Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

**ОТЧЕТ по лабораторной работе №**6

по дисциплине «Базы данных»

**Выполнил:**

студент группы 5130902/20201 \_\_\_\_\_\_\_\_\_\_\_\_\_ А. И. Сафонов

подпись

**Проверил:**

Кандидат тех. наук, доцент \_\_\_\_\_\_\_\_\_\_\_\_ С. А. Нестеров

подпись

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г.

Санкт-Петербург, 2025

Оглавление

[Яндекс практикум 15](#_Toc190869791)

[Авиаперевозки 25](#_Toc190869792)

Лабораторная 8

Задание. Используя базу из файлов к лабораторной работе, напишите запрос, добавляющий новый статус «устарела» в таблицу BookStatus.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки. Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Задание. Напишите запрос, добавляющий в таблицу BookStatus1 названия статусов книг из таблицы BookStatus.

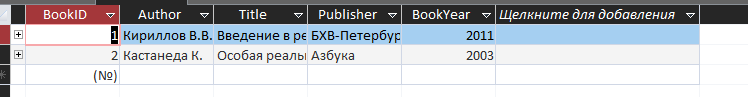
Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

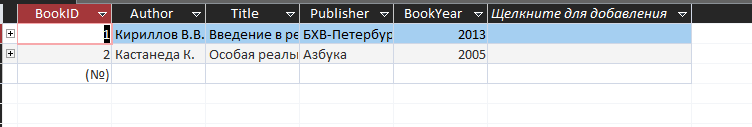
Контент, сгенерированный ИИ, может содержать ошибки. Изображение выглядит как текст, снимок экрана, Шрифт, число

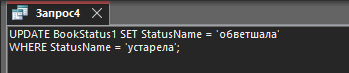
Контент, сгенерированный ИИ, может содержать ошибки.

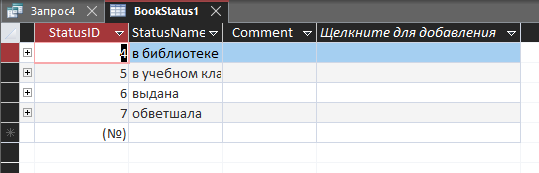
Задание. Напишите два запроса, первый из которых увеличивает на 2 год издания всех книг в таблице Book1, а второй изменяет в таблице BookStatus1 название статуса c «устарела» на «обветшала»





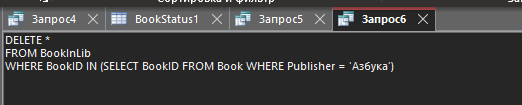


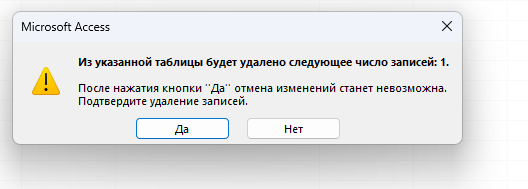




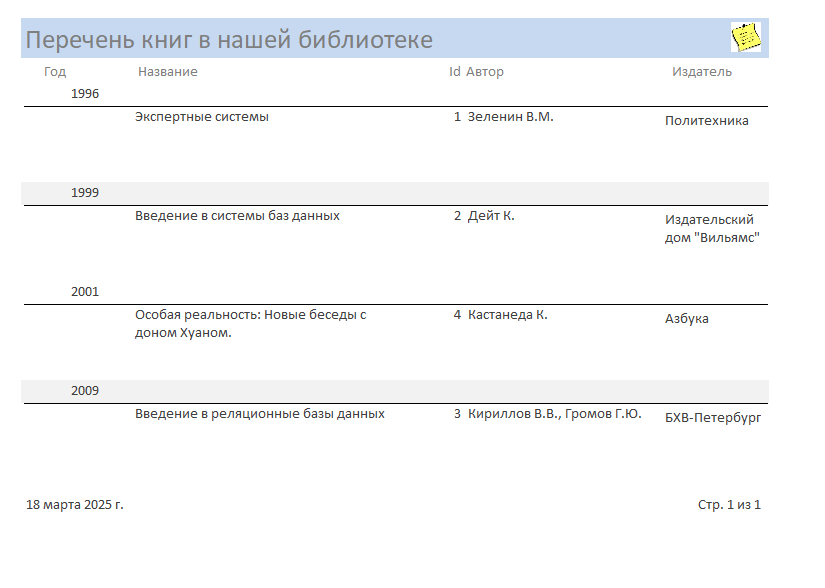
Задание. Напишите два запроса, первый из которых удаляет все записи из таблицы BookStatus1, а второй удаляет из таблицы BookInLib записи об экземплярах книг, изданных издательством «Азбука» (подсказка: в секции WHERE можно использовать условие IN и подзапрос).



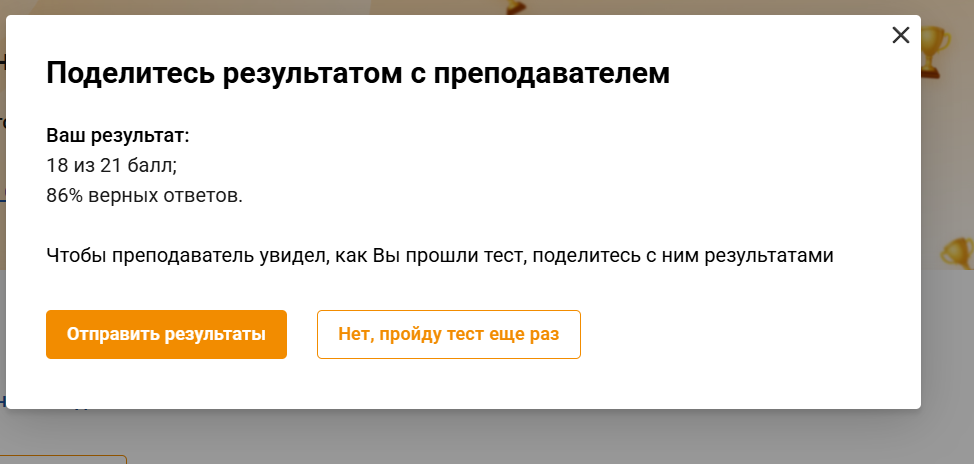


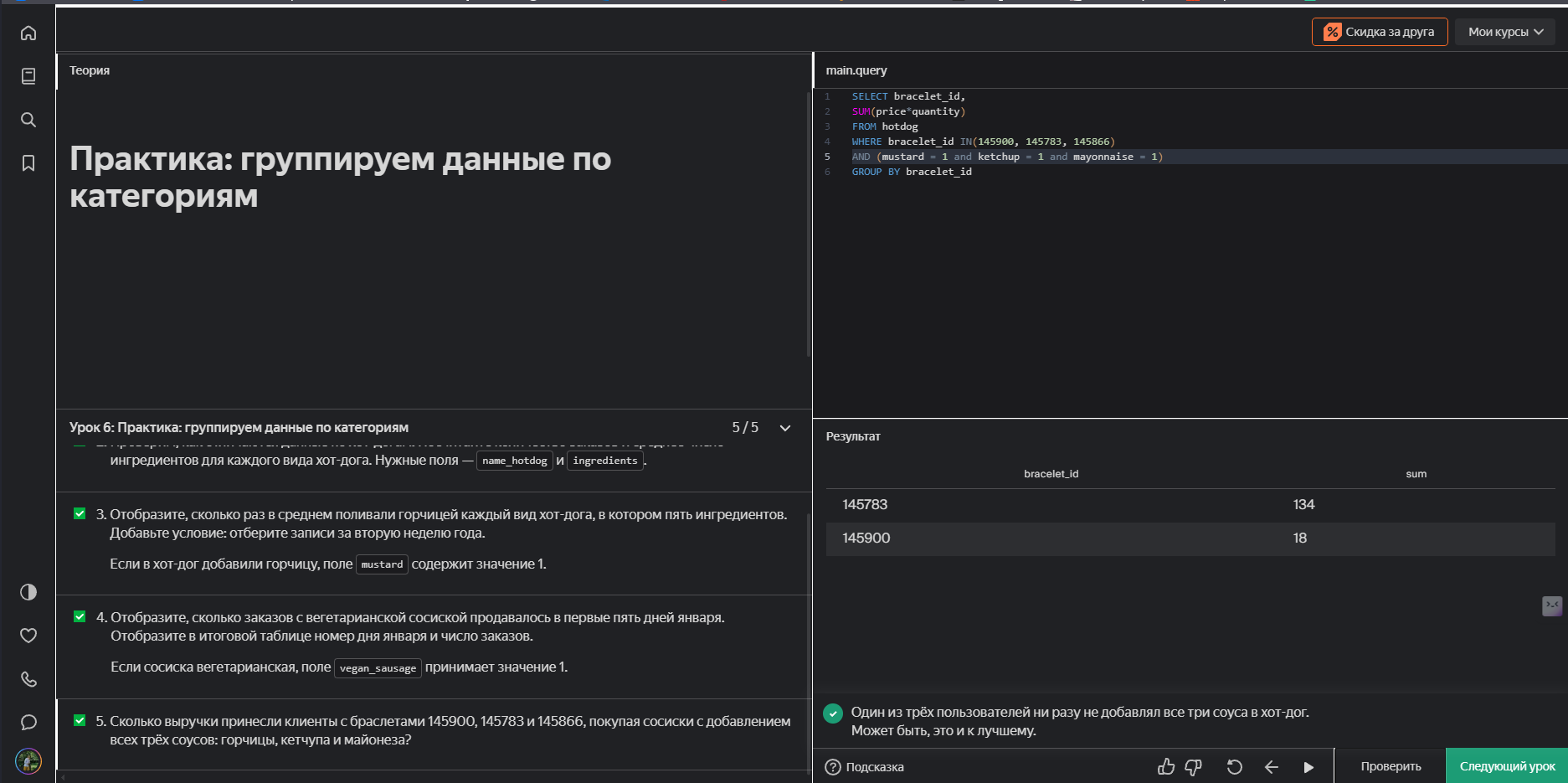
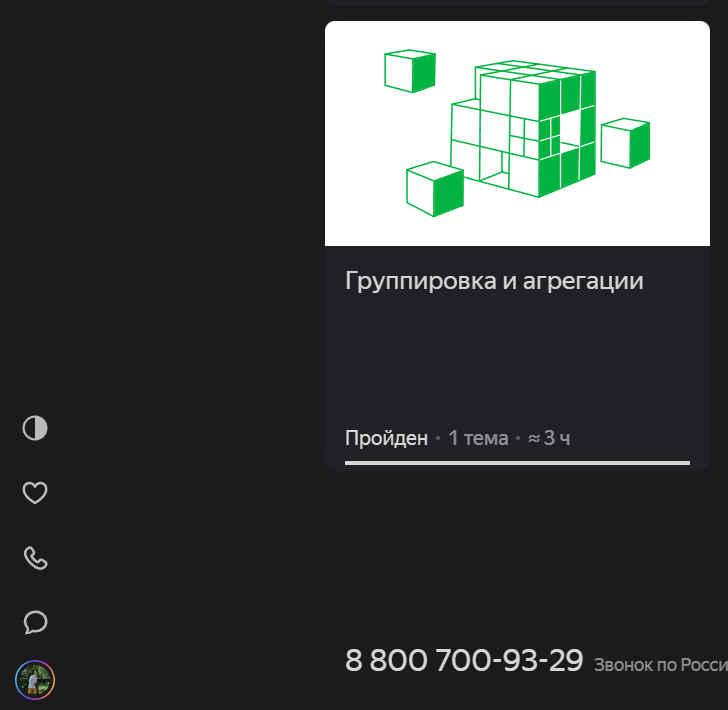


Задание. Перейдите на вкладку Создание и с помощью Мастера создайте отчет, выводящий данные о книгах из таблицы Book. Данные должны быть сгруппированы по году выхода книги, а в рамках группы отсортированы по названию. Скорее всего, в построенном в Мастере отчете какие-то данные не будут помещаться в отведенные им рамки. Поэтому в конструкторе разместите элементы в области данных так, чтобы все надписи выводились полностью, а названия столбцов замените на русские. В примечании отчета выведите надпись «Перечень книг в нашей библиотеке». В заголовок отчета добавьте какой-нибудь рисунок, например, из файлов к лабораторной. Сделайте так, что после каждой группы книг, определяемой одним годом издания, шла черта, отделяющая данную группу от других.



Urait



Яндекс Практикум  

НОРМАЛИЗАЦИЯ ДО НФБК

Требуется, основываясь на описании предметной области, спроектировать базу данных, все отношения в которой нормализованы до НФБК.

Предметная область – учет накопителей на жестких дисках (HDD), используемых в организации. О дисках мы знаем следующее:

- у жесткого диска есть производитель;

- у производителя есть web-сайт (считаем, что только один), где можно почерпнуть много важной для нас информации;

- для жесткого диска всегда определена его модель;

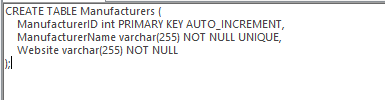
- модель определяет объем диска (в гигабайтах), скорость вращения шпинделя, тип используемого интерфейса;

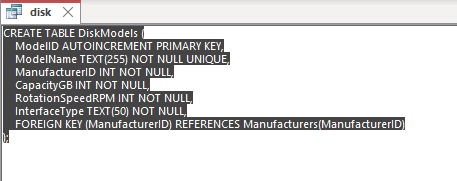
- по названию модели можно определить производителя;

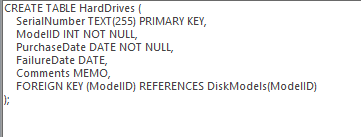
- у конкретного экземпляра жесткого диска есть серийный номер, мы знаем его модель, дату приобретения, дату выхода из строя (если диск вышел из строя; возможность ремонта и восстановления не рассматриваем); может возникнуть необходимость внести текстовые комментарии по поводу его работы. При проектировании надо учитывать, что:

- для любой модели обязательно должен быть указан производитель и объем; - для любого диска должна быть указана модель;

- фирмы-производители и модели дисков именуются уникальным образом; серийные номера дисков также уникальны.







Роман Симаков теория нормальных форм

Нормальная форма - свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Процесс преобразования отношений базы данных к виду, отвечающему нормальным формам, называется нормализацией. Нормализация предназначена для приведения структуры БД к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение физического объёма базы данных. Конечной целью нормализации является уменьшение потенциальной противоречивости, хранимой в базе данных информации. Общее назначение процесса нормализации заключается в следующем:

* Исключение некоторых типов избыточности.
* Устранение некоторых аномалий обновления.

Разработка проекта базы данных, который является достаточно «качественным» представлением реального мира, интуитивно понятен и может служить хорошей основой для последующего расширения.

Устранение избыточности производится, как правило, за счёт декомпозиции отношений таким образом, чтобы в каждом отношении хранились только первичные факты.

Теория нормализации является очень ценным достижением реляционной теории и практики, поскольку она даёт научно строгие и обоснованные критерии качества проекта БД и формальные методы для усовершенствования этого качества. Этим теория нормализации резко выделяется на фоне чисто эмпирических подходов к проектированию, которые предлагаются в других моделях данных. Более того, можно утверждать, что во всей сфере информационных технологий практически отсутствуют методы оценки и улучшения проектных решений, сопоставимые с теорией нормализации реляционных баз данных по уровню формальной строгости.

Всего выделяют 6 нормальных форм, но мы рассмотрим только первые 3.

**Первая нормальная форма (1НФ)**

Отношение находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении этой переменной каждый кортеж отношения содержит только одно значение для каждого из атрибутов.

В реляционной модели отношение всегда находится в первой нормальной форме по определению понятия отношение.

Что же касается различных таблиц, то они могут не быть правильными представлениями отношений и, соответственно, могут не находиться в 1НФ. В соответствии с определением Кристофера Дейта для такого случая таблица нормализована (т.е. находится в первой нормальной форме) тогда и только тогда, когда она является прямым и верным представлением некоторого отношения. Конкретнее, рассматриваемая таблица должна удовлетворять следующим пяти условиям:

* Нет упорядочивания строк сверху вниз (другими словами, порядок строк не несет в себе никакой информации).
* Нет упорядочивания столбцов слева направо (другими словами, порядок столбцов не несет в себе никакой информации).
* Нет повторяющихся строк.
* Каждое пересечение строки и столбца содержит ровно одно значение из соответствующего домена (и больше ничего).
* Все столбцы являются обычными.

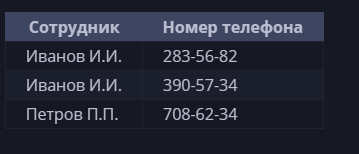
«Обычность» всех столбцов таблицы означает, что в таблице нет «скрытых» компонентов, которые могут быть доступны только в вызове некоторого специального оператора взамен ссылок на имена регулярных столбцов, или которые приводят к побочным эффектам для строк или таблиц при вызове стандартных операторов. Таким образом, например, строки не имеют идентификаторов кроме обычных значений потенциальных ключей (без скрытых «идентификаторов строк» или «идентификаторов объектов»). Они также не имеют скрытых временных меток.

**Пример**

Исходная ненормализованная (то есть не являющаяся правильным представлением некоторого отношения) таблица:



Таблица, приведённая к 1НФ, являющаяся правильным представлением некоторого отношения:



Атомарность

Многие авторы дополняют определение первой нормальной формы требованием атомарности (неделимости) значений. Однако концепция «атомарности» является слишком неясной. Например, многие типы данных (строки, даты, числа с фиксированной точкой и т. д.) при необходимости легко могут быть декомпозированы на составляющие элементы с помощью стандартных операций, предоставляемых СУБД. К. Дейт заключает, что «понятие атомарности не имеет абсолютно никакого смысла».

Исторически концепция «атомарности» берёт начало от «простых доменов» (англ. simple domains), предложенных автором реляционной модели данных Э. Ф. Коддом. Цель «нормальной формы», которую предложил Кодд в статье «Реляционная модель данных для больших совместно используемых банков данных», не была связана с каким-либо теоретическим аспектом, например, с борьбой с аномалиями или избыточностью. Кодд предложил использовать «простые домены» только для облегчения будущей программной реализации, а именно:

* Для облегчения хранения отношений в виде двумерных массивов. Отношение, все домены которого являются простыми, может быть представлено при хранении двухмерным массивом с однородными столбцами.
* Для облегчения передачи данных в гетерогенных системах. Простота представления отношений массивами, осуществимая в случае приведения всех отношений в нормальную форму, предоставляет преимущества не только при хранении, но также при передаче больших объёмов данных между системами, использующими во многом отличные представления данных.

**Вторая нормальная форма (2НФ)**

Отношение находится во второй нормальной форме тогда и только тогда, когда оно находится в первой нормальной форме и каждый неключевой атрибут функционально полно зависит от его потенциального ключа.

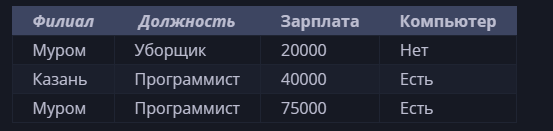
Полная функциональная зависимость означает, что каждый неключевой атрибут зависит от всего потенциального ключа, а не от его части.

Если потенциальный ключ является простым, то есть состоит из единственного атрибута, то любая функциональная зависимость от него является полной. Если потенциальный ключ является составным, то, согласно определению второй нормальной формы, в отношении не должно быть неключевых атрибутов, зависящих от части составного потенциального ключа.

Вторая нормальная форма по определению запрещает наличие неключевых атрибутов, которые вообще не зависят от потенциального ключа. Таким образом, вторая нормальная форма в том числе запрещает создавать отношения как несвязанные (случайные) наборы атрибутов.

**Пример**

Пусть в следующем отношении первичный ключ образует пара атрибутов {Филиал, Должность}:



Допустим, что зарплата зависит от филиала и должности, а наличие компьютера зависит только от должности.

Существует функциональная зависимость Должность → Наличие компьютера, в которой левая часть (детерминант) является лишь частью первичного ключа, что нарушает условие второй нормальной формы.

Для приведения к 2NF исходное отношение следует декомпозировать на два отношения:

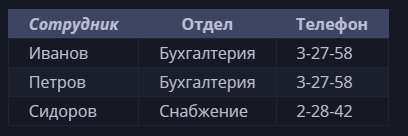


**Третья нормальная форма (3НФ)**

Отношение находится в третьей нормальной форме тогда и только тогда, когда оно находится во второй нормальной форме и ни один неключевой атрибут не зависит от другого неключевого атрибута.

**Пример**

Рассмотрим в качестве примера отношение:



Каждый сотрудник относится исключительно к одному отделу; каждый отдел имеет единственный телефон. Атрибут Сотрудник является первичным ключом. Личных телефонов у сотрудников нет, и телефон сотрудника зависит исключительно от отдела.

В примере:

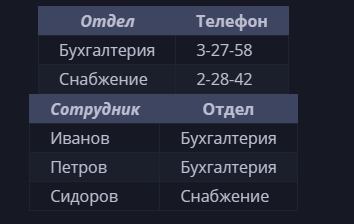
* Отдел зависит от Сотрудника.
* Телефон зависит от Отдела.
* Телефон зависит от Сотрудника (транзитивно).

Отношение находится во второй нормальной форме, поскольку каждый атрибут функционально полно зависит от потенциального ключа Сотрудник.

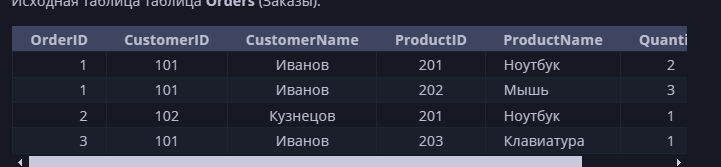
Телефон зависит от неключевого атрибута Отдел, следовательно, отношение не находится в третьей нормальной форме.

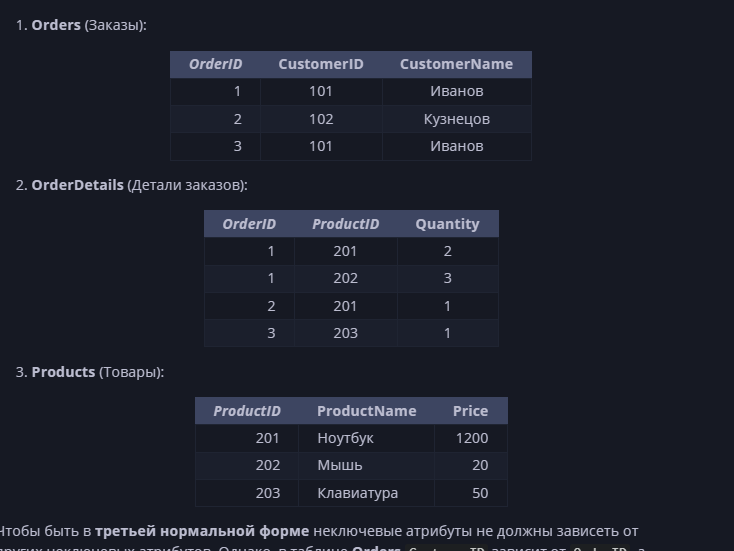
Кроме этого, очевидно, что присутствует избыточность, т.к. телефоны повторяются для всех сотрудников одного отдела.

Для приведения отношения к третьей нормальной форме, необходимо разделить его на два:



Упражнение





Чтобы перейти к 3 форме:

