

Лабораторная работа №3

ТЕМА: Знакомство с основными функциями системы управления базами данных ACCESS под Windows.

ЦЕЛЬ: Научить приемам проектирования баз данных и методам эффективного распределения данных между таблицами.

ПЛАН ЗАНЯТИЯ:

- I.** Основные теоретические сведения.
- II.** Знакомство с основными принципами проектирования баз данных.
- III.** Ознакомление с основными функциями системы управления базами данных ACCESS.
- IV.** Создание базы данных.
- V.** Связывание таблиц.

I. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

MS ACCESS – это система управления базами данных под управлением Windows. MS ACCESS входит в пакет Microsoft Office.

БАЗА ДАННЫХ (БД) предназначена для хранения и организации информации, ввода новых записей, изменения уже существующих записей, выборки и управления информацией.

ОБЪЕКТОМ ОБРАБОТКИ MS ACCESS является файл базы данных, имеющий произвольное имя и расширение **.MDB**

В этот файл входят основные **ОБЪЕКТЫ** MS ACCESS:

- таблицы,
- формы,
- запросы,
- отчеты,
- макросы,
- модули.

ТАБЛИЦА является **БАЗОВЫМ** объектом MS ACCESS. Все остальные объекты являются производными и создаются только на базе ранее подготовленных таблиц.

ТАБЛИЦА – это объект, предназначенный для хранения данных в виде записей (строк) и полей (столбцов). Обычно каждая таблица используется для хранения сведений по одному конкретному вопросу.

ФОРМА не является самостоятельным объектом MS ACCESS, она просто помогает вводить, просматривать и модифицировать информацию в таблице или запросе.

ЗАПРОСЫ И ОТЧЕТЫ выполняют самостоятельные функции.

ЗАПРОС – объект БД, позволяющий получить нужные данные из одной или нескольких таблиц.

ОТЧЕТ – объект БД, предназначенный для печати данных.

МАКРОС – набор специальных макрокоманд (открыть форму, напечатать отчет).

МОДУЛЬ – это программа, написанная на специальном языке.

Каждый объект MS ACCESS имеет имя. Длина имени не более 64 символов, включая русские буквы и пробел (исключение – точка и некоторые специальные символы).

ПОЛЕ – элемент таблицы, столбец или ячейка.

ЗАПИСЬ – полный набор данных об определенном объекте, в режиме таблицы это строка.

С каждым объектом MS ACCESS предусмотрено два **РЕЖИМА РАБОТЫ**:

ОПЕРАТИВНЫЙ РЕЖИМ – когда просматривается, изменяется, выбирается информация;

РЕЖИМ КОНСТРУКТОРА – когда создается, изменяется макет, структура объекта.

Кроме того, в файл БД входит еще один документ, имеющий собственное окно – **СХЕМА ДАННЫХ**. В этом окне создают, просматривают, изменяют и разрывают связи между таблицами. Эти связи помогают контролировать данные, создавать запросы и отчет.

ПРИМЕЧАНИЕ: Создание таблицы в Access 2010 представляет собой процесс очень похожий на создание таблицы в Access 2007, однако в Access 2010 существует несколько новых функций, которые относятся к таблицам, это:

МАКРОСЫ ДАННЫХ. Теперь с событиями в таблице можно связывать макросы, позволяющие выполнять определенные действия при изменении, вставке или удалении записи. Например, с помощью макросов можно проверять данные или выполнять вычисления.

ТИП ВЫЧИСЛЯЕМЫХ ДАННЫХ. Этот новый тип данных позволяет создавать поля на основе вычислений, производимых над другими полями в той же таблице. Например, можно создать поле «Итог по строке», содержащее произведение значений полей «Количество» и «Цена за единицу». Такое поле будет автоматически обновляться при изменении значений полей «Количество» и «Цена за единицу».

ВЗАИМОДЕЙСТВИЕ С ВЕБ-СЛУЖБАМИ. Кроме подключения к внешним источникам данных, таким как книги Excel и списки SharePoint, теперь можно подключаться к данным на веб-сайтах с интерфейсом веб-служб. Например, можно подключиться к данным на сайте интернет-магазина и создать собственное приложение просмотра товаров или оформления заказов.

МОДЕЛИ ВМЕСТО ШАБЛОНОВ ТАБЛИЦ И ПОЛЕЙ. В Access 2007 были представлены шаблоны таблиц и полей. Шаблоны таблиц – это пустые таблицы, на основе которых можно создавать таблицы, совместимые с указанными ниже встроенными списками SharePoint.

- контакты,
- задачи,
- вопросы,
- события,
- основные средства.

В Access 2010 для добавления заранее созданных компонентов в базу данных используются модели. Модели могут содержать таблицы и объекты базы данных других типов, такие как запросы и формы.

II. ЗНАКОМСТВО С ОСНОВНЫМИ ПРИНЦИПАМИ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

Этапы проектирования баз данных

При создании базы данных главный вопрос, который необходимо решить – это вопрос о ее наиболее эффективной структуре. Правильно спроектированная база данных позволяет:

- обеспечить *быстрый доступ* к данным;
- исключить ненужное *повторение данных*, которое приводит к нерациональному использованию дискового пространства компьютера, и, кроме того, может явиться причиной ошибок при вводе;
- обеспечить *целостность данных* таким образом, чтобы при изменениях в одном объекте автоматически происходили соответствующие изменения в связанных с ним объектах.

В качестве примера предположим, что вы хотите создать базу данных для учета заказов в библиотеке. Вся информацию для этой задачи можно разместить в одной таблице **"ВЫДАЧА КНИГ"** (таблица 1), которая будет содержать следующие данные:

- информацию о каждой книге – код (шифр) книги, автор, название книги, наименование издательства, город, год издания, количество страниц, стоимость, тематика;
- сведения о читателях (фамилия, имя, отчество, домашний адрес, телефон);
- дату заказа данной книги.

Таблица 1. Основные поля таблицы «Выдача книг»

Код книги	Автор	Название книги	Наименование издательства	Город	Год издания	Объем	Стоимость	Тема	Дата заказа	Фамилия читателя	Имя читателя	Отчество читателя	Домашний адрес	Домашний телефон
-----------	-------	----------------	---------------------------	-------	-------------	-------	-----------	------	-------------	------------------	--------------	-------------------	----------------	------------------

Обратите внимание на то, что фамилию, имя и отчество мы поместили в отдельные поля. Информацию в полях надо хранить в виде минимально возможных элементов, т.к. наличие в одном поле нескольких элементов существенно затрудняет извлечение отдельного элемента из этого поля.

Такую таблицу можно рассматривать как *однотабличную базу данных*. Однако, при подобной организации в ней содержится значительное количество повторяющейся информации. Например, сведения о конкретном читателе библиотеки будут *дублироваться* при регистрации каждого нового сделанного им заказа. То же самое можно сказать и про сведения о книгах. Все это приведет к тому, что придется тратить значительное время на ввод повторяющихся данных. При этом, во-первых, возрастает вероятность ошибок при вводе, а во-вторых, в случае необходимости изменения какой-либо информации, например при изменении телефона читателя,

придется корректировать все записи, содержащие сведения о заказах этого читателя. Кроме того, наличие повторяющейся информации приводит к неоправданному увеличению размера базы данных.

Для решения подобных проблем используется *процесс уменьшения избыточности информации в базе данных*, который называется **нормализацией**. Остановимся на некоторых практических аспектах нормализации таблиц.

ПЕРВОЕ ПРАВИЛО ПРИ ПРОЕКТИРОВАНИИ БАЗЫ ДАННЫХ: *в таблице должны отсутствовать повторяющиеся группы полей*. В нашем примере это можно осуществить, переместив поля с повторяющейся информацией о читателях в отдельную таблицу "**ЧИТАТЕЛИ**". В результате получаем две таблицы: таблицу "**ВЫДАЧА КНИГ**" с информацией о заказанной книге и таблицу "**ЧИТАТЕЛИ**", содержащую сведения о читателях.

Таким образом, можно избавиться от дублирования информации о читателях. А чтобы в таблице "**ВЫДАЧА КНИГ**" велась регистрация читателей, заказывающих книги, присвоим каждому читателю уникальный код (записывать его будем в поле **КОД ЧИТАТЕЛЯ** таблицы "**ЧИТАТЕЛИ**") и добавим поле **КОД ЧИТАТЕЛЯ** в таблицу "**ВЫДАЧА КНИГ**", где и будем указывать код читателя, заказавшего книгу. Результат подобного преобразования показан в таблицах 2 и 3.

Таблица 2. Поля таблицы «**ВЫДАЧА КНИГ**» после выделения таблицы «**ЧИТАТЕЛИ**»

Код книги	Автор	Название	Наименование издательства	Город	Год издания	Объем	Стоимость	Тема	Дата заказа	Код читателя
-----------	-------	----------	---------------------------	-------	-------------	-------	-----------	------	-------------	--------------

В этом случае общий объем хранимых данных значительно сокращается. Теперь при регистрации нового заказа в таблице "**ВЫДАЧА КНИГ**" вместо полной информации о читателе указывается его личный код, т.е. создается дубликат только небольшого элемента данных – поля **КОД ЧИТАТЕЛЯ**. Наличие же совпадающих полей в обеих таблицах позволяет связать данные о заказанной книге с данными о соответствующем читателе. Отметим, что связь между этими таблицами – это связь типа *один-ко-многим*.

Таблица 3. Поля таблицы «**ЧИТАТЕЛИ**»

Код читателя	Фамилия	Имя	Отчество	Домашний адрес	Домашний телефон
--------------	---------	-----	----------	----------------	------------------

Однако в таблице "**ВЫДАЧА КНИГ**", как и прежде, вместе с каждым заказом хранится информация о книге. Поэтому имеет смысл также отделить ее в отдельную таблицу "**КНИГИ**", а в таблице "**ВЫДАЧА КНИГ**" фиксировать только код книги. В результате в таблице "**ВЫДАЧА КНИГ**" остаются только три поля: **КОД ЧИТАТЕЛЯ**,

КОД КНИГИ и **ДАТА ЗАКАЗА**. Связь между таблицами осуществляется через совпадающие значения полей **КОД КНИГИ**, и связь эта – типа *ОДИН-КО-МНОГИМ*.

Точно таким же образом из таблицы "**КНИГИ**" можно выделить в отдельные таблицы информацию об издательствах и информацию о тематике книг. Полученная схема базы данных представлена на рисунке 1.

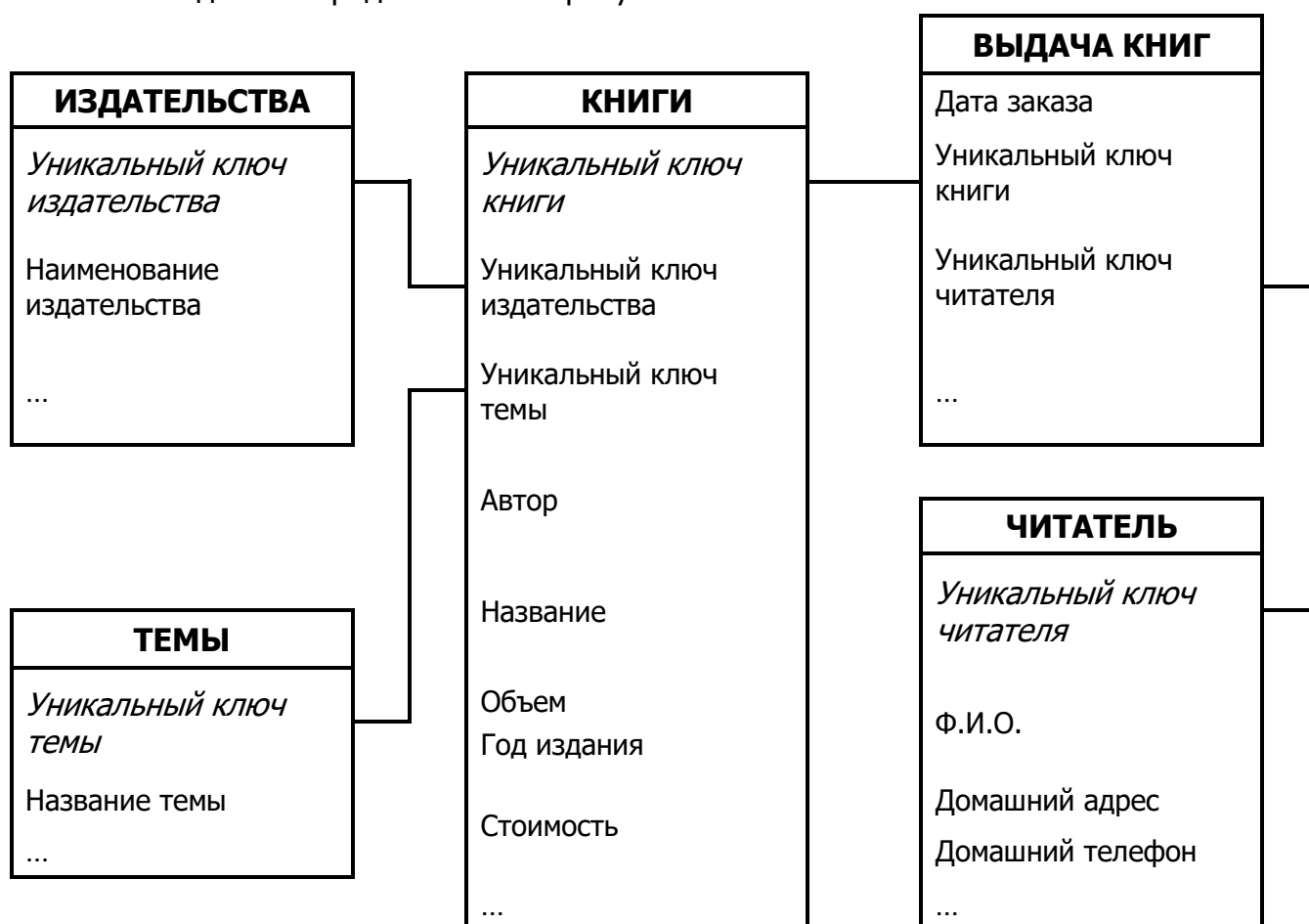


Рис. 1. Таблицы базы данных Библиотека

ВТОРОЕ ПРАВИЛО ПРИ ПРОЕКТИРОВАНИИ БАЗЫ ДАННЫХ: *таблица не должна иметь повторяющихся записей.* Для удовлетворения этого условия таблица должна иметь *уникальный идентификатор* или *первичный ключ*, то есть поле, значения в котором не повторяются. Поле **КОД КНИГИ**, содержащее шифр книги, является уникальным и поэтому может быть первичным ключом для таблицы "**КНИГИ**". Поле **КОД ЧИТАТЕЛЯ** таблицы "**ЧИТАТЕЛИ**" также может быть первичным ключом. А в таблице "**ВЫДАЧА КНИГ**" нет полей, содержащих уникальные данные. Поэтому для нее можно создать "искусственный" первичный ключ: добавить поле, которое будет содержать номер записи.

В результате описанных выше операций проект базы данных "Библиотека" включает в себя пять таблиц, показанных на рисунке 1.

Обратите внимание, что ключевые поля выделены *курсивом*, а все связи в нашей базе данных – это связи типа *ОДИН-КО-МНОГИМ*.

Вы, наверное, заметили, что каждый раз при создании новой таблицы в существующую таблицу включается дополнительное поле, связывающее эти таблицы. Эти "связывающие" поля обычно называют *чужими ключами*. В процессе создания базы данных необходимо внимательно следить за созданием чужих ключей, т.к. задаваемые при создании таблиц связи первичных ключей используются чужими

ключами для объединения данных из нескольких таблиц. Для ускорения поиска информации можно создать индексы по чужим ключам.

Итак, при создании базы данных можно выделить следующие этапы:

1. **ПОСТАНОВКА ЗАДАЧИ.** На этом этапе необходимо решить, *какая* информация будет храниться в планируемой базе данных, и *как* она будет использоваться. Исходя этого, можно будет определить, какие таблицы должны храниться в базе данных и какие элементы информации (поля) должны быть включены в каждую таблицу.
2. **ОПИСАНИЕ СТРУКТУРЫ ТАБЛИЦ БАЗЫ ДАННЫХ.** На этом этапе необходимо описать каждую таблицу — указать, какие поля будут содержаться в таблице, тип и размер хранимых в полях данных, установить первичные ключи.
3. **ОПРЕДЕЛЕНИЕ СВЯЗЕЙ МЕЖДУ ТАБЛИЦАМИ.** После определения всех таблиц необходимо указать ACCESS, какие действия надо предпринимать для объединения содержимого таблиц, составляющих базу данных.
4. **ТЕСТИРОВАНИЕ И УСОВЕРШЕНСТВОВАНИЕ.** На этом этапе необходимо ввести несколько записей в каждую таблицу и проверить, можно ли извлечь нужную информацию из таблиц. Попробуйте схематически изобразить формы и отчеты и понять, содержат ли они ожидаемую информацию.

При реализации проекта базы данных необходимо учитывать следующие ограничения:

- таблица базы данных может содержать не более 255 полей;
- таблица не может иметь более 32 индексов;
- общая длина строки таблицы (за исключением полей Мемо и объектов OLE) не может превышать примерно 2Kb;
- на число записей в таблице ограничений не существует, но файл базы данных MS ACCESS не может превышать 2Gb;

III. ОЗНАКОМЛЕНИЕ С ОСНОВНЫМИ ФУНКЦИЯМИ СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ACCESS


ЗАПУСК ПРОГРАММЫ:

Перед началом занятия необходимо выполнить следующие операции на ПК (предполагается, что Microsoft Office уже инсталлирован на диск C:):

1. В левом нижнем углу экрана "Рабочего стола" Windows щелкните по кнопке "Пуск".
2. В появившемся окне поставьте указатель мыши на пункт меню "Программы", в следующем окне щелкните мышью на Microsoft ACCESS.
3. На экране появится окно программы.

ПРИМЕЧАНИЕ: Запуск возможен и с рабочего стола (при условии, что на рабочем столе имеется ярлык Microsoft ACCESS)

ПОРЯДОК РАБОТЫ:

1. Откройте справочную систему MS ACCESS. Для этого необходимо перейти на вкладку **ФАЙЛ** на панели инструментов, и нажать на кнопку **СПРАВКА**. В появившемся окне щелкнуть по ссылке "Справка Microsoft Office". Справку можно открыть и другим способом, щелкнув кнопкой мыши на значок  в правом верхнем углу экрана (см. рисунок 2).

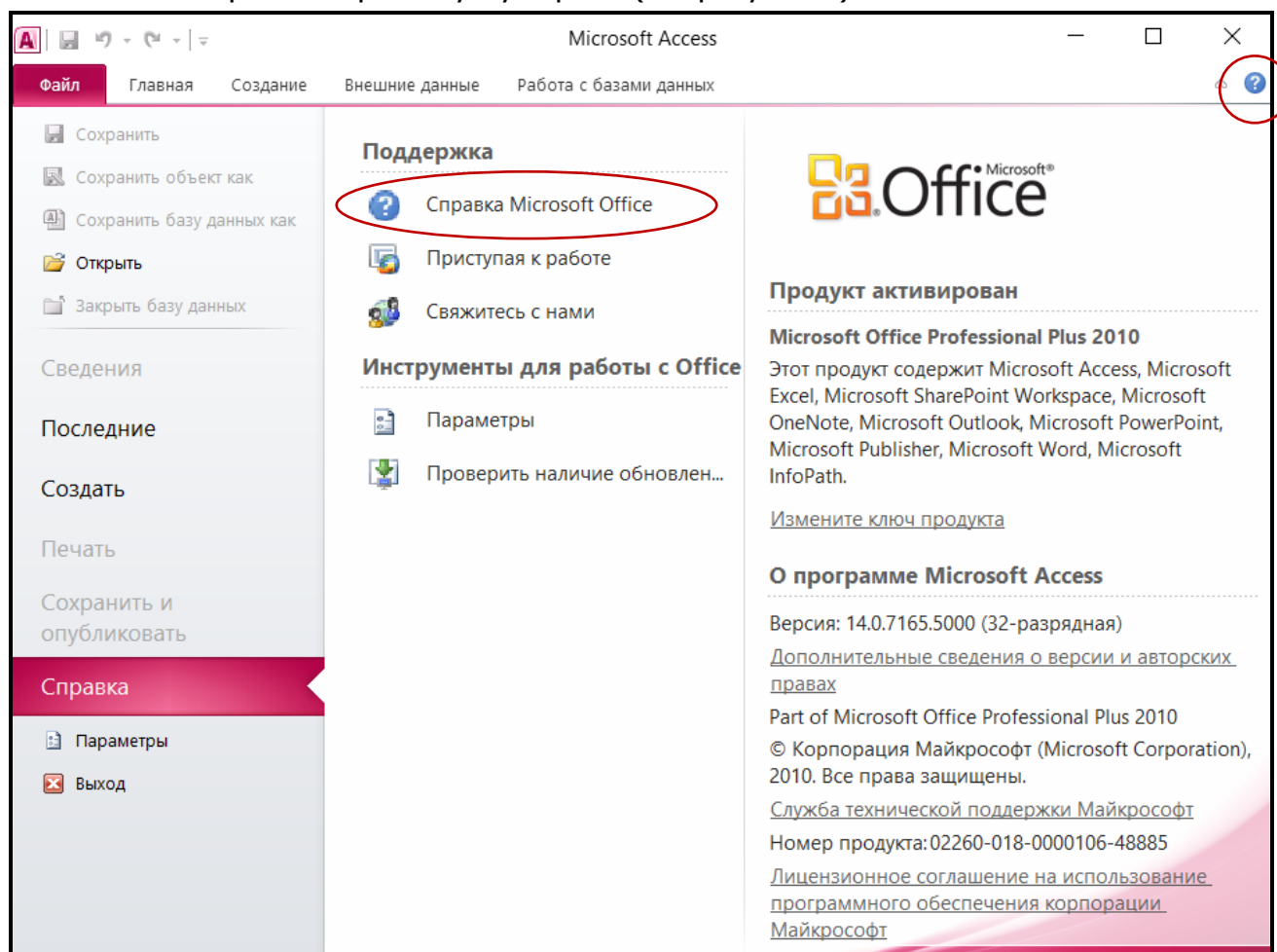


Рис. 2. Вызов справки Microsoft ACCESS

После щелчка по ссылке "*Справка Microsoft Office*" появится окно диалога "*Справка: Access*" (см. рисунок 3).

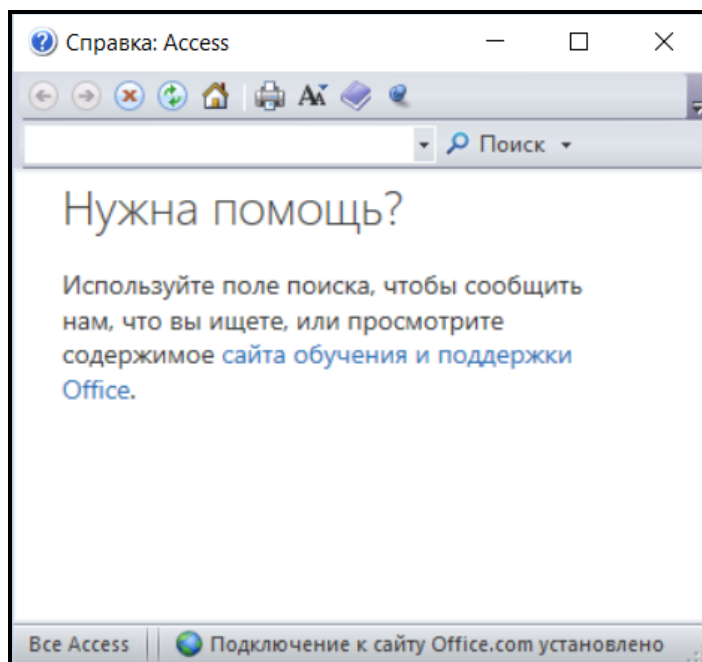



Рис. 3. Окно справочной системы MS ACCESS

В появившемся окне, как показано на рисунке 4 нажмите значок  Поиск и в раскрывшемся списке выберите пункт "*Access Справка*".

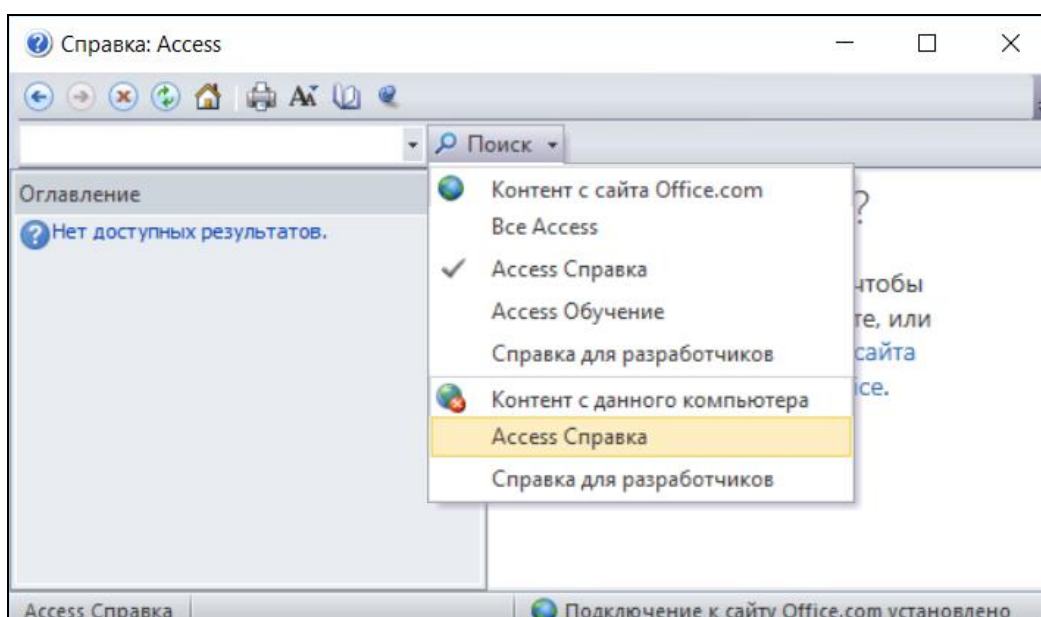


Рис. 4. Организация поиска Access Справки в Окне "Справка: Access"

2. В раскрывшемся списке разделов справки (см. рисунок 5) двойным щелчком откройте главу "Приступая к работе" и выберите параграф справки "*Начало работы в Microsoft ACCESS 2010*". И пользуясь указаниями справочной системы, ознакомьтесь со следующими темами:

- Начало работы с новой базой данных
- Новые возможности приложения Microsoft Access.
- Какой формат файла следует использовать в ACCESS 2010.

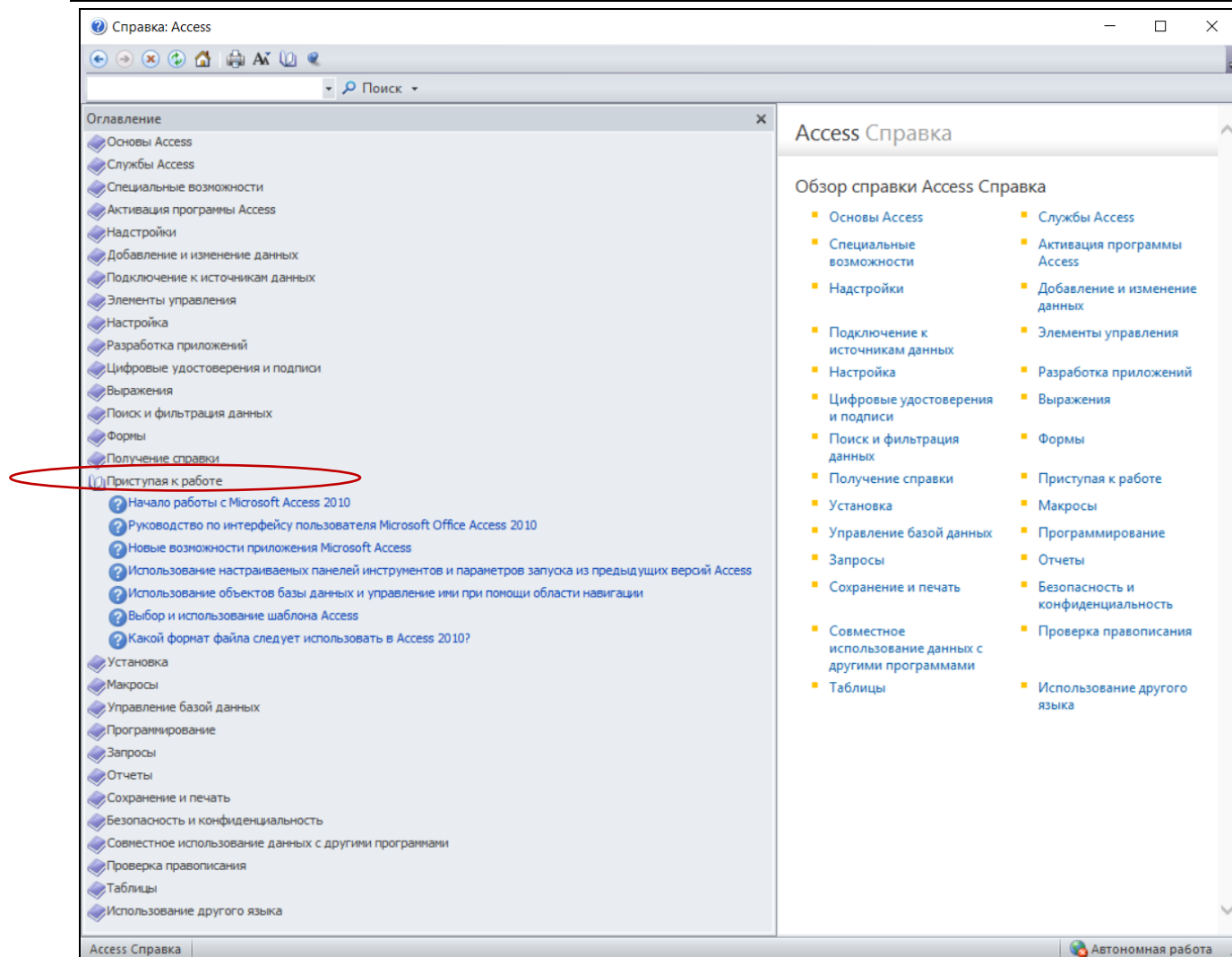


Рис. 5. *Окно справочной системы MS ACCESS*

3. Изучите также раздел "Основы Access". И пользуясь указаниями справочной системы, ознакомьтесь со следующими темами:

- Создание базы данных на компьютере.
- Основные сведения о создании баз данных.
 - Создание связей между таблицами.
 - Усовершенствование структуры.
 - Применение правил нормализации.
- Введение в таблицы.
 - Характеристики таблиц.
 - Создание таблицы.
 - Сохранение таблицы.
 - Определение первичного ключа таблицы.
 - Задание свойств таблицы.
 - Задание свойств поля.

4. Закройте окно "Справочной системы".

ПРИМЕЧАНИЕ: В качестве примера и демонстрации базы данных можно использовать имеющуюся базу данных "Библиотека"

IV. СОЗДАНИЕ БАЗЫ ДАННЫХ

Создание новой базы

Теперь, когда вы поняли основные принципы проектирования базы данных, пора приступить к созданию реальной базы данных. Продолжая, рассматривать пример базы данных Библиотека, мы можем создать новую базу данных и определить ее таблицы.

Для создания новой базы данных необходимо перейти на вкладку **ФАЙЛ** на панели инструментов и нажать на кнопку **СОЗДАТЬ**. В появившемся диалоговом окне выбрать пункт "Новая база данных".

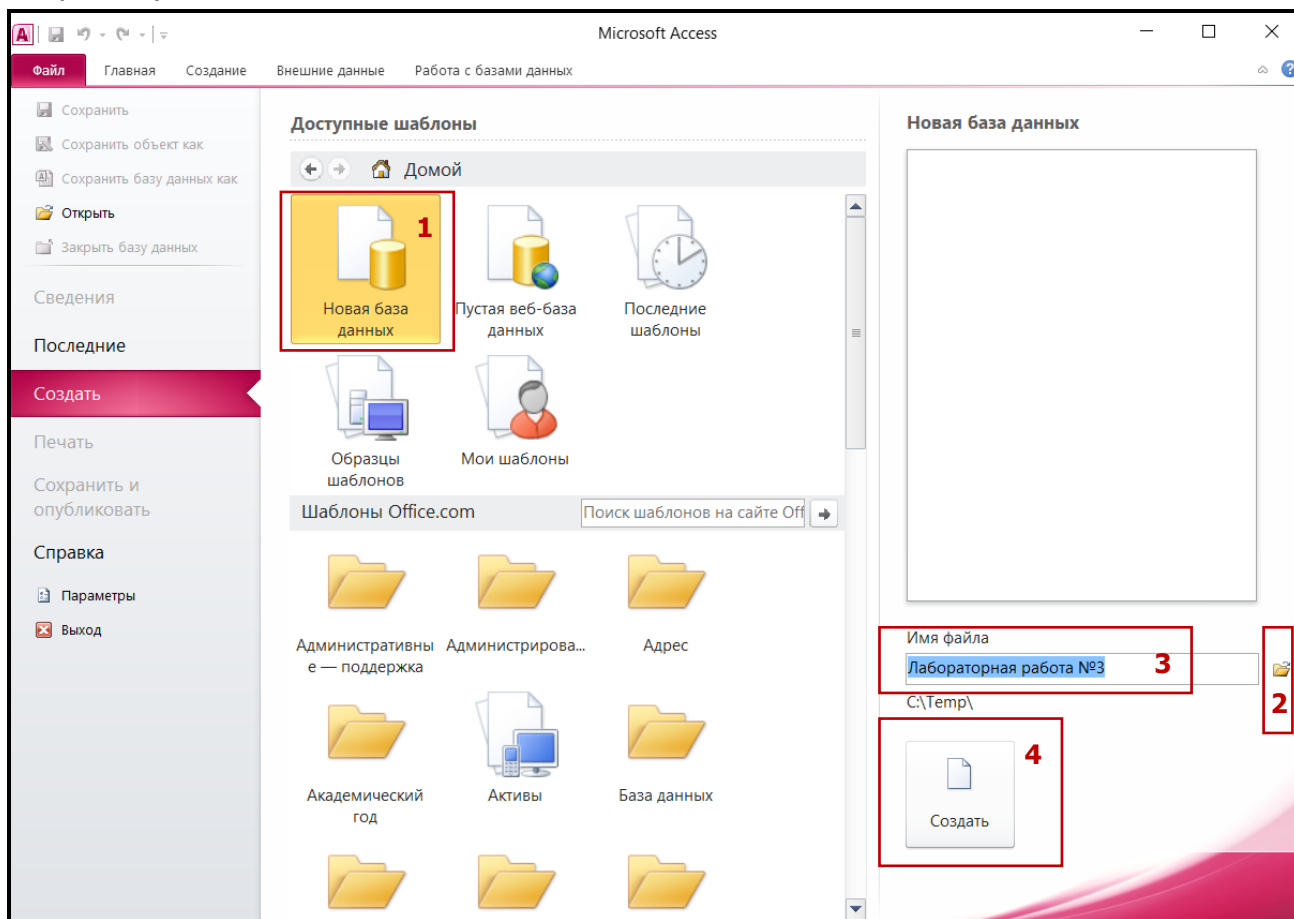


Рис. 6. Создание новой базы данных

Далее укажите нужный диск и каталог (например, **C:\Temp**), а в поле **ИМЯ ФАЙЛА** введите *Лабораторная работа №3* в качестве имени новой базы данных. ACCESS автоматически добавит к имени файла расширение **.accdb**. На экране появится окно базы данных, и т.к. вы создали новую базу данных, список таблиц базы будет пуст.

Создание таблиц

Создавать новую таблицу имеет смысл в том случае, если появился новый источник данных, не относящийся ни к одной из существующих таблиц.

Таблицу можно создать путем создания новой базы данных, вставки таблицы в существующую базу данных, а также создания ссылки на таблицу или ее импорта из другого источника данных, такого как книга Microsoft Office Excel, документ Microsoft Office Word, текстовый файл, веб-служба или другая база данных. При создании

пустой базы данных в нее автоматически вставляется новая пустая таблица. Затем можно приступить к вводу данных в таблицу, чтобы начать определение полей.

1. При создании новой базы данных в нее будет автоматически добавлена «Таблица1». Чтобы ее переименовать, необходимо щелкнуть в рабочей области правой клавишей мыши и в появившемся контекстном меню (см. рисунок 7) выбрать пункт **СОХРАНИТЬ**.

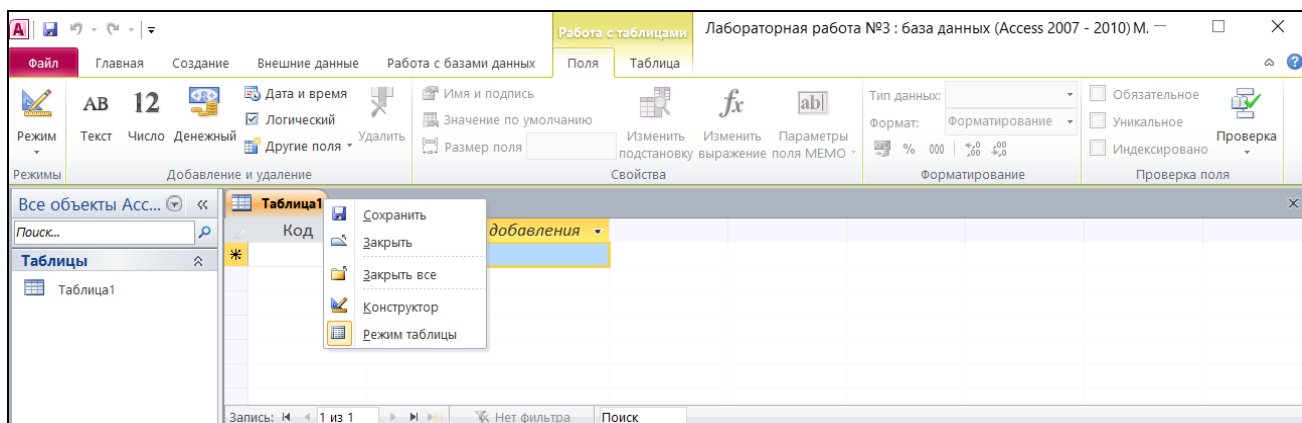


Рис. 7. Создание новой таблицы

2. Далее в появившемся диалоговом окне **СОХРАНЕНИЕ** (см. рисунок 8) ввести необходимое имя таблицы.

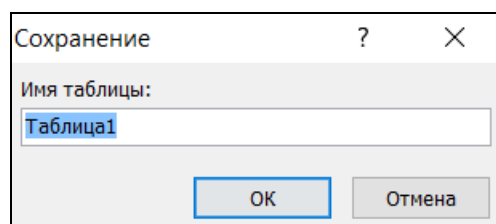


Рис. 8. Сохранение (переименование) таблицы

3. После того как таблица будет создана, ее можно открыть в режиме **КОНСТРУКТОРА** и, если необходимо, изменить макет таблицы.

Создание таблиц в режиме конструктора

Кроме того, существует еще один способ создания таблиц через пункт меню **СОЗДАНИЕ**.

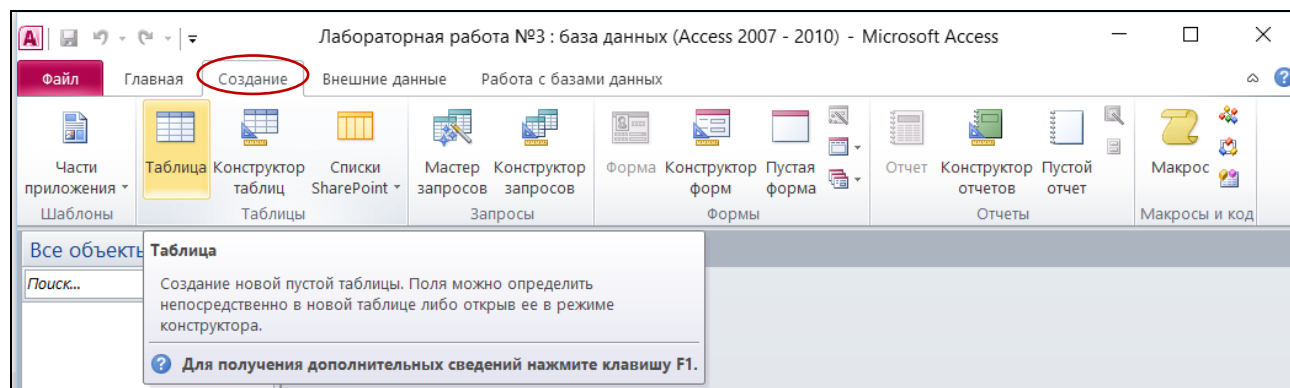


Рис. 9. Создание таблицы через пункт меню Создать

С помощью этого способа создадим три таблицы базы данных Библиотека: **"ИЗДАТЕЛЬСТВА", "КНИГИ", "ТЕМЫ"**.

Для создания таблицы **"ИЗДАТЕЛЬСТВА"** выполните следующие действия.

1. Нажмите пиктограмму **КОНСТРУКТОР ТАБЛИЦ**. При создании таблицы в режиме конструктора ACCESS представит окно пустой таблицы (см. рисунок 10).

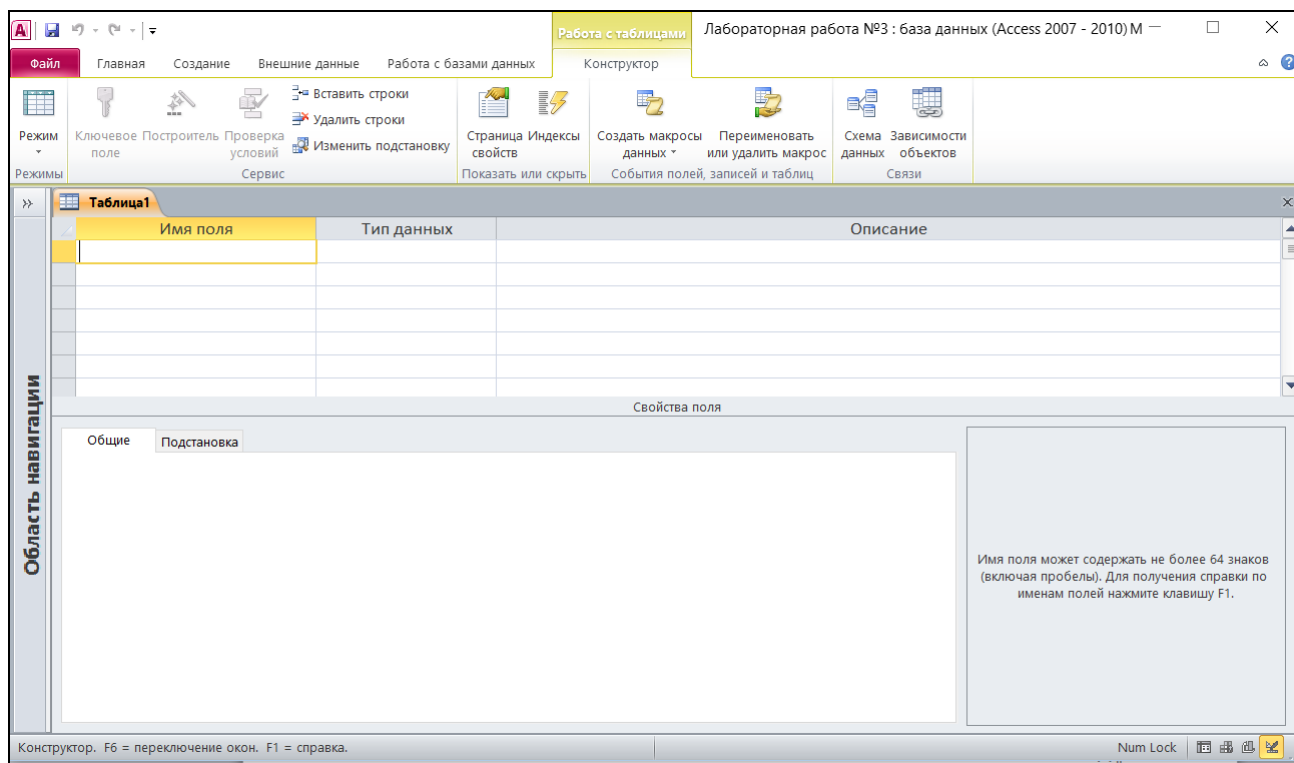


Рис. 10. Новая таблица в режиме Конструктора

В этом окне в строке заголовка указано имя таблицы, принятое ACCESS по умолчанию: **"Таблица1"**. В верхней части окна имеются три столбца для задания атрибутов создаваемой таблицы: **ИМЯ ПОЛЯ, ТИП ДАННЫХ, ОПИСАНИЕ**. Кроме того, каждое поле таблицы обладает дополнительными свойствами, отображаемыми в нижней части окна и определяющими условия ввода данных. В правом нижнем углу выводится краткая информация о полях или их свойствах. Эта информация меняется при перемещении курсора по окну таблицы.

Включение полей в таблицу начинается с ввода сведений о полях. В каждую строку необходимо ввести имя поля, выбрать для него тип данных, а также можно включить необязательное описание, которое будет отображаться в строке состояния при вводе данных в это поле. После выбора типа данных ACCESS выведет дополнительные свойства поля в нижней части окна. Набор свойств меняется в зависимости от выбранного типа данных, причем некоторые из этих свойств устанавливаются по умолчанию.

2. Приступим к созданию таблицы **"ИЗДАТЕЛЬСТВА"**. В первой строке в столбце **ИМЯ ПОЛЯ** наберите **"КОД ИЗДАТЕЛЬСТВА"**. При задании имен полей необходимо учитывать следующие ограничения:

- имена полей могут содержать *не более 64 символов*;
- в имени поля допускается использование русских и латинских букв, цифр, пробелов, а также специальных символов, *за исключением* точки (.), восклицательного знака (!) и прямых скобок ([]);

- имя поля *не может* начинаться с пробела;
- два поля в одной таблице *не могут иметь одинаковые имена*.

Для перехода к вводу значений в следующую ячейку строки (в данном случае ячейку столбца **ТИП ДАННЫХ**) можно воспользоваться мышью или клавишами **<ENTER>**, **<TAB>**, а также клавишами управления курсором.

3. После присвоения имени полю следует выбрать тип данных, для хранения которых предназначено это поле. По умолчанию для каждого поля автоматически выбирается текстовый тип данных. Для изменения типа поля следует выбрать нужный тип данных из списка в столбце **ТИП ДАННЫХ** нажатием кнопки вызова списка. Допустимые в ACCESS типы данных, дополнительные свойства полей, их назначение и размеры приведены в таблицах 4 и 5.

Таблица 4. *Доступные типы данных MS ACCESS 2010*

ТИП ДАННЫХ	НАЗНАЧЕНИЕ	РАЗМЕР
1	2	3
Текстовый	Алфавитно-цифровые знаки. Используется для текста или чисел, не применяемых в расчетах (например, кода товара). Числовые значения, которые хранятся как текст, проще сортировать и фильтровать, но их сложно использовать в вычислениях.	До 255 байт (каждый символ занимает 1 байт)
Поле МЕМО	Буквы и цифры (более 255 знаков) или форматированный текст. Используется для текста длиннее 255 знаков или форматированного текста. Типичные примеры использования поля Мемо – примечания, длинные описания и абзацы с форматированием, например с полужирным начертанием или курсивом.	До 1 гигабайта знаков или 2 гигабайт памяти (2 байта на знак), из которых в элементе управления можно отобразить 65 535 знаков
Числовой	Числовые значения (целые или дробные). Используется для хранения числовых данных, применяемых в вычислениях, за исключением денежных значений (для них используется тип данных "Денежный").	1, 2, 4 и 8 байт или 16 байт (если используется для кода репликации)
Дата/время	Даты и время. Используется для хранения значений даты и времени. Обратите внимание, что каждое сохраняемое значение содержит одновременно и компонент даты, и компонент времени.	8 байт
Денежный	Денежные значения. Используется для хранения денежных значений (валюты).	8 байт
Счетчик	Уникальное числовое значение, автоматически вставляемое при добавлении записи. Используется для создания уникальных значений, которые могут применяться в качестве первичного ключа. Обратите внимание, что значения для полей "Счетчик" могут генерироваться путем добавления единицы, добавления заданного значения или на основе случайных чисел.	4 байта или 16 байт, если используется для кода репликации

Продолжение таблицы 4

1	2	3
Логический	Логические значения (да/нет). Вы можете использовать любой из трех форматов: "Да/Нет", "Истина/Ложь" или "Вкл/Выкл".	1 бит (8 бит = 1 байт).
Поле объекта OLE	OLE-объекты или другие двоичные данные. Используется для хранения объектов OLE других программ Microsoft Windows.	До 1 Гб
Вложение	Рисунки, изображения, двоичные файлы, файлы Microsoft Office. Стандартный тип данных для сохранения цифровых изображений и любого типа двоичных файлов.	Для сжатых вложений – 2 гигабайта. Для несжатых вложений – примерно 700 Кбайт (в зависимости от возможной степени сжатия).
Гиперссылка	Гиперссылки. Используется для хранения гиперссылок вызова веб-страниц одним щелчком с помощью URL-адреса или файлов с помощью формата универсального имени UNC. Кроме того, можно использовать ссылку на объекты Access, хранящиеся в базе данных.	До 1 гигабайта знаков или 2 гигабайт памяти (2 байта на знак), из которых в элементе управления можно отобразить 65 535 знаков.
Мастер подстановок	<i>Не является типом данных, а вызывает мастер подстановок.</i> <i>Используется для запуска мастера подстановок, с помощью которого можно создать поле, позволяющее выбрать значение из другой таблицы, запроса или списка значений, используя поле со списком.</i>	<i>На основе таблицы или запроса — размер привязанного столбца.</i> <i>На основе значения — размер текстового поля, содержащего значение.</i>

Таблица 5. Дополнительные свойства полей

НАЗВАНИЕ	НАЗНАЧЕНИЕ
1	2
Размер поля	<p>Определяется максимальный размер данных, для хранения которых предназначено данное поле. Поле с текстовым типом данных может иметь размер от 1 до 255 байт (символов), для числового типа данных размер поля может быть следующим:</p> <p><u>байт</u>: целые числа от 0 до 255. Занимает 1 байт;</p> <p><u>целое</u>: числа от -32 768 до +32 767. Занимает 2 байта;</p> <p><u>длинное целое</u>: числа от -2 147 483 648 до +2 147 483 647. Занимает 4 байта;</p> <p><u>с плавающей точкой</u>: хранит числа с точностью до 6 знаков от $-3,4 \cdot 10^8$ до $+3,4 \cdot 10^8$. Занимает 4 байта;</p> <p><u>с плавающей точкой</u>: хранит числа с точностью до 10 знаков от $-1,797 \cdot 10^{308}$ до $+1,797 \cdot 10^{308}$. Занимает 8 байт.</p>

Продолжение таблицы 5

1	2
Формат поля	Определяется формат представления данных при выводе на экран или печать.
Число десятичных знаков	Определяется число знаков, выводимых после десятичного разделителя.
Маска ввода	Определяется образец для всех данных, которые вводятся в это поле. Позволяет контролировать правильность данных при вводе.
Подпись	Определяется обозначение, которое будет использоваться в качестве заголовка столбца в режиме таблицы и должно быть распечатано в форме или отчете. При отсутствии данного значения за основу автоматически будет взято имя поля.
Значение по умолчанию	Определяется значение, которое автоматически вводится в поле при формировании новой записи данных.
Условие на значение	Определяется выражение, проверка которого выполняется при вводе или изменении данных в поле. Позволяет контролировать правильность данных при вводе.
Сообщение об ошибке	Определяется сообщение об ошибке, которое будет выводиться на экран, когда вводимое значение не удовлетворяет условию на значение.
Обязательное поле	Определяется, требует ли поле обязательного ввода значения.
Пустые строки	Определяется разрешение на ввод пустых строк, т.е. строк, не содержащих символов. Пустая строка записывается как " " (два символа прямых кавычек без пробела).
Индексированное поле	Определяется, нужно ли индексировать поле для ускорения операции поиска и сортировки. Ключевое поле таблицы индексируется автоматически.

Теперь вы знаете достаточно о типах данных и свойствах полей, для того чтобы закончить создание таблицы **"ИЗДАТЕЛЬСТВА"**. Вся необходимая информация для этого приведена в таблице 6.

Таблица 6. Определение полей таблицы "ИЗДАТЕЛЬСТВА"

ИМЯ ПОЛЯ	ТИП ДАННЫХ	ОПИСАНИЕ	СВОЙСТВА ПОЛЯ
Код издательства	Счетчик	Код издательства	—
Наименование издательства	Текстовый	Название издательства	Размер поля 30
Город	Текстовый	Город	Размер поля 20

Каждая таблица в реляционной базе данных должна иметь *первичный ключ*. В таблице "**ИЗДАТЕЛЬСТВА**" разумно в качестве такого ключа выбрать поле **КОД ИЗДАТЕЛЬСТВА**.

Для определения первичного ключа необходимо:

- выделить строку с описанием поля **КОД ИЗДАТЕЛЬСТВА**, установив курсор мыши в области маркировки поля слева от этого поля и нажав кнопку мыши;
- выполнить команду **ОПРЕДЕЛИТЬ КЛЮЧ** меню **ПРАВКА** или нажать

кнопку **ОПРЕДЕЛИТЬ КЛЮЧ**  на панели инструментов.

4. Последнее действие, которое необходимо произвести с новой таблицей — это сохранить ее. Для этого выполните команду **СОХРАНИТЬ** меню **ФАЙЛ**, в диалоговом окне **СОХРАНЕНИЕ** введите имя таблицы "**ИЗДАТЕЛЬСТВА**" и нажмите кнопку **ОК**. Для завершения работы с таблицей выполните команду **ЗАКРЫТЬ** меню **ФАЙЛ**.

Модификация структуры таблицы

Вполне вероятно, что при создании таблицы вы могли допустить ошибки. Действия по модификации структуры таблицы приведены в таблице 7. Для их выполнения таблица должна быть открыта в режиме конструктора.

Таблица 7. Модификация структуры таблицы

ДЕЙСТВИЕ	РЕАЛИЗАЦИЯ
Вставка новых полей в таблицу	выделить строку определения поля, перед которым нужно вставить новое поле: <Insert>
Удаление полей из таблицы	выделить строки определения полей, которые нужно удалить:
Восстановление поля	Отменить удаление нажав кнопку  или Закрыть окно КОНСТРУКТОРА , и указать в запросе о сохранении изменений — НЕТ
Изменение порядка следования полей	выделить строку определения поля, которое нужно переместить; мышью переместить строку так, чтобы она оказалась над полем, перед которым ее необходимо расположить
Удаление первичного ключа (без удаления самого поля)	Повторно нажать на кнопку КЛЮЧЕВОЕ ПОЛЕ  .

Задание

Самостоятельно создайте таблицы **"КНИГИ"** и **"ТЕМЫ"** Вся необходимая информация по определению полей таблиц приведена в таблице 8.

Таблица 8. Определение полей таблиц **"КНИГИ"** и **"ТЕМЫ"**

ТАБЛИЦА	ИМЯ ПОЛЯ	ТИП ДАННЫХ	ОПИСАНИЕ	СВОЙСТВА ПОЛЯ	ПЕРВИЧНЫЙ КЛЮЧ
Книги	Код книги	Счетчик	Код книги	—	Да
Книги	Название книги	Текстовый	Название книги	Размер поля — 100	—
Книги	Автор	Текстовый	Фамилия автора	Размер поля — 50	—
Книги	Код издательства	Числовой	Код издательства	Размер поля — длинное целое,	—
Книги	Объем	Числовой	Количество страниц	Размер поля — целое	—
Книги	Год издания	Числовой	Год издания	Размер поля —	—
Книги	Стоимость	Денежный	Стоимость книги	Формат поля — денежный	—
Книги	Код темы	Числовой	Код темы	Размер поля — длинное целое,	—
Темы	Код темы	Счетчик	Код темы	—	Да
Темы	Название Тема	Текстовый	Описание предметной области	Размер поля — 100	—

V. УСТАНОВКА СВЯЗЕЙ МЕЖДУ ТАБЛИЦАМИ

Типы связей создаваемые Microsoft Access

Тип отношения в создаваемой Microsoft Access связи зависит от способа определения связываемых полей.


Отношение «один-ко-многим». Отношение «один-ко-многим» является наиболее часто используемым типом связи между таблицами. В отношении «один-ко-многим» каждой записи в таблице А могут соответствовать несколько записей в таблице В, но запись в таблице В не может иметь более одной соответствующей ей записи в таблице А.

Отношение «один-к-одному». При отношении «один-к-одному» запись в таблице А может иметь не более одной связанной записи в таблице В и наоборот. Отношения этого типа используются не очень часто, поскольку большая часть сведений, связанных таким образом, может быть помещена в одну таблицу. Отношение «один-к-одному» может использоваться для разделения очень широких таблиц, для отделения части таблицы по соображениям защиты, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице.

Отношение «многие-ко-многим». При отношении «многие-ко-многим» одной записи в таблице А могут соответствовать несколько записей в таблице В, а одной записи в таблице В несколько записей в таблице А. Этот тип связи возможен только с помощью третьей (связующей) таблицы, первичный ключ которой состоит из двух полей, которые являются внешними ключами таблиц А и В. Отношение «многие-ко-многим» по сути дела представляет собой два отношения «один-ко-многим» с третьей таблицей.

После определения таблиц базы данных необходимо указать MS ACCESS, как эти таблицы связаны друг с другом. Позднее ACCESS будет использовать эти связи, в запросах, формах и отчетах.

Напомним, что связь между таблицами может быть установлена при наличии в этих таблицах полей, которые содержат совпадающие данные. Именно с помощью сопоставления записей различных таблиц с одинаковыми значениями связующих полей и осуществляется такая связь. Эти поля не обязательно должны иметь одинаковые имена, но необходимо, чтобы *совпадали типы данных и размеры связующих полей*. Именно по этой причине при создании таблицы **"КНИГИ"** для поля **КОД ИЗДАТЕЛЬСТВА**, через которое будет определяться связь между таблицами **"ИЗДАТЕЛЬСТВА"** и **"КНИГИ"**, определен числовой тип данных размером длинное целое, т.к. поле **КОД ИЗДАТЕЛЬСТВА** таблицы **"ИЗДАТЕЛЬСТВА"** мы определили как счетчик (см. ранее).

Для определения связей между таблицами из окна базы данных выполните команду **Схема данных** меню **Сервис** или нажмите кнопку **Схема данных**  на панели инструментов. Т.к. связи в этой базе данных определяются впервые, ACCESS откроет пустое окно **Схема данных**, а затем — диалоговое окно **Добавление таблицы**, приведенное на рисунке 11.

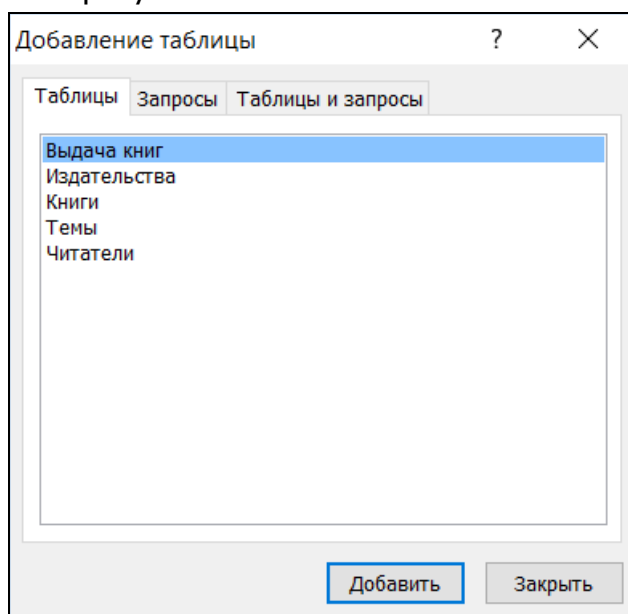


Рис. 11. Диалоговое окно *Добавление таблицы*

Далее необходимо выбрать таблицы, между которыми вы хотите сформировать связь. Для этого нужно выделить таблицу и нажать кнопку **Добавить**. Для выделения нескольких таблиц следует нажать клавишу **<Ctrl>** и, не отпуская ее, выделить по очереди все нужные объекты. Для выделения всех объектов или непрерывной группы объектов следует выделить первый объект, затем нажать клавишу **<Shift>** и, не

отпуская ее, указать на последний объект. Добавьте таким образом таблицы **"ИЗДАТЕЛЬСТВА"**, **"КНИГИ"**, **"ТЕМЫ"** и закройте окно Добавление таблицы, щелкнув мышью по кнопке **Заккрыть**. После этого в окне Схема данных появятся три таблицы, причем ключевые поля этих таблиц будут выделены полужирным шрифтом.

Для создания связи между таблицами необходимо с помощью мыши перенести связующее поле из списка полей одной таблицы на соответствующее поле в списке полей другой таблицы. Если одно из полей является ключевым, то переносить следует его. Поле, которое переносится, принадлежит *главной* таблице, вторая таблица называется *подчиненной*. Часто ACCESS сам может принять решение о том, какую из таблиц считать главной (например, если одно из связываемых полей является ключевым или просто имеет уникальный индекс, главной будет таблица, содержащая это поле). Если это не удастся, то главной считается таблица, с которой было начато прокладывание связи. Одна и та же таблица может выступать в одной связи как главная, а в другой — как подчиненная. Статус таблицы влияет на действия, предпринимаемые ACCESS при редактировании и обновлении записей из связанных таблиц.

Установим, таким образом, связь между таблицами **"ИЗДАТЕЛЬСТВА"** и **"КНИГИ"**. Выделите поле **КОД ИЗДАТЕЛЬСТВА** таблицы **"ИЗДАТЕЛЬСТВА"**, нажмите левую кнопку мыши и, удерживая ее в нажатом положении, переместите его на поле **КОД ИЗДАТЕЛЬСТВА** таблицы **"КНИГИ"**. На экране появится диалоговое окно **Связи** (рисунок 9). В этом окне следует проверить правильность имен полей в обоих столбцах и исправить их, если это необходимо.

Чтобы в базе данных не хранились книги несуществующих издательств (издательств, не зарегистрированных в таблице **"ИЗДАТЕЛЬСТВА"**), установите флажок **Обеспечение целостности данных**. После этого ACCESS не позволит добавить в таблицу **"КНИГИ"** запись с неправильным кодом издательства.

Ниже расположена группа переключателей **Отношение**, с помощью которых можно определить тип связи: *один-к-одному* или *один-ко-многим*. В нашем случае — связь типа *один-ко-многим*.

Обратите внимание, что после того как была установлена опция **ОБЕСПЕЧЕНИЕ ЦЕЛОСТНОСТИ ДАННЫХ**, стали доступными еще две опции: **КАСКАДНОЕ ОБНОВЛЕНИЕ СВЯЗАННЫХ ПОЛЕЙ** и **КАСКАДНОЕ УДАЛЕНИЕ СВЯЗАННЫХ ЗАПИСЕЙ**.

Свойства связи

ОБЕСПЕЧЕНИЕ ЦЕЛОСТНОСТИ ДАННЫХ. Включение этого свойства запрещает помещать в поле связанной таблицы значения, которые отсутствуют в поле главной таблицы. При этом поле главной таблицы должно являться **ключевым** или, по крайней мере, **уникальным**. Т.е., значения в поле связанной таблицы будут однозначно указывать на конкретные записи в главной таблице.

КАСКАДНОЕ ОБНОВЛЕНИЕ СВЯЗАННЫХ ПОЛЕЙ. Если задано это свойство, при изменении значения поля в главной таблице будут автоматически меняться соответствующие значения в связанной таблице.

КАСКАДНОЕ УДАЛЕНИЕ СВЯЗАННЫХ ЗАПИСЕЙ. Если удаляется запись из главной таблицы, должны автоматически удалиться связанные с ней записи из связанной таблицы.

Каскадное обновление и удаление могут быть заданы только в том случае, если включено обеспечение целостности данных. Если обеспечение целостности не задано, построенная связь никак не будет ограничивать множества допустимых значений

связанных полей. В MS Access такие связи играют роль только при построении запросов.

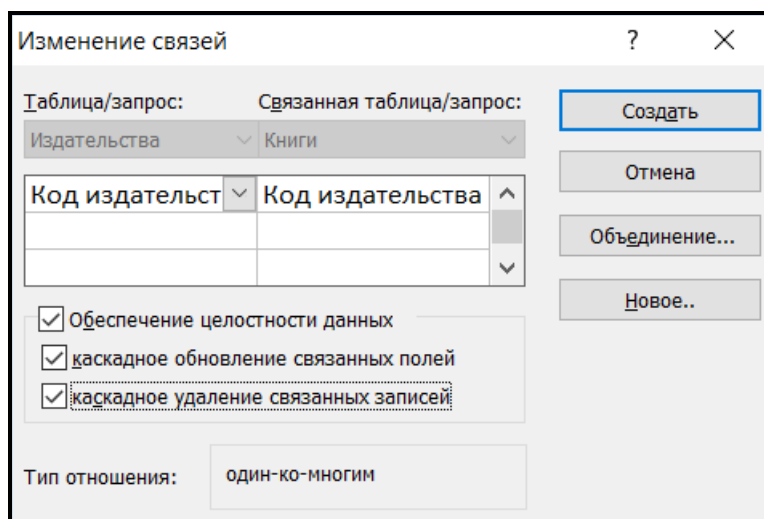


Рис. 12. Окно *Изменение связей*

Первая опция означает, что при изменении значения связующего поля в главной таблице соответствующие поля подчиненной таблицы автоматически будут обновлены. Вторая опция означает, что в случае удаления записи из главной таблицы все связанные с ней записи из подчиненной таблицы будут удалены автоматически. Установите оба этих флажка.

После определения всех параметров связи нажмите кнопку **Создать**. ACCESS установит указанную связь и проведет линию между двумя таблицами, на которой будет указан тип отношения. Эта связь останется в базе данных до тех пор, пока не будет удалена.

Аналогичным образом установите связь между таблицами **"КНИГИ"** и **"ТЕМЫ"**. После этого окно **Схема данных** будет выглядеть так, как показано на рисунке 13.

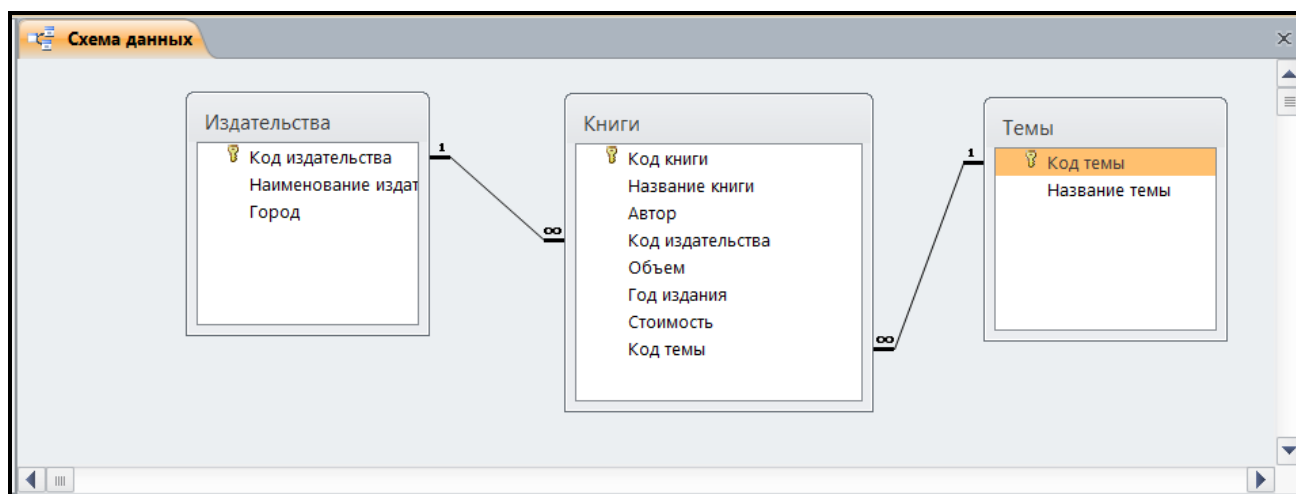


Рис. 13. Связи между таблицами базы данных Библиотека

В заключение, сохраните произведенные вами действия, выполнив команду **Сохранить макет** меню **Файл**, и закройте окно **Схема данных**, выбрав команду **Заккрыть** меню **Файл**.

ACCESS предоставляет возможность просмотреть связи, установленные в базе данных, изменить или вообще удалить ставшие ненужными связи. Действия по корректировке связей между таблицами базы данных приведены в таблице 9.

Таблица 9. *Корректировка связей между таблицами*

ДЕЙСТВИЕ	РЕАЛИЗАЦИЯ
Просмотр связей текущей базы данных	Работа с базами данных/Схема данных
Отображение в окне Схема данных всех связей, определенных в текущей базе данных	Работа со связями/Показать все
Отображение в окне Схема данных только связей, определенных для конкретной таблицы	выделить таблицу; – Работа со связями /Показать прямые
Изменение параметров связи	двойной щелчок мышью по линии связи
Удаление связи	выделить линию связи; –
Удаление таблицы из окна Схема данных	выделить таблицу; –
Добавление новых таблиц в окно Схема данных	Работа со связями/Отобразить таблицу

Задание

1. Дополните базу данных Библиотека таблицами "**ЧИТАТЕЛИ**" и "**ВЫДАЧА КНИГ**". Вся необходимая информация по определению полей таблиц приведена в таблице 10 и таблице 11.

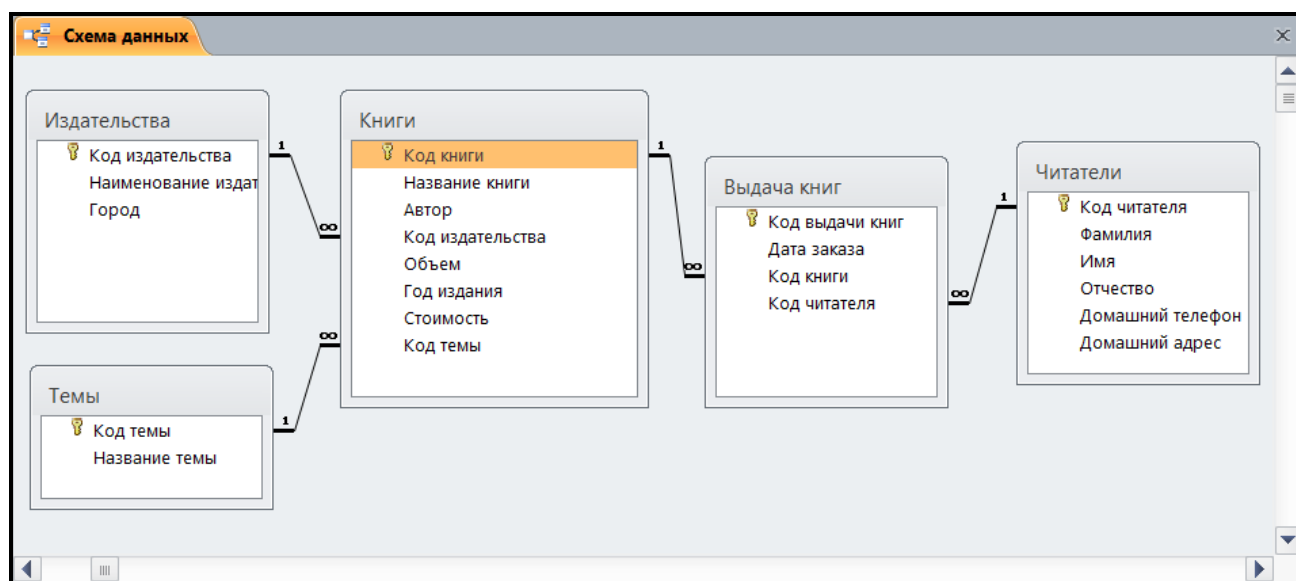
Таблица 10. *Определение полей таблицы "ЧИТАТЕЛИ"*

ТАБЛИЦА	ИМЯ ПОЛЯ	ТИП ДАННЫХ	ОПИСАНИЕ	СВОЙСТВА ПОЛЯ	ПЕРВИЧНЫЙ КЛЮЧ
Читатели	Код читателя	Счетчик	Код читателя	—	Да
Читатели	Фамилия	Текстовый	Фамилия читателя	Размер поля — 50	—
Читатели	Имя	Текстовый	Имя читателя	Размер поля — 50	—
Читатели	Отчество	Текстовый	Отчество читателя	Размер поля — 30	—
Читатели	Домашний телефон	Текстовый	Домашний телефон читателя	Размер поля — 15, пустые строки – да	—
Читатели	Домашний адрес	Текстовый	Домашний адрес читателя	Размер поля — 100	—

Таблица 11. *Определение полей таблицы "ВЫДАЧА КНИГ"*

Выдача книг	Код дата заказа	Счетчик	Код даты заказа	—	Да
Выдача книг	Код читателя	Числовой	Код читателя	Размер поля — длинное целое, индексированное поле — да (допускаются совпадения)	—
Выдача книг	Код книги	Числовой	Код книги	Размер поля — длинное целое, индексированное поле — да (допускаются совпадения)	—
Выдача книг	Дата заказа	Дата/время	Дата заказа книги	Формат поля — краткий формат даты	—

- Для полей **ДОМАШНИЙ ТЕЛЕФОН** таблицы "ЧИТАТЕЛИ" и **ДАТА ЗАКАЗА** таблицы "ВЫДАЧА КНИГ" создайте маски ввода с помощью конструктора (для вызова конструктора воспользуйтесь кнопкой с тремя точками, расположенной справа от свойства **Маска ввода**).
- Установите между полями таблиц связи, приведенные на рисунке 14.

**Рис. 14.** *Связи между таблицами расширенной базы данных Библиотека*

Продемонстрируйте результаты вашей работы преподавателю!

Контрольные вопросы:

1. Дайте определение базы данных.
2. Основные объекты Microsoft ACCESS.
3. Понятие поля и записи в Microsoft ACCESS.
4. Режимы работы с БД в Microsoft ACCESS.
5. Перечислите основные этапы проектирования базы данных.
6. Какие требования предъявляются к базе данных?
7. Сформулируйте правила при проектировании базы данных.
8. Какие ограничения учитываются при создании базы данных в СУБД ACCESS?
9. Какие типы данных используются в СУБД ACCESS?
10. Что такое «маска ввода»?
11. Назовите известные вам виды связей.
12. Перечислите основные свойства полей ACCESS.
13. Что означают в СУБД ACCESS каскадное обновление связанных полей и каскадное удаление связанных записей?