегам, но есть и зависимость от процессов компании. На удалённой работе добавляется необходимость ясно фиксировать договорённости. Фриланс требует самостоятельно оценивать объём, искать заказчиков, ограничивать число правок и организовывать передачу результата. Начинающему специалисту особенно важен человек, который сможет про-

верить расчёт и объяснить ошибку; это стоит учитывать наряду с размером оплаты.

Практическая работа 1.

Выберите рабочую роль

Создайте файл «Карта профессии». Опишите три задачи, которые вам интересно решать, и для каждой назовите будущего пользователя результата. Например: «Руководителю продаж нужен еженедельный отчёт по завершённым заказам, чтобы распределять нагрузку». Затем перечислите, каких данных и навыков требует каждая задача. Пока не добавляйте инструменты, которые не связаны с выбранными задачами.

Изучите несколько актуальных вакансий своего региона или подходящей удалённой работы. Выпишите конкретные действия: «писать SQL-запросы», «согласовывать метрики», «строить регулярную отчётность». Если требование непонятно, сформулируйте вопрос для изучения. Результатом должна стать небольшая карта пробелов, а не бесконечный список технологий.

Критерий завершения прост: вы можете за минуту объяснить, какую работу хотите выполнять, кому она помогает и какой первый проект продемонстрирует вашу готовность. Для книги таким проектом станет анализ магазина «Маяк». В следующей главе мы превратим его общий вопрос в понятные требования.

Глава 2. Постановка задачи и язык показателей

Уточняем запрос руководителя

Фраза «сделайте дашборд по продажам» описывает желаемый предмет, но ещё не задачу. Спросите, какие решения человек принимает каждую неделю и где ему сейчас не хватает информации. Если он хочет видеть результат рекламы, уточните, собирается ли перераспределять бюджет, проверять подрядчика или планировать денежный поток. Для этих решений понадобятся разные показатели.

В проекте «Маяк» заказчиком выступает руководитель магазина. Ему нужно решить, какие изменения проверить в рекламных каналах во втором полугодии. Мы анализируем заказы с датой создания с 1 января по 30 июня 2025 года включительно. Данные фиксируют окончательные статусы и известные возвраты на момент учебной выгрузки. Отдельной даты возврата нет, поэтому весь возврат относится к месяцу исходного заказа. Это аналитика когорт заказов, а не календарный отчёт движения денег.

Такое уточнение сразу снимает потенциальный спор. Бухгалтер мог показать июльский возврат июньского заказа в

июле, а наша таблица уменьшит июньскую выручку. Оба отчёта могут быть корректны для разных целей. Ошибка возникнет, если сравнить их итоги без согласования периода признания.

Запишите задачу одним абзацем: «По учебным заказам первого полугодия определить, какие каналы и составляющие расходов связаны с изменением маржинального дохода после маркетинга; подготовить ежемесячную витрину и предложить ограниченную проверку улучшения». Затем перечислите, что не входит в работу: прогноз чистой прибыли, оценка пожизненной ценности клиента и причинный эффект рекламы по наблюдательным данным.

Единица наблюдения

Единица наблюдения — объект, который описывает одна строка. В orders это заказ. В customers — покупатель. В marketing — месяц и канал. В traffic — тоже месяц и канал, но значения представляют сеансы, а не людей. Пока вы не определили эту единицу для каждой таблицы, объединять их опасно.

Представьте заказ на три товара. Если таблица содержит три строки позиций, количество строк равно трём, а заказов только один. Расход на доставку 300 рублей относится ко всему заказу. После неосторожного объединения он может оказаться в каждой позиции, и сумма станет 900 рублей. Формулы работают, запрос выполняется, но ответ неверен из-за выбранной детализации.

Покупатели и сеансы также не взаимозаменяемы. Один человек может открыть сайт несколько раз и сделать несколько заказов. Показатель «завершённые заказы / сеансы» мы назовём отношением завершённых заказов к сеансам. Это полезная операционная метрика, но она не равна вероятности покупки уникальным посетителем. В другой модели такой коэффициент способен превышать 100%.

Договоримся о деньгах

В наших данных денежные поля выражены в целых рублях. У завершённого заказа `gross` — сумма товаров до скидки, `discount` — скидка, `refund` — возвращённая покупателю сумма. Чистая выручка для учебного анализа равна `gross - discount - refund`. Английское имя поля — `net_revenue`. Оно не означает универсальное бухгалтерское определение выручки.

Маржинальный доход до маркетинга равен `net_revenue - cost - shipping`. Поле `cost` обозначает себестоимость проданных товаров, `shipping` — переменные расходы на доставку. В учебной модели при полном возврате себестоимость и доставка не восстанавливаются: товар считается утраченным для дальнейшей продажи. Поэтому возвратный заказ может дать отрицательный вклад. В реальной компании нужно уточнить судьбу возвращённого товара и финансовые проводки.

Результат после маркетинга равен сумме маржинального дохода минус `spend`. Он не учитывает аренду, зарплаты и другие постоянные расходы. Название «чистая прибыль» здесь было бы неверным. Мы используем длинное, но точное название «маржинальный доход после маркетинга» и короткое имя `after_marketing` в коде.

У отменённых заказов все денежные поля равны нулю.

Они участвуют в числе созданных заказов, но не в числе завершённых. В реальном бизнесе отмена иногда приводит к расходам. Тогда правила придётся изменить. Не скрывайте отличия реальных данных от учебной схемы, чтобы получить знакомый результат.

Паспорт метрики

Для каждого важного показателя создайте короткий паспорт. Запишите название, деловой вопрос, формулу, единицу измерения, детализацию, период, фильтры и ответственного за определение. Например, доля возвратных заказов — число завершённых заказов с $\text{refund} > 0$, делённое на число завершённых заказов. Это доля заказов, а не доля возвращённых рублей и не доля возвращённых товарных единиц.

Отдельно укажите поведение при нулевом знаменателе. Если завершённых заказов нет, доля возвратов неопределена. Отображайте пустое значение или прочерк с пояснением. Ноль утверждал бы, что возвратов среди существующих продаж нет, хотя продаж не было. В витрине храните числитель и знаменатель: это позволит корректно пересчитать показатель при другом наборе фильтров.

Нельзя получать общую маржинальность простым средним процентов каналов. У условного канала А выручка 1 000 рублей и вклад 500, у Б — 9 000 и вклад 900. Среднее двух долей равно 30%, а общая доля — $1\,400 / 10\,000 = 14\%$. Разница возникает из-за весов. Для итоговой карточки делите сумму вкладов на сумму выручки.

Практическая работа

2. Согласуйте результат

Подготовьте одностраничный бриф проекта. Включите вопрос руководителя, период, определения трёх денежных показателей, единицу строки каждой таблицы и перечень исключений. Укажите, что магазин и все записи учебные. Добавьте критерии приёмки: итоги SQL и Python совпадают, все месяцы присутствуют, фильтр канала изменяет связанные карточки, ограничения видны пользователю.

Проверьте бриф на неоднозначность. Что произойдёт с отменой? К какому месяцу отнесён возврат? Можно ли сравнить результат с банковской выпиской? Почему расходы на маркетинг нельзя присоединить к каждому заказу и затем просуммировать? Ответы должны следовать из текста, а не зависеть от вашей памяти. Теперь можно принимать данные и проверять, соответствуют ли они договорённостям.

Глава 3. Получение данных и первая проверка

Что находится в комплекте

Разархивируйте материалы в отдельную папку. Файл `orders_raw.csv` содержит 4 827 строк: в него намеренно добавлены двенадцать точных повторов и неодинаковая запись названий каналов. `orders_clean.csv` содержит 4 815 заказов без этих дефектов и служит эталоном. Не начинайте упражнение с эталона, иначе пропустите очистку. Файлы `customers.csv` и `products.csv` — справочники. В `marketing.csv` и `traffic.csv` по 24 строки: шесть месяцев и четыре канала.

База `shop.sqlite` уже содержит чистые таблицы `orders`, `customers`, `products`, `marketing` и `traffic`. Скрипт `analyze.py` добавляет обогащённые заказы, витрину `mart_channel_month` и таблицу `monthly`. Исходный SQL-этап книги работает с `orders`; в нём дубли уже исключены. Это специально разделённые маршруты: в Excel и Python вы изучаете очистку, а в SQL сосредоточиваетесь на логике запросов. Если импортируете сырой CSV самостоятельно, сначала выполните те же проверки качества.

Скрипт `build_project.py` позволяет заново создать учебные

данные. Он использует фиксированное начальное состояние генератора случайных чисел. Не меняйте код и исходное состояние во время прохождения, иначе контрольные суммы станут другими. Скрипт не загружает данные из интернета и не требует учётных записей. Для проверки собственного кода достаточно приложенных файлов.

Словарь исходных полей

В `orders` поле `order_id` идентифицирует заказ, `order_date` хранит дату его создания в формате год-месяц-день. `customer_id` связывает заказ с покупателем, `product_id` — с товаром. `channel` принимает `organic`, `search`, `social` или `email`. `status` имеет значения `completed` и `cancelled`. `quantity` — количество единиц товара в заказе. Поля `gross`, `discount`, `refund`, `cost` и `shipping` хранят денежные суммы в рублях на весь заказ.

В `customers` одна строка соответствует одному `customer_id`; `city` обозначает город. В `products` находятся `product_id`, категория `category`, справочная цена `unit_price` и справочная себестоимость `unit_cost`. Для расчёта финансов используйте денежные суммы заказа, а не текущую цену справочника. В реальной системе справочная цена могла измениться после покупки. Историческую сумму нельзя восстанавливать по сегодняшнему прайс-листу без дополнительных условий.

В `marketing` сумма `spend` относится к комбинации `month` и `channel`. Значение `month` имеет вид 2025-01. В `traffic` поле `sessions` показывает число сеансов за ту же комбинацию. Расходы не разбиты по заказам, а сеансы не содержат идентификаторов пользователей. Поэтому мы не можем установить, какой конкретно сеанс привёл к заказу, и не можем расчи-

тать пользовательскую конверсию.

Словарь нужен не только для новичка. Представьте, что коллега заменил выгрузку и поле `cost` стало означать себестоимость одной единицы товара. Запрос по-прежнему выполнится, но перестанет учитывать количество. В словаре должно быть написано «на весь заказ». При изменении схемы эта формулировка поможет обнаружить несовместимость до публикации отчёта.

Принять файл — отдельная операция

До преобразования сохраните полученный оригинал. Зафиксируйте имя, дату получения, источник, период и число строк. В рабочую папку положите копию. Если через неделю источник пришлёт исправленную выгрузку, вы сможете понять, что изменилось. Нельзя одновременно изучать дефекты и незаметно исправлять единственный экземпляр данных.

CSV — текстовый формат, а не полноценная электронная таблица. В нём могут различаться разделители, кодировка, десятичный знак и формат дат. Открытие двойным щелчком доверяет автоматическому распознаванию редактора. При этом идентификатор с ведущими нулями иногда превращается в число, а неоднозначная дата интерпретируется в другой локали. Надёжнее запускать импорт и явно выбирать параметры.

Учебные CSV записаны в UTF-8 с маркером BOM и разделителем-запятой. В Excel используйте импорт из текста или CSV и проверьте предпросмотр. Число колонок должно совпадать со словарём. Не исправляйте неправильный разбор ручным перемещением ячеек. Вернитесь к параметрам импорта и устраните причину.

Пять проверок до анализа

Первая проверка — структура. Есть ли обязательные поля и не появились ли неожиданные колонки? Вторая — уникальность ключей. Третья — заполненность обязательных значений. Четвёртая — допустимость значений и диапазонов. Пятая — логические связи: возврат не должен превышать сумму после скидки, а заказ не может ссылаться на отсутствующего покупателя без отдельного правила обработки.

Не все необычные значения ошибочны. Отрицательная сумма могла быть корректирующей операцией. Пустой город мог означать, что его не собирали. Крупная покупка могла быть оптовой. Аналитик сначала определяет смысл, затем выбирает действие. Автоматически удалять всё, что отличается от среднего, значит менять исследуемую картину без основания.

Разделите дефекты на блокирующие и предупреждения. Повтор ключа с разными денежными суммами блокирует расчёт до выяснения версии записи. Несколько вариантов написания канала можно исправить однозначным правилом. Пустой город допускает публикацию общих итогов, но требует категории «Не указан» в географическом разрезе и предупреждения о полноте. Решение должно зависеть от влияния на вопрос пользователя.

Дубликат записи и повторное событие

Две полностью одинаковые строки в учебном CSV — технический повтор. Их можно оставить в одном экземпляре после фиксации правила. Но два заказа одного покупателя на один товар в один день не обязательно дубликаты. У каждого может быть собственный `order_id`. Удаление по имени, дате и сумме уничтожит настоящую покупку.

Если один `order_id` встречается дважды с разными статусами, перед вами может быть журнал изменений. Тогда нужна дата обновления и правило выбора последней версии. Без неё нельзя бездумно оставлять «первую» строку: порядок выгрузки не гарантирует хронологию. Запросите описание источника и сохраните конфликтные записи отдельно.

В учебных данных после удаления двенадцати точных повторов остаётся 4 815 уникальных `order_id`. Затем названия каналов приводятся к нижнему регистру, а пробелы по краям удаляются. Финансовые суммы от такой нормализации не меняются. Сравнение общей выручки до и после удаления дублей должно быть отражено в журнале качества, чтобы изменение не выглядело необъяснимым.

Практическая работа

3. Протокол приёма

Создайте таблицу проверок с колонками «объект», «правило», «результат», «действие». Для `orders_raw` запишите 4 827 исходных строк, двенадцать точных повторов, 4 815 строк после очистки и нормализацию каналов. Для справочников проверьте уникальность ключа. Для `marketing` и `traffic` проверьте уникальность пары месяц-канал и наличие всех 24 комбинаций.

Не ограничивайтесь отметкой «ошибок нет». Сохраните численный результат каждой проверки: сколько пустых ключей, сколько неизвестных каналов, сколько несовпавших ссылок. Если правило допускает исключение, укажите, почему. Такой протокол позволяет повторить проверку при новом обновлении и сравнить качество источника во времени.

Дополнительное задание: сделайте копию данных, замените `channel` в одной строке на `partner` и попробуйте провести её через проверку. Она должна остановить обработку или создать явное предупреждение. Если новый канал незаметно исчезает из отчёта, фильтрация скрыла проблему. Исправьте процесс прежде, чем переходить к визуализации.

Глава 4.

Excel

для проверки и первого анализа

Организуем рабочую книгу

Создайте книгу с листами «Описание», «Исходные данные», «Заказы», «Проверки» и «Сводка». На первом листе укажите период, единицу строки и определения показателей. Исходный лист используйте как копию импорта. Преобразования выполняйте на рабочем листе или в Power Query. Итоговые показатели не должны зависеть от незаписанных ручных исправлений.

Преобразуйте диапазон в таблицу Excel и присвойте ей имя Orders. Таблица расширяется при добавлении строк, а структурированные ссылки понятнее адресов вроде K2:K4816. Названия исходных колонок можно оставить английскими, чтобы они совпадали с CSV и кодом. Русские пояснения поместите в словарь. Не объединяйте ячейки внутри данных: объединение мешает фильтрации, сортировке и автоматической обработке.

Проверьте типы данных. `order_id` и `quantity` — целые числа, денежные поля — числа, `order_date` — дата. Формат «денежный» меняет внешний вид, но не превращает текст в число. Если сумма не считается или сортировка идёт странно, посмотрите содержимое строки формул и примените явное преобразование. Случайные пробелы и неразрывные пробелы тоже могут мешать распознаванию.

Формулы на уровне заказа

Добавьте столбец `net_revenue`. В таблице Excel формула одинаково применится ко всем строкам:

`=[@gross]-[@discount]-[@refund]`

Следующий столбец `contribution` показывает вклад до маркетинга:

`=[@net_revenue]-[@cost]-[@shipping]`

Чтобы выделить завершённый заказ, добавьте `completed_flag`:

`=IF([@status]="completed",1,0)`

В русской локали IF называется ЕСЛИ, а разделителем аргументов часто служит точка с запятой. Тогда последняя формула выглядит так:

`=ЕСЛИ([@status]="completed";1;0)`

Здесь и далее английские имена функций помогают сопоставлять примеры с документацией. Если редактор не принимает формулу, сначала проверьте имя функции и разделитель аргументов. Не меняйте смысл вычисления, пытаясь добиться отсутствия сообщения об ошибке.

Поле `month` можно получить как первое число месяца и

отображать форматом год-месяц. Это сохраняет тип даты и облегчает сортировку. Если нужен текстовый ключ для объединения с marketing, используйте YEAR и MONTH либо Power Query. При применении TEXT к дате помните, что коды формата зависят от локали. Не смешивайте текст «2025-01» и дату 01.01.2025 в одной колонке объединения.

Проверяем формулы на нескольких строках

Найдите завершённый заказ без возврата и пересчитайте его вручную. Если $\text{gross} = 2\,000$, $\text{discount} = 100$, $\text{refund} = 0$, $\text{cost} = 1\,100$, $\text{shipping} = 180$, то $\text{net_revenue} = 1\,900$, $\text{contribution} = 620$. Это отдельный условный пример. Он помогает проверить формулу, но не задаёт контрольную сумму всех данных.

Затем проверьте возвратный заказ с теми же исходными значениями и $\text{refund} = 1\,900$. Чистая выручка станет нулевой, а вклад — минус 1 280 рублей. Отрицательное значение соответствует нашему допущению об утраченной себестоимости и расходе доставки. Если в вашем результате появляется ноль, где-то было применено необоснованное ограничение отрицательных значений.

Проверьте отменённый заказ. Денежные поля в комплекте равны нулю, completed_flag тоже равен нулю. Количество созданных заказов при этом увеличивается на единицу. Контроль трёх разных ситуаций полезнее, чем просмотр сотни случайных строк с одинаковым поведением.

Сводная таблица без ловушек

После очистки двенадцати точных повторов и нормализации каналов создайте сводную по таблице Orders [10]. Поместите month в строки, channel — в столбцы, net_revenue и contribution — в значения. Проверьте, что для денежных полей выбрана сумма, а не количество. Если Excel предложил количество, часть значений могла остаться текстом. Вернитесь к исходному типу и исправьте его.

Создайте отдельную сводную для числа заказов и суммы completed_flag. Число строк после очистки должно быть 4 815, завершённых заказов — 4 428. Чистая выручка за полугодие равна 9 453 295 рублей, вклад до маркетинга — 1 970 070. Сравните эти суммы с expected.json. При несовпадении ищите первый этап расхождения: импорт, удаление дублей, фильтр статусов или формула.

Не добавляйте spend к каждой строке заказа. Сначала получите сводку с одной строкой на месяц и канал, затем присоедините маркетинг с той же детализацией. Например, январские 97 000 рублей по social должны попасть в итог января один раз. Если присоединить их к 175 заказам и сложить, получится 16 975 000 рублей. Это ошибка модели, которую нельзя исправить форматированием.

В итоговой сводке вычисляйте маржинальность делением общей суммы вклада на общую сумму выручки. Для про-

стой небольшой книги можно вынести формулу за пределы сводной и явно сослаться на итоговые значения. В модели данных используйте меру с делением сумм. В обоих случаях проверьте результат на примере с разными весами из предыдущей главы.

Поиск по справочнику

Чтобы добавить категорию товара, нужно найти `product_id` в справочнике. В современных версиях Excel доступна `XLOOKUP`, в русской локали — `ПРОСМОТРХ`; её наличие зависит от версии. Для старой версии можно использовать точный поиск через `INDEX` и `MATCH`. Пример ниже предполагает таблицу `Products`:

```
=INDEX(Products[category],  
MATCH([@product_id],Products[product_id],0))
```

Ноль в `MATCH` означает точное совпадение. Приблизжённый поиск без специальных условий может вернуть соседний товар. Если справочник содержит два одинаковых `product_id`, поиск скроет проблему, выбрав одно совпадение. Перед поиском всегда проверяйте уникальность ключа справочника.

Не оборачивайте всё в `IFERROR` с пустой строкой. Так вы избавитесь от видимых ошибок, но потеряете сигнал о неизвестном товаре. Сначала создайте колонку «Ключ найден» и посчитайте несовпадения. Затем решите, какие из них блокируют расчёт. Пустое значение в отчёте должно быть результатом осмысленного правила, а не способом скрыть дефект.

Практическая работа

4. Первая записка

Постройте помесечную сводку и линейный график `net_revenue`. Рядом покажите `contribution`. На одном графике эти показатели могут сильно различаться по масштабу; безопаснее использовать два согласованных графика с одинаковой осью времени. Подпишите единицы измерения и не добавляйте декоративную трёхмерность.

Напишите три предложения. Первое описывает наблюдение, второе указывает, что ещё нужно проверить, третье формулирует следующий шаг. Например: «Во втором квартале вклад до маркетинга снизился при более высокой выручке, чем в начале периода. Пока неизвестно, связан ли результат с отдельным каналом или с общим изменением расходов. Далее сравним структуру каналов и составляющие вклада». Не объявляйте рекламу причиной до соответствующей проверки.

Сохраните книгу и убедитесь, что другой человек сможет определить источник каждой итоговой цифры. Если важная сумма введена вручную, подпишите её как ручной параметр и объясните происхождение. Рабочая книга должна помогать искать ошибку, а не только показывать окончательный ответ.

Глава 5.

Power

Query

и повторяемая очистка

Записываем последовательность преобразований

Когда выгрузка приходит регулярно, ручная очистка становится источником расхождений. Один раз вы удалили пробелы, другой раз забыли, третий раз отсортировали только часть диапазона. Power Query сохраняет шаги обработки и позволяет применять их к следующему файлу. Смысл инструмента — в воспроизводимости, а не в количестве освоенных кнопок.

Импортируйте `orders_raw.csv` через меню данных. В редакторе проверьте разделитель и кодировку. Сначала задайте типы, затем удалите точные повторы всех исходных колонок, приведите `channel` к нижнему регистру и удалите пробелы по краям. Последовательность и названия шагов сохраните так, чтобы через месяц было понятно, зачем нужен каждый из них.

Удаление дублей только по `order_id` было бы допустимо после проверки отсутствия конфликтующих версий. В нашем упражнении безопаснее удалить полные повторы, а потом отдельно проверить уникальность `order_id`. Если уникальность не достигнута, процесс должен сообщить о конфликте. Нельзя молча считать его очередным техническим повтором.

Типы и локаль

Локаль задаёт правила интерпретации текста. Дата 03/04/2025 в разных соглашениях может означать март или апрель. Для учебного ISO-формата год-месяц-день неоднозначность меньше, но реальный источник может прислать другой вариант. Используйте преобразование типа с нужной локалью и проверьте минимальную и максимальную даты после него.

Для денежных сумм исходного проекта подходит целое число. В реальной выгрузке с копейками выбирайте подходящий точный тип и согласуйте округление. Если данные представлены текстом «1 250,50», нужно убрать разделитель тысяч и распознать десятичную запятую. Простая замена всех запятых точками до разбора CSV может разрушить разделение колонок.

Ошибки преобразования не удаляйте без просмотра. Создайте отдельный запрос с ошибочными строками и посчитайте их. Если это обязательная дата, такие строки могут менять период отчёта. Если это необязательное описание, общие суммы могут сохраниться. Действие выбирается по смыслу поля и объёму проблемы.

Объединение и добавление

Добавление запросов складывает строки одинаковой структуры друг под другом. Так можно объединить январскую и февральскую выгрузки заказов. Объединение запросов сопоставляет строки по ключу и добавляет колонки. Так к заказу присоединяют город из `customers`. Эти операции решают разные задачи, хотя обе увеличивают содержимое результата.

Перед объединением заказов с `customers` запишите ожидаемое отношение: много заказов к одному покупателю. После операции число строк заказов не должно измениться. Затем посчитайте ненайденные ключи. В учебном комплекте их нет. Если строки выросли, проверьте дубли в справочнике; если сократились, возможно, выбран внутренний тип объединения и часть заказов потеряна.

Для расходов сначала сгруппируйте заказы по `month` и `channel`. Суммируйте `net_revenue`, `contribution` и `completed_flag`, посчитайте заказы. Затем объедините агрегат с `marketing` по двум полям. Порядок выбора полей должен соответствовать друг другу. Совпадение только по `channel` размножит каждую месячную строку на шесть месяцев расходов.

Обновление и проверка результата

Сохраните путь к файлу так, чтобы его можно было изменить в одном месте. При передаче книги коллеге локальный путь вашего компьютера может оказаться недоступным. В инструкции укажите, где выбрать новый источник. Не храните пароли в видимых ячейках и не включайте их в опубликованную книгу.

После обновления повторите контроль числа строк, периода и денежных итогов. Факт успешной загрузки означает только отсутствие технической ошибки выполнения. Источник мог прислать пустой файл, другой месяц или другую валюту. Поэтому у отчёта должны быть содержательные проверки, а не только зелёный индикатор обновления.

Создайте лист «Контроль обновления». Покажите время обработки, максимальную дату заказа, исходное и итоговое число строк, число конфликтов ключа и итог выручки. При использовании реальных поступающих данных дополните его датой формирования источника. Дата запуска Excel и свежесть данных — разные вещи.

Практическая работа 5.

Испытайте обновление

Сделайте копию сырого CSV и добавьте ещё один точный повтор. Обновите запрос. Итоговое число уникальных заказов должно остаться 4 815, а число удалённых повторов — вырасти с двенадцати до тринадцати. После проверки верните исходный файл, чтобы дальнейшие контрольные результаты относились к комплекту.

Второе испытание: измените название обязательной колонки в копии источника. Обработка должна завершиться заметной ошибкой или проверкой схемы. Если она выдаёт старый результат без предупреждения, пользователь может принять его за свежий. В инструкции опишите, где видно неудачное обновление и как вернуть рабочий источник.

Итог этапа — не только сводка магазина. У вас появился процесс, который можно повторить и объяснить. Дальше тот же принцип перенесём в SQL: запрос должен ясно задавать исходные строки, условия и уровень детализации результата.

Глава 6.

SQL

от выборки к проверяемой сводке

Подключаем учебную базу

Откройте shop.sqlite в редакторе, который умеет работать с SQLite. Можно использовать любой доступный графический клиент с таким подключением. Укажите файл существующей базы, а не создание новой пустой. Найдите таблицы orders, customers, products, marketing и traffic. Перед выполнением запроса проверьте имя подключения: одинаковые названия таблиц могут находиться в разных файлах.

Если графического клиента нет, запросы можно выполнять через Python, где модуль sqlite3 входит в стандартную библиотеку. В этой главе достаточно знать способ запуска; устройство Python подробно разберём позже. Создайте рядом с базой файл run_sql.py:

```
from pathlib import Path
import sqlite3
```

```
root = Path(__file__).resolve().parent
database = root / 'shop.sqlite'
if not database.exists():
    raise FileNotFoundError(database)
with sqlite3.connect(database) as con:
    rows = con.execute(
        'SELECT COUNT(*) FROM orders'
    ).fetchall()
```

```
print(rows)
```

Результат содержит число 4815. Скобки вокруг него — форма представления списка строк, а не часть данных. Если получено сообщение по `such table`, проверьте путь: SQLite может создать новый файл при подключении к несуществующему пути. Полезно сначала убедиться, что файл существует, и только потом открывать его.

SQL описывает, какой результат вы хотите получить. В `SELECT` перечисляются выходные поля, в `FROM` — источник, в `WHERE` — условия для строк. Порядок строк без `ORDER BY` не гарантирован. Даже если первые пять запусков показывают одинаковую последовательность, рассчитывать на неё нельзя [2].

Выбираем нужные поля

Начните с запроса, который показывает несколько заказов и не меняет данные:

```
SELECT
order_id,
order_date,
channel,
status,
gross,
discount,
refund
FROM orders
ORDER BY order_id
LIMIT 10;
```

Каждая запятая разделяет выражения. Последнее поле перед FROM пишется без запятой. Точка с запятой завершает запрос. Ключевые слова можно писать в разном регистре; единый стиль облегчает чтение. В готовых запросах перечисляйте поля явно: SELECT * полезен для первичного осмотра, но делает результат зависимым от будущих изменений схемы.

Добавим рассчитанный показатель и дадим ему имя через AS:

```
SELECT  
order_id,  
gross - discount - refund AS net_revenue,  
gross - discount - refund  
- cost - shipping AS contribution  
FROM orders  
ORDER BY order_id  
LIMIT 10;
```

Псевдоним делает вывод понятным и позволяет использовать результат в следующем уровне запроса. Не рассчитывайте, что имя `net_revenue` будет доступно в соседнем выражении того же `SELECT` во всех системах. Для переносимого кода повторите выражение или вынесите расчёт в подзапрос.

Фильтруем даты и статусы

Получим завершённые заказы первого квартала:

```
SELECT
order_id,
order_date,
channel,
gross - discount - refund AS net_revenue
FROM orders
WHERE status = 'completed'
AND order_date >= '2025-01-01'
AND order_date < '2025-04-01'
ORDER BY order_date, order_id;
```

Полуоткрытый интервал включает начало и исключает следующую границу. Он удобен и для дат, и для временных меток. Условие «до 31 марта 00:00» при наличии времени исключило бы почти весь последний день. В учебной базе даты хранятся ISO-текстом, поэтому их лексикографический порядок совпадает с календарным. Для произвольного текста это утверждение неверно.

Если нужны два канала, используйте IN. Если объединяете AND и OR, ставьте скобки вокруг смысловой группы. Иначе условие может допустить любой social-заказ вне периода:

```
SELECT order_id, channel, order_date  
FROM orders  
WHERE channel IN ('search', 'social')  
AND order_date >= '2025-04-01'  
AND order_date < '2025-07-01'  
ORDER BY order_id;
```

Текстовые значения заключаются в одинарные кавычки. Равенство в SQL записывается одним знаком =. Для отсутствующих значений применяют IS NULL, а не = NULL. NULL обозначает неизвестное или отсутствующее значение и участвует в логике иначе, чем обычное число или пустая строка.

Считаем итоги

Проверим число строк, завершённые заказы и денежные суммы:

```
SELECT
COUNT(*) AS orders,
SUM(CASE WHEN status = 'completed'
THEN 1 ELSE 0 END) AS completed,
SUM(gross - discount - refund) AS net_revenue,
SUM(gross - discount - refund
- cost - shipping) AS contribution
FROM orders;
```

Вы получите 4 815, 4 428, 9 453 295 и 1 970 070 соответственно. CASE проверяет условие для каждой строки. SUM складывает полученные значения. COUNT(*) считает строки, а COUNT(column) — строки, где указанная колонка не равна NULL. В наших обязательных ключах пропусков нет, но различие станет важным при внешнем объединении.

SUM по пустому набору в SQLite возвращает NULL. Если по смыслу нужна сумма ноль, используйте COALESCE(SUM(...), 0), предварительно убедившись, что отсутствие строк действительно означает отсутствие событий, а не сбой загрузки. Замена неизвестного значения нулём не должна скрывать проблему полноты данных.

Теперь сгруппируем результат по каналам:

```
SELECT
channel,
COUNT(*) AS orders,
SUM(gross - discount - refund) AS net_revenue,
SUM(gross - discount - refund
- cost - shipping) AS contribution
FROM orders
GROUP BY channel
ORDER BY net_revenue DESC, channel;
```

После GROUP BY одна строка результата описывает канал. Все выбранные поля должны либо входить в группировку, либо вычисляться агрегатной функцией. SQLite допускает некоторые более свободные конструкции, но использовать их для учебного отчёта не стоит: произвольная дата из группы рядом с суммой всех заказов вводит читателя в заблуждение.

WHERE

и

HAVING

WHERE отбирает исходные строки, HAVING — группы после агрегации. Чтобы найти каналы с выручкой выше заданной суммы, условие по сумме нужно поставить в HAVING:

SELECT

```
channel,  
SUM(gross - discount - refund) AS net_revenue  
FROM orders  
WHERE status = 'completed'  
GROUP BY channel  
HAVING SUM(gross - discount - refund) > 2000000  
ORDER BY net_revenue DESC;
```

Запрос не означает «заказы дороже двух миллионов». Он сначала выбирает завершённые заказы, складывает их внутри канала и затем оставляет группы, чей итог превышает порог. Разница между фильтром строк и фильтром групп — частая тема собеседований, потому что она отражает понимание уровня расчёта.

Практическая работа 6.

Напишите первые запросы

Получите все отменённые заказы за июнь, затем число таких заказов по каждому каналу. Отдельно найдите десять завершённых заказов с наибольшей чистой выручкой. При равенстве суммы используйте `order_id` как дополнительный критерий сортировки, чтобы результат был воспроизводимым.

Затем напишите запрос с выручкой по месяцам. Для SQLite возьмите `substr(order_date, 1, 7)`. Не добавляйте в `SELECT` полный `order_date` рядом с месячной суммой. Сверьте январь: 1 435 795 рублей. Если получен другой результат, проверьте таблицу, выражение выручки и фильтр периода. Отменённые заказы в нашей модели имеют нулевые суммы, поэтому их исключение не меняет выручку, но меняет число созданных заказов.

Глава 7. Объединения и модель данных

Как соединяются таблицы

JOIN сопоставляет строки по условию. Внутреннее объединение INNER JOIN оставляет найденные пары, левое LEFT JOIN сохраняет все строки слева и добавляет совпадения справа. Если соответствия нет, поля правой таблицы становятся NULL. Если соответствий несколько, левая строка повторяется для каждого. Именно последнее свойство часто приводит к завышенным суммам.

Соединим заказ с городом покупателя:

```
SELECT  
o.order_id,  
o.customer_id,  
c.city,  
o.gross - o.discount - o.refund AS net_revenue  
FROM orders AS o  
LEFT JOIN customers AS c  
ON o.customer_id = c.customer_id  
ORDER BY o.order_id  
LIMIT 20;
```

Буквы `o` и `c` — короткие имена таблиц. Они помогают различать одноимённые колонки. Перед объединением мы ожидаем один `customer_id` в справочнике. После объединения число строк должно остаться 4 815. Это ожидание нужно проверять, а не держать в уме.

```
SELECT customer_id, COUNT(*) AS copies
FROM customers
GROUP BY customer_id
HAVING COUNT(*) > 1;
```

Правильный результат — пустая выборка. Чтобы найти отсутствующие соответствия, используйте ключ правой таблицы:

```
SELECT o.order_id, o.customer_id
FROM orders AS o
LEFT JOIN customers AS c
ON o.customer_id = c.customer_id
WHERE c.customer_id IS NULL;
```

В комплекте таких строк нет. В реальной задаче выясните, означает ли отсутствие задержку обновления справочника, ошибочный идентификатор или допустимого гостевого покупателя. Выбор `INNER JOIN` без проверки просто убрал бы эти заказы из выручки.

Почему

LEFT JOIN

может стать внутренним

Предположим, вам нужны заказы и, если есть, город Москва. Условие `c.city = 'Москва'` в `WHERE` удалит все строки без совпадения справа, потому что сравнение с `NULL` не даёт истинного результата. Если задача — оставить все заказы и присоединить только московское соответствие, условие нужно поместить в `ON`. Если задача — посчитать именно московские заказы, `WHERE` подходит. Размещение условия определяется смыслом, а не правилом «всегда пишите здесь».

Рассмотрите запрос перед выполнением. Что означает строка после объединения? Что должно произойти с ненайдённым ключом? Сколько совпадений справа допустимо? Эти три вопроса полезнее попытки запомнить все виды `JOIN` по картинке.

Сначала агрегируем, затем соединяем расходы

Нам нужна одна строка на месяц и канал. Создадим общий табличный блок через WITH. Такой блок называется CTE; он задаёт именованный промежуточный результат внутри запроса:

```
WITH sales AS (  
  SELECT  
    substr(order_date, 1, 7) AS month,  
    channel,  
    COUNT(*) AS orders,  
    SUM(CASE WHEN status = 'completed'  
      THEN 1 ELSE 0 END) AS completed,  
    SUM(gross - discount - refund) AS net_revenue,  
    SUM(gross - discount - refund  
      - cost - shipping) AS contribution  
  FROM orders  
  GROUP BY substr(order_date, 1, 7), channel  
)  
SELECT  
  s.month,  
  s.channel,  
  s.orders,  
  s.completed,
```

```
s.net_revenue,  
s.contribution,  
m.spend,  
s.contribution - m.spend AS after_marketing  
FROM sales AS s  
LEFT JOIN marketing AS m  
ON s.month = m.month  
AND s.channel = m.channel  
ORDER BY s.month, s.channel;
```

В результате должно быть 24 строки. Сумма spend — 1 614 000 рублей, сумма after_marketing — 356 070. Если в marketing есть канал без единого заказа, такой запрос его не покажет, потому что слева находятся продажи. Для нашего полного учебного набора это допустимо. В рабочем процессе создайте каркас всех месяцев и каналов либо объединение ключей из продаж и расходов, а затем присоединяйте оба источника.

Не заменяйте отсутствующий spend нулём до проверки полноты. Отсутствующая строка могла означать, что рекламный кабинет не выгрузился. Для organic ноль записан явно и является значением. Пустое значение после JOIN говорит о несовпадении ключа или отсутствии записи. Эти ситуации нельзя смешивать.

Заказы и позиции

Рассмотрим отдельный маленький пример, которого нет в основной таблице. Есть два заказа: у первого доставка 300 рублей и две позиции, у второго доставка 200 и одна позиция. Общая доставка равна 500. Если объединить заказы с позициями, получатся три строки, и сумма доставки станет 800. Выручку позиций суммировать можно, доставку заказа — уже нет.

Одно решение — сначала агрегировать позиции до заказа, затем соединить с таблицей заказов. Другое — распределить доставку между позициями по согласованному правилу. Например, пропорционально выручке позиции. Тогда сумма распределённых частей должна точно восстанавливать стоимость заказа, а остаток округления нужно отнести по явному правилу.

Нельзя исправлять такую ошибку выражением `SUM(DISTINCT shipping)`. Если у двух настоящих заказов одинаковая доставка 300 рублей, `DISTINCT` оставит одно значение и занижит сумму. Уникальность суммы не равна уникальности заказа. Сначала восстановите нужную детализацию, затем складывайте.

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.