Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«Нижегородский Государственный Университет им.

Н.И.Лобачевского» (ННГУ)

Национальный исследовательский Университет

Институт Информационных Технологий Математики и Механики

**Отчёт по лабораторной работе №2**

Выполнил:

студент группы 3821Б1ПМ3

Заботин Максим Андреевич

Проверила:

доцент кафедры математического

обеспечения и суперкомпьютерных

технологий

Малкина Е.В

Нижний Новгород

2022 г.

Оглавление

[Введение 3](#_Toc131630457)

[Цели работы 4](#_Toc131630458)

[Задания 4](#_Toc131630459)

[Ход работы 5](#_Toc131630460)

[Таблица «Поставщики» 5](#_Toc131630461)

[Таблица «Детали» 6](#_Toc131630462)

[Таблица «Поставки» 8](#_Toc131630463)

[Схема базы данных 9](#_Toc131630464)

[Выводы 12](#_Toc131630465)

# Введение

Базы данных-это сложные, эффективные структуры, позволяющие хранить в себе разнообразную информацию о каком-то количестве объектов, а также связи между этими объектами, что позволяет удобно размещать информацию, а также находить и обновлять её. Предметами, описываемыми базами данных могут быть совершенно различные сущности, которые состоят из нескольких частей, значения которых содержатся в базах данных. При описании предметной области важно правильно определить, какие будут сущности для неё, а также как их описать и связать, чему мы научились в ходе предыдущей лабораторной работе. В прошлой лабораторной работе мы создали реляционную модель базы данных для конторы, занимающейся закупкой авто-деталей, и в этой лабораторной работе мы по ранее подготовленной модели создадим базу данных. Для этого будем использовать приложение «Microsoft Access».

Данная лабораторная работа поможет получить начальные навыки по созданию базы данных по уже готовой модели, и очень полезна, ведь в дальнейшем мы будем совершенствовать эту базу данных, добавляя всё более сложные элементы и делая её всё сложнее и правильнее.

# Цели работы

Приобретение первичных навыков создания структуры реляционной базы данных и первоначальный ввод данных в базу данных с использованием СУБД MS Access. Создание структуры реляционной базы данных для построенной в лабораторной работе 1 концептуальной модели предметной области.

# Задания

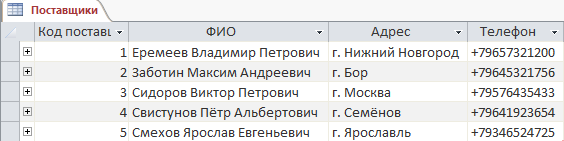
1. При помощи среды СУБД MS Access создать структуры таблиц для представления предметной области в рамках реляционной модели с указанием типов данных и их характеристик.
2. Для каждой создаваемой таблицы:
   1. Определить условия на значения и сообщения об ошибках некоторых полей.
   2. Определить начальное значение для некоторых полей.
   3. Задать ключ.
   4. Задать внешний ключ (если он есть).
   5. Определить (если это возможно) значения некоторых полей с помощью мастера подстановок.
   6. Определить обязательные поля.
3. Определить схему базы данных, связи между таблицами и наложить условия целостности на таблицы, связанные отношением «один-к-многим». Показать на примерах, что меняется при включении/выключении каждого из флажков «Обеспечение целостности данных», «Каскадное обновление связанных записей» и «Каскадное удаление связанных записей».
4. Ввести данные в таблицы. При вводе выяснить, что дает наложение условий на значения полей

# Ход работы

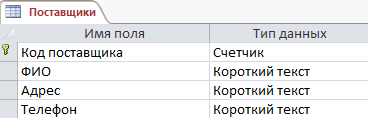
Итак, в ходе работы была создана база данных, по модели, написанной в первой лабораторной работе, по указанным там схемам были созданы функциональные зависимости и далее будут представлены все получившиеся таблицы в базе данных, в виде снимков экрана, а также будут представлены условия на значения, значения по умолчанию, для каждого из атрибутов, а также будут определены обязательные поля.

## Таблица «Поставщики»

В режиме таблицы данная часть будет выглядеть таким образом:

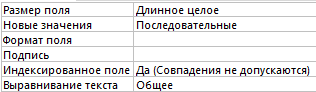


Теперь посмотрим, как выглядит конструктор данной таблицы:



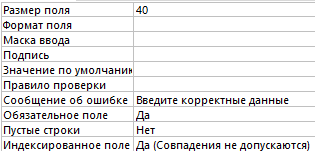
Сразу видим, что ключом здесь является Код поставщика, типа счётчик, а все остальные атрибуты являются текстовыми. Рассмотрим поподробнее типы данных и характеристики для каждого атрибута.

Код поставщика:



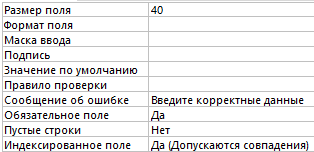
Данный атрибут будет размером длинного целого, так как счётчик, то значения будут последовательными, и совпадений не будет допускаться (это поле СУБД сама будет заполнять при добавлении новой записи)

ФИО:



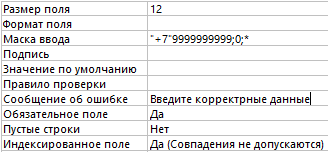
Выставим ограничение в длине поля в 40 символов, ведь больше навряд ли пригодится, это поле будет обязательным для заполнения и не будет иметь значение по умолчанию, не будет допускать пустых строк, ведь не бывает пустых ФИО, также мы примем в расчёт, что в нашей конторе не попадутся несколько одинаковых поставщиков. Сообщение об ошибке будет таким, как представлено на рисунке.

Адрес:



Для данного поля, типа короткий текст сделаем максимально 40 символов, будем допускать совпадения адреса, ведь поставщики могут находиться на одной базе деталей. Пустых строк быть не может, и это является обязательным полем.

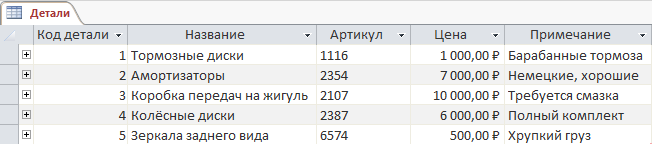
Телефон:



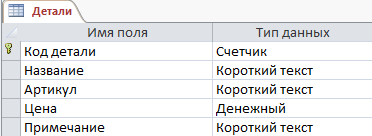
Размер данного поля будет фиксировано 12 символов, а также мы установим маску ввода, которая будет по виду как номер телефона, чтобы было удобнее вводить и воспринимать это поле. Совпадений по телефону быть не может, и это обязательное поле, в котором не может быть пустых строк.

## Таблица «Детали»

В режиме таблицы данная часть будет выглядеть таким образом:

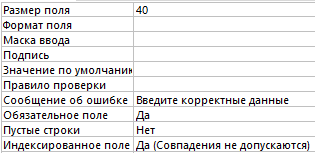


Теперь посмотрим, как выглядит конструктор данной таблицы:



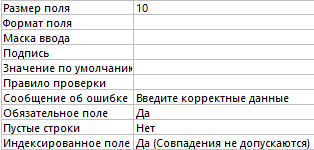
Первичным ключом является код детали, типа счётчик, что позволяет СУБД самостоятельно индексировать записи и исключать повторений. Характеристики совпадают с Кодом поставщика и можно не дублировать, перейдём сразу к другим атрибутам.

Название:



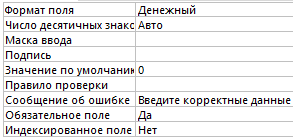
Максимальная длина в 40 символов, это поле является обязательным и не допускает повторений, а также пустых строк. Сообщение об ошибке добавлено. Перейдём к другому полю.

Артикул:



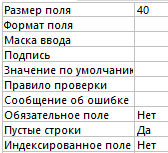
Характеристики данного атрибута схожи с предыдущим, за исключением того, что размер артикула будет в 10 символов, и будет состоять не только из букв, но и цифр, символов.

Цена:



Для удобства цену сделаем денежным форматом, что логично. В таком случае СУБД автоматически будет подставлять валюту. Значение по умолчанию будет 0, это поле будет обязательным, а также, цена может повторяться.

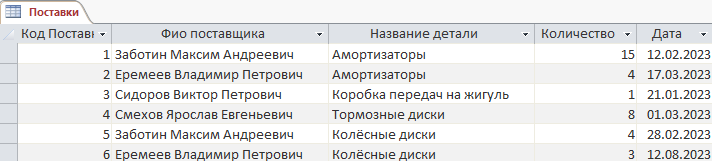
Примечание:



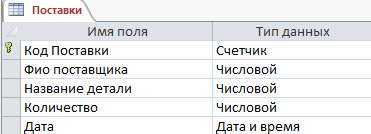
Примечание будет использоваться для каких-либо пометок, не превышающих 40 символов, потому что не надо слишком длинные сообщения хранить для пометок. Это поле необязательно и может повторяться.

## Таблица «Поставки»

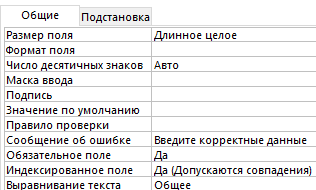
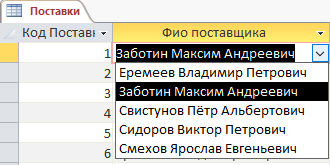
В режиме таблицы данная часть будет выглядеть таким образом:

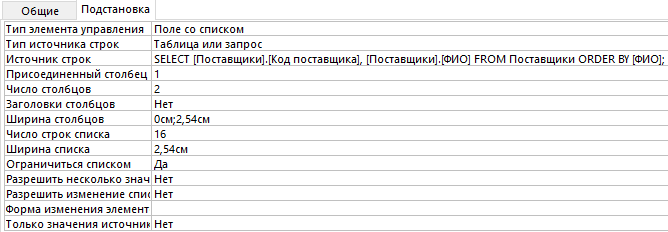


Теперь посмотрим, как выглядит конструктор данной таблицы:



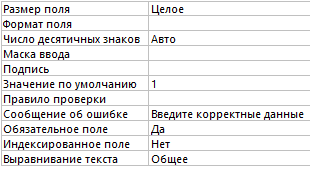
Код поставки по характеристикам аналогичен двум другим кодам в данной таблице, он также является первичным ключом. Счётчики управляются СУБД и не пересекаются, что позволяет удобно индексировать записи. ФИО поставщика и название детали заполняются с помощью мастера подстановок, которым управляет сама СУБД. Они сделаны числовыми, потому что по сути первым делом наследуются коды поставщиков и деталей, являющиеся внешними ключами, а уже для наглядности выводятся текстовые значения этих полей, ведь удобнее смотреть по тексту, а не по коду. СУБД сделает так, чтобы коды не путались, и сама всё подставляет. Теперь, после таких действий в режиме таблицы при создании новой записи можно сразу выбирать поставщика и деталь, которую он поставляет, а синхронизации с таблицами обеспечат актуальность данных



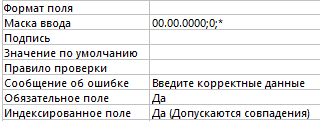
Данные атрибуты имеют одинаковую характеристику и дублирование считаю бессмысленным. Перейдём к следующим атрибутам.

Количество:



Количество деталей в поставке сделаем по умолчанию равным 1, а также обозначим это поле обязательным. Поле будет размером целого числа, потому что дробных деталей быть не может. Сообщение об ошибке присутствует.

Дата:



Ну и, наконец, последний атрибут в нашей таблице, это дата. Она выбрана формата дата и время, что логично, но за маску ввода взята лишь дата, ведь точного времени прибытия поставки предугадать и запросить невозможно из-за различных ситуаций. Поле является обязательным для заполнения, а также допускает дублирования данных, ведь в один день может быть несколько поставок.

Итак, была описана вся структура базы данных, с указанием характеристик атрибутов и в целом видом таблиц, а также база данных была заполнена, поэтому можно считать, что полностью выполнены задания 1 и 2.

Далее перейдём к схеме базы данных

## Схема базы данных

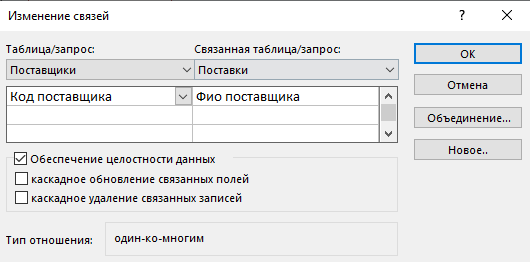
Схему представим с помощью встроенной в среду разработки функции, которая показывает зависимости между таблицами

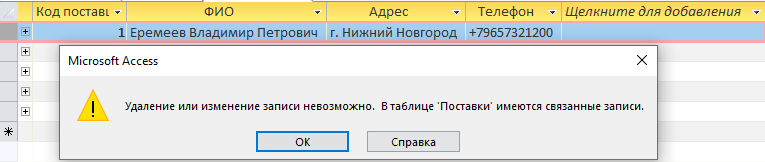


При использовании связей вызывается мастер подстановки, который связывает таблицы и позволяет удобно использовать данные.

Параметры для связей «Обеспечение целостности данных» и «Каскадное обновление связанных записей» и «Каскадное удаление связанных записей» возможно устанавливать для связей между таблицами «один к одному» и «один ко многим». Целостность данных позволяет поддерживать связи между таблицами, защищает от случайных удалений и изменений.

При применении первого параметра к связанным таблицам обеспечивается, что в вторичную таблицу не может быть добавлена запись, которой не существует в первичной таблице (не существует такого ключа), также в первичной таблице не получится удалить запись, если не удалены связанные с ней записи во вторичной таблице, и, наконец, если в первичной таблице изменяется значение ключевого значения, то это приводит к изменению соответствующих ему значений во вторичной таблице. При установке данного параметра к связанным таблицам, то если пользователь попытается сделать что-то не предусмотренное данной связью, Access выдаст предупреждение.





Если установлены 2 других параметра, а именно, каскадные удаления и изменения, то СУБД будет автоматически изменять данные сразу во всех таблицах, и при этом обеспечивая целостность данных. При удалении записи в первичной таблице, будут удаляться все записи во вторичной. Это может быть не всегда полезно, но в большинстве случаев очень удобно, чтобы несколько раз самостоятельно не делать удаления в разных таблицах.

Пример: в таблицу Поставки невозможно добавить новую запись, с ключом, не существующим в таблице Поставщики или Детали (то есть несуществующего поставщика или деталь).

# Выводы

В ходе данной лабораторной работы была разработана модель базы данных для конторы, занимающейся поставками автозапчастей. База данных была создана по реляционной модели, разработанной в лабораторной работе 1. Были заполнены все поля, а также для всех атрибутов были заданы соответствующие им характеристики. База данных приобрела соответствующие связи функциональные зависимости. Также были изучены основные понятия, связанные с моделированием и созданием базы данных с помощью приложения «Microsoft Access», что понадобится в дальнейшем.

# Заключение

Базы данных-очень увлекательные и полезные структуры, хранящие данные и связи между ними. В дальнейшем разработанная в этой лабораторной работе база данных будет совершенствоваться и становиться сложнее и гармоничнее. Необходимо и дальше совершенствовать свои навыки.