**Лабораторная работа № 2. Проектирование баз данных. Нормализация**

**КУРНОСЕНКО СОФЬЯ**

→ ***Теория (методичка)***

При проектировании реляционной базы данных необходимо исследовать предметную область с целью определения объектов, нормализовать данные и установить связи между ними. **Нормализация** **данных** − это процесс, в результате выполнения которого таблицы базы данных проверяются на наличие зависимостей между столбцами и, если необходимо, исходная таблица разделяется на несколько таблиц.

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1.Изучить способ приведения информации к *первой нормальной форме*, проанализировав пример в правой части. | Пусть исходная информация о продажах имеет следующие поля:  **Наименование\_товара, Цена, Количество\_на\_складе, Наименование\_заказанного\_ товара, Цена\_заказанного\_товара, Количество\_заказанного\_товара, Общая\_стоимость, Дата\_поставки, Заказчик, Адрес\_заказчика, Расчетный\_счет\_заказчика, Телефон\_заказчика**.  Данные подлежат нормализации.  Чтобы таблица соответствовала **1-й нормальной форме** (1NF), необходимо, чтобы все значения ее полей были неделимыми и не вычисляемыми, а все записи – уникальными (не должно быть полностью совпадающих строк).  Если много заказчиков купят один и тот же товар, то в таблице будут повторяться одни и те же исходные реквизиты товара. Для уменьшения избыточности информации, нужно преобразовать первоначальную таблицу к двум таблицам:  **Наименование\_товара, Цена, Количество\_на\_складе**  и  **Наименование\_заказанного\_товара, Цена\_заказанного\_товара, Количество\_зака-занного\_товара, Общая\_стоимость, Дата\_поставки, Заказчик, Адрес\_заказчика, Расчетный\_счет\_заказчика, Телефон\_заказчика**.  Первая таблица соответствует первой нормальной форме, а во второй таблице имеется избыточность. Если один заказчик купит много товаров, то в таблице будут повторяться одни и те же исходные реквизиты заказчика. К тому же поле **Общая\_стоимость** может быть вычислено по полям **Цена\_заказанного\_товара** и **Количество\_заказанного\_товара**.  Поэтому вторую таблицу надо разбить на две:  **Заказчик, Адрес\_заказчика, Расчетный\_счет\_заказчика**  и  **Наименование\_заказанного\_товара, Цена\_заказанного\_товара, Количество\_за-казанного\_товара, Дата\_поставки, Заказчик, Телефон\_заказчика**.  Поле **Общая\_стоимость**, как вычисляемое, в структуру таблицы не включается. |
| 2.Изучить способ приведения ко *второй нормальной форме*, проанализировав пример в правой части. | Чтобы таблица соответствовала **2-й нормальной форме** (2NF), необходимо, чтобы она находилась в 1-й нормальной форме и все не ключевые поля полностью зависели от ключевого.  Из полученных трех таблиц первые две соответствуют второй нормальной форме, а в третьей таблице нет ключевого поля. Если в качестве такового взять поле **Наименование\_заказанного\_товара**, то оно может принимать одно и то же значение для различных заказчиков. Поэтому требуется ввести новое поле, которое являлось бы первичным ключом для всех остальных.  Таким ключом может стать поле **Номер\_заказа**, которое надо добавить, т. е. третья таблица теперь должна содержать поля:  **Номер\_заказа, Наименование\_заказанного\_товара, Цена\_заказанного\_товара, Количество\_заказанного\_товара, Дата\_поставки, Заказчик, Телефон\_заказчика**.  Определим имена таблиц:  ТОВАРЫ (**Наименование товара, Цена, Количество\_на\_складе**);  ЗАКАЗЧИКИ (**Заказчик, Адрес\_заказчика, Расчетный\_счет\_заказчика**);  ЗАКАЗЫ (**Номер заказа, Наименование\_заказанного\_товара, Цена\_заказанного\_ товара, Количество\_заказанного\_товара, Дата\_поставки, Заказчик, Телефон\_заказчика*)***.  Ключевые поля подчеркнуты. |
| 3.Ознакомиться с *третьей нормальной формой*. | Для перехода к **3-й нормальной форме** (3NF), необходимо обеспечить, чтобы все таблицы находились во 2-й нормальной форме и все не ключевые поля в таблицах не зависели взаимно друг от друга.  В таблице ЗАКАЗЫ поля **Заказчик** и **Телефон\_заказчика** взаимно зависимы. Чтобы привести таблицу к 3 нормальной форме надо поле **Телефон\_заказчика** переместить в таблицу ЗАКАЗЫ.  Теперь полученные таблицы соответствуют всем требованиям.  ТОВАРЫ (**Наименование товара, Цена, Количество\_на\_складе**);  ЗАКАЗЧИКИ (**Заказчик, Адрес\_заказчика, Расчетный\_счет\_заказчика, Телефон\_заказчика**);  ЗАКАЗЫ (**Номер заказа, Наименование\_заказанного\_товара, Цена\_заказанного\_ товара, Количество\_заказанного\_товара, Дата\_поставки, Заказчик*)***. |
| 4. Определить *типы данных* для полей нормализованных таблиц базы данных **ПРОДАЖИ**. | MSS поддерживает следующие типы данных: числовые, символьные, для хранения даты и времени, денежные, двоичные и специальные.   | **Тип данных** | **Описание** | **Тип данных** | **Описание** | | --- | --- | --- | --- | | **integer**  **(int)** | Целочисленные значения, занимают 4 байта. | **date** | Дата, занимает 3 байта | | **smallint** | Целочисленные значения, занимают 2 байта | **time[(p)]**  **0 ≤ p ≤ 7** | Время, занимает от 3 до 5 байтов. **p** − колич. знаков после точки в секундах | | **tinyint** | Неотрицательные целочисленные значения, занимают 1 байт | **smalldatetime** | Дата и время, занимает 2 байта | | **bigint** | Целочисленные значения, занимают 8 байт | **datetime** | Дата и время, занимает 4 байта | | **real** | Вещественные числа с плавающей точкой | **datetime2** | Дата и время, занимает 8 байтов | | **decimal (p[,s] )**  **(dec)**  или  **numeric (p[,s] )** | Вещественные значения с фикс. точкой, **р** − общее количество цифр, **s** − количество цифр после точки. Занимает от 5 до 17 байт. | **сhаr[(n)]** | Строки фиксированной длины из однобайтовых симво­лов, n − количество символов | | **float[(p)]** | Вещественные числа с плавающей точкой. Если **р** < 25 − одинарная точность (4 байта), при **р** > 25 − двойная точность (8 байтов) | **nchar[(n)]** | Строки фиксированной длины символов Unicode. Каждый символ занимает 2 байта. | | **money** | Денежные значения, занимают 8 байтов | **varchar[(n)]** | Строки переменной длины из однобайтовых симво­лов | | **smallmoney** | Денежные значения, занимают 4 байта | **nvarchar[(n)]** | Строки переменной длины символов Unicode − 2 байта | | **binary(n)** | Задает битовую строку, длиной ровно **n** байтов | **varbinary(n)** | Задает битовую строку, длиной не более **n** байтов | |

→ ***Теория (моя)***

**ЗАДАНИЯ**

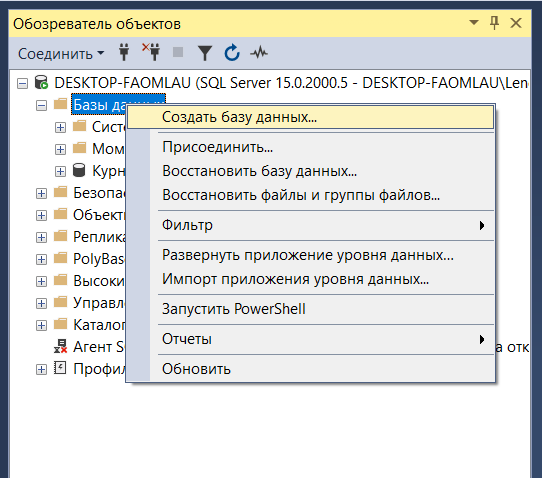
*В соответствии со своим вариантом, номер которого определяет преподаватель, провести нормализацию исходной информации из таблицы, представленной ниже, создав как минимум три таблицы. При необходимости использовать дополнительные поля.*

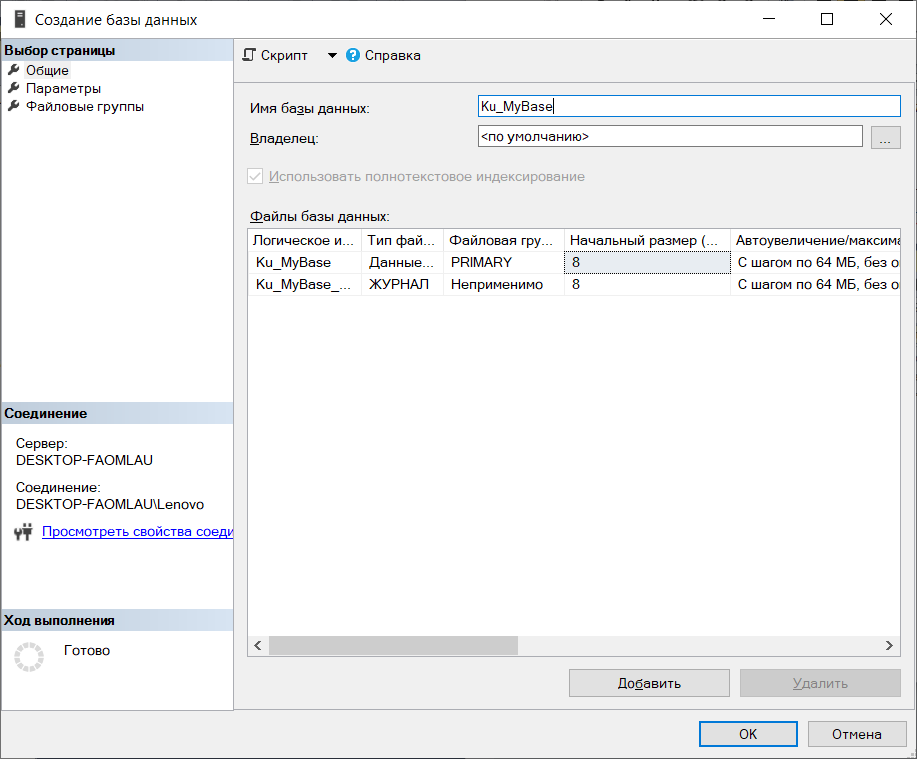
*Создать базу данных (имя базы −* ***X\_MyBase****,**где* ***X*** *− первые буквы своей фамилии) с помощью команд* ***Server Management Studio****, определить структуру таблиц, установить связи и заполнить таблицы информацией.*

|  |  |
| --- | --- |
| 8 | **Грузовые перевозки.** Компания осуществляет перевозки по различным маршрутам. Информационные поля: Название маршрута, Дальность, Количество дней в пути, Оплата, Фамилия водителя, Имя, Отчество, Стаж, Дата отправки, Дата возвращения. |

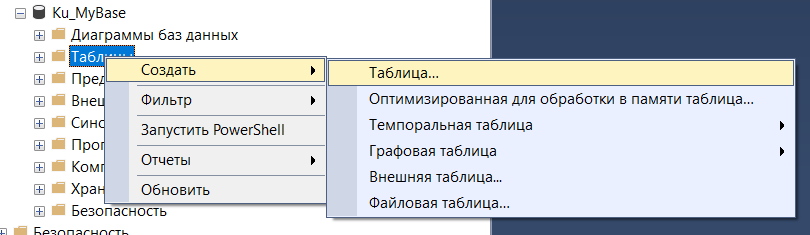
→ ***Выполнение заданий***

Создаю базу данных (имя базы – Ku\_MyBase, где Ku – первые буквы моей фамилии) с помощью команд Server Management Studio:



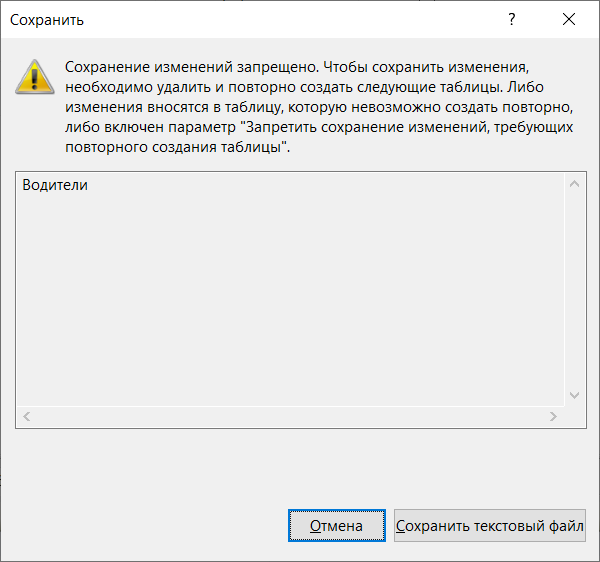


Создаю таблицы:



**Отключить параметр «Запретить сохранение изменений, требующих повторного создания таблицы»**

При внесении изменений в таблицу и попытке их сохранения возникает предупреждение:

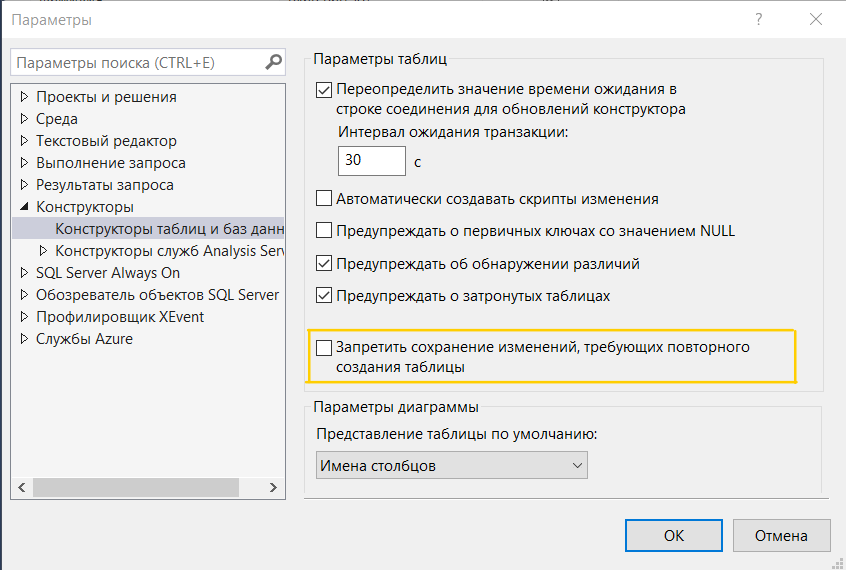


Если Вы не хотите вникать в SQL, то Вы можете просто отключить параметр «Запретить сохранение изменений, требующих повторного создания таблицы» и в таком случае Вы сможете вносить в таблицы все перечисленные выше изменения, которые ранее были недоступны, включая изменение порядка столбцов.

Однако Microsoft не рекомендует отключать этот параметр, ссылаясь на то, что при определенных обстоятельствах сохранение изменений, требующих повторного создания таблицы, может привести к потере метаданных и прямой потере данных. Например, если у таблицы включен функционал «Отслеживания изменений».

В большинстве случаев потери данных, конечно же, не будет происходить, поэтому данный параметр отключить можно, но лучше использовать SQL.

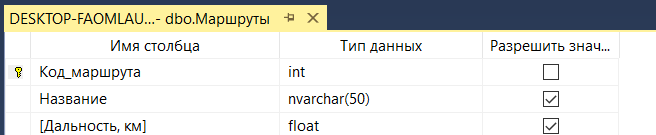
*Чтобы отключить данный параметр, зайдите в SSMS в меню «Сервис -> Параметры» и на вкладке «Конструкторы» снимите галочку «Запретить сохранение изменений, требующих повторного создания таблицы» и нажмите «ОК».*



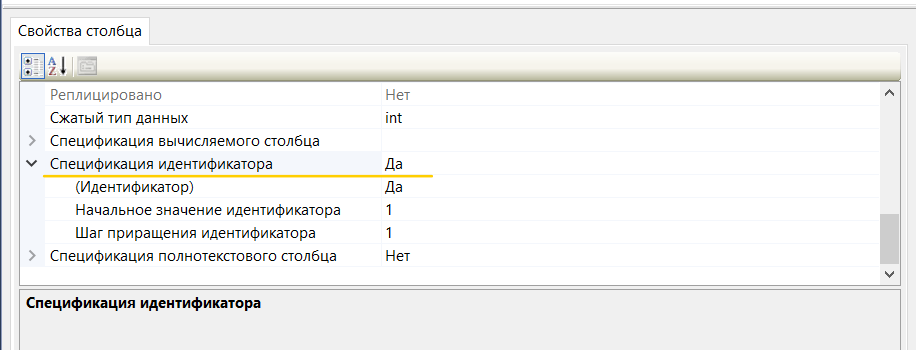
Я решила создать 4 таблицы.

**Структура таблиц представлена ниже:**

1) *Маршруты*

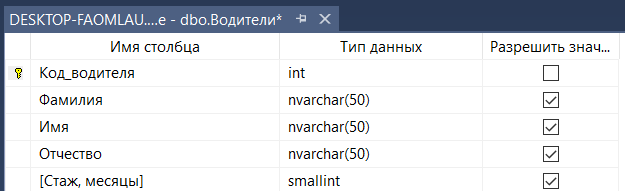


Тип данных столбца Код\_маршрута – int, при этом в свойствах я делаю его столбцом идентификаторов:



Дальность нужно будет писать в километрах.

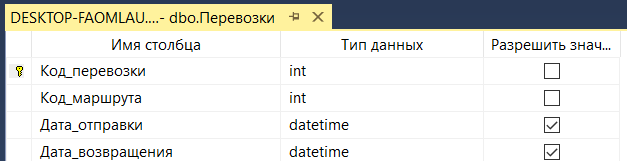
2) *Водители*



Код\_водителя, как и в случае с таблицей «Маршруты», ─ столбец идентификаторов.

Стаж нужно будет прописывать в месяцах

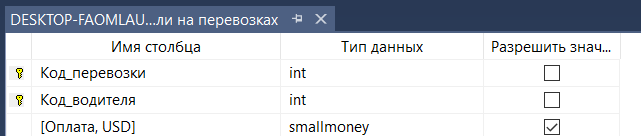
3) *Перевозки*



Код\_перевозки – идентификатор.

Код\_маршрута – внешний ключ, ссылающийся на первичный ключ таблицы «Маршруты».

4) *Водители на перевозках*

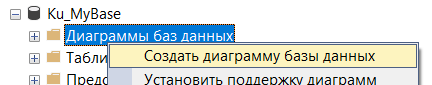


Данная таблица нужна, так как одну перевозку могут осуществлять два водителя.

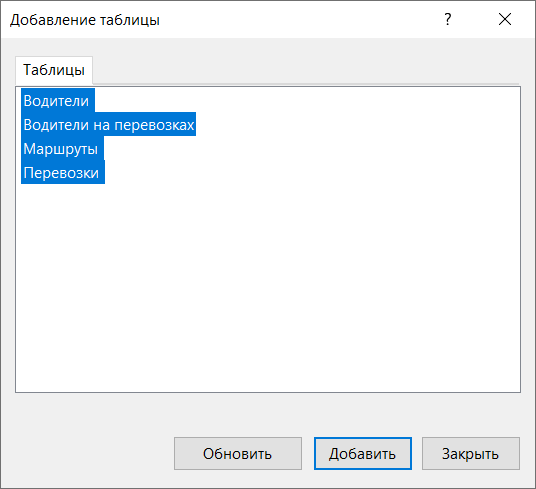
Помимо этого поле «Оплата, USD» зависит от дальности маршрута и от стажа водителя.

Кстати, в этот раз у нас составной ключ.

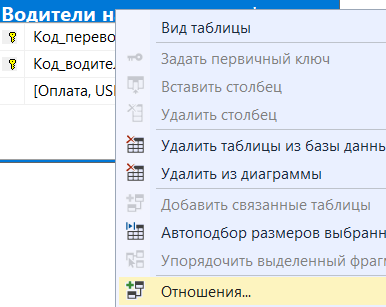
Задаю отношения между таблицами:



Выбираю все таблицы:

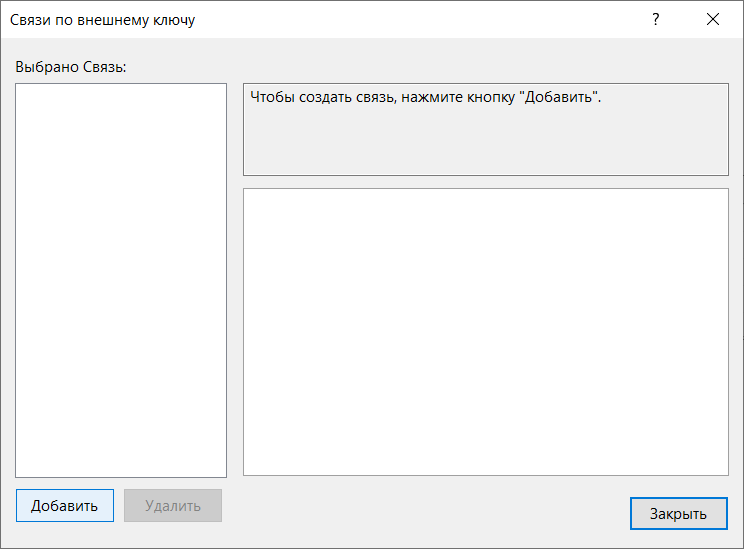


Задаем связи в окне с появившимися таблицами:

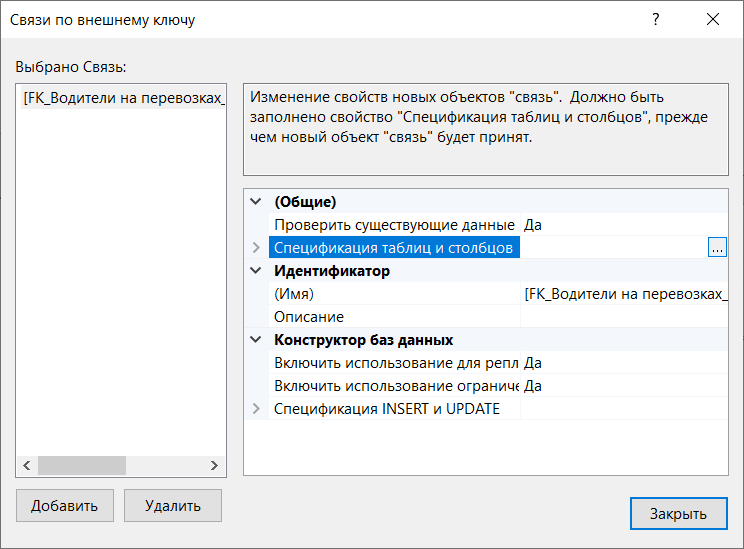


Таблица, на которой мы начали устанавливать связь, станет таблицей внешнего ключа. Таким образом, в окне, которое после этого появится, мы можем лишь выбрать таблицу первичного ключа и установить то, какие поля будут связаны.

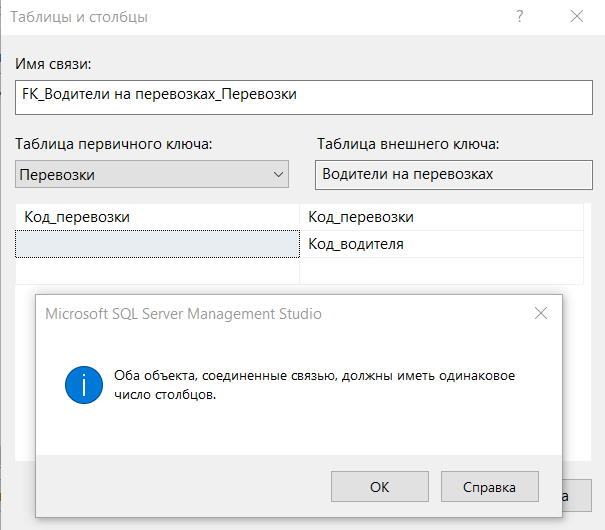
Добавляем новую связь:



Нажимаем троеточие в пункте **Спецификация таблиц и столбцов**:

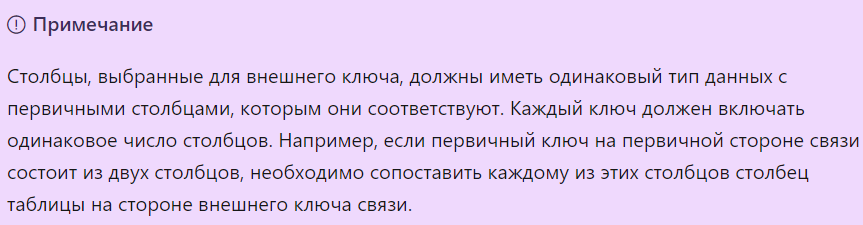


Задала связь, но появилось такое окно.

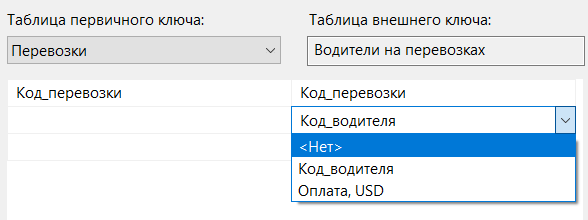


Вот что об этом говорят в спецификации:

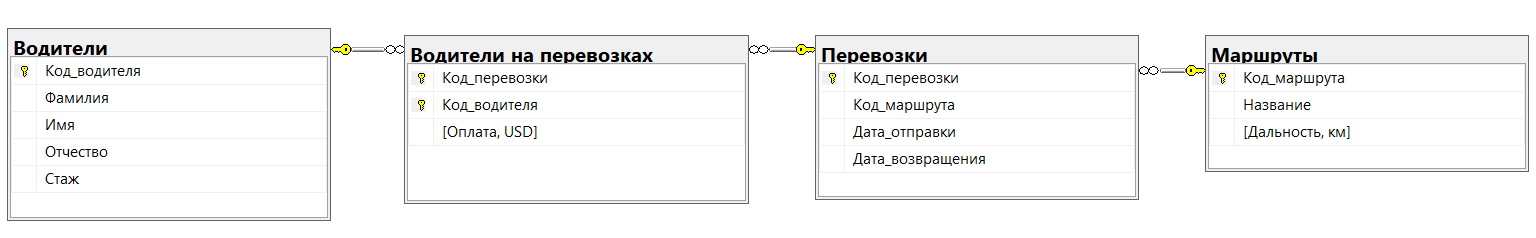
<https://docs.microsoft.com/ru-ru/sql/ssms/visual-db-tools/tables-and-columns-dialog-box-visual-database-tools?view=sql-server-ver15>



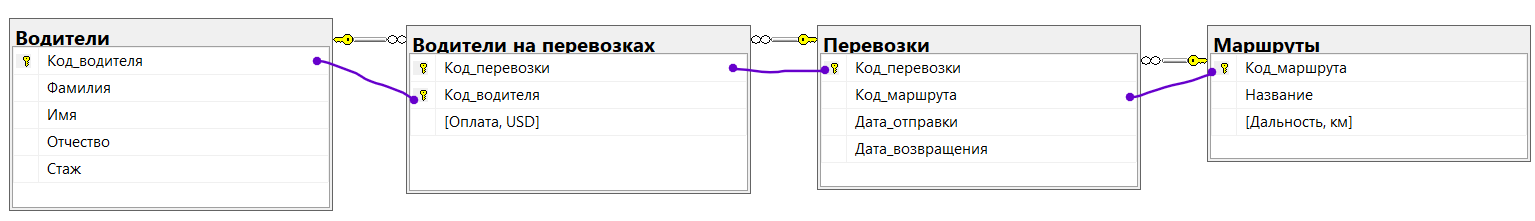
Поэтому в поле, где написано «Код\_перевозки» выбираю «Нет» из выпадающего списка:



**Связи между таблицами:**

****

Связи между столбцами обозначены фиолетовым:

****