

# **ЛАБОРАТОРНАЯ РАБОТА № 1 ПЕРВОЙ НЕДЕЛИ КУРСА «БАЗЫ ДАННЫХ» НАЧАЛО РАБОТЫ С MS ACCESS**

В данной лабораторной работе мы начнем знакомство с СУБД Microsoft Access, рассмотрим создание базы данных, добавление таблиц в базу, определение связей между таблицами.

Access входит в пакет офисных приложений Microsoft Office и является одной из самых популярных настольных СУБД для компьютеров, работающих под управлением операционной системы Windows. Некоторые особенности, сделавшие этот продукт столь широко используемым:

- Access – реляционная СУБД;
- Access тесно интегрирована с другими приложениями Microsoft Office;
- Access хранит всю базу данных в одном файле (если специально не делать разделение базы данных или не использовать внешние данные), следовательно, облегчен перенос БД с одного компьютера на другой.

## **Создание новой базы данных**

Для создания новой базы, запустите Access и выберите в меню Файл → Создать → Новая база данных. В правой части окна укажите имя файла БД и его расположение, нажмите кнопку «Создать».

*Задание. Создайте новую базу данных.*

## **Объекты базы данных Access**

В одной базе может быть множество объектов разных типов – таблиц, запросов, форм, отчетов. Таблицы – основной объект базы Access, так как именно в них хранятся пользовательские данные. Кроме того, при создании новой БД Access создает системные таблицы, где содержится информация об объектах базы.

В Access существует два основных режима работы с таблицами – режим Таблицы и режим Конструктора. В первом из них осуществляется непосредственно работа с данными – ввод, редактирование, просмотр, сортировка... В режиме Конструктора создается и модифицируется структура таблицы: задаются имена и типы полей, формируются накладываемые на них ограничения и т. д.

В новой базе данных автоматически создается и открывается в режиме таблицы Таблица1 (в русскоязычной версии Access). Для удобства в

Access реализована возможность создания новых полей и в режиме Таблицы, основываясь на введенных пользователем значениях.

*Задание. Откройте Таблицу1 в режиме таблицы и создайте три новых поля, введя в первой строке таблицы сначала текстовое значение, затем число, а после дату в формате «дд.мм.гггг» (день, месяц, год с точкой в качестве разделителя). Обратите внимание, что автоматически первое поле/столбец уже имеет тип – счетчик. Ячейка, находящаяся в этом столбце, заполняется самостоятельно и не требует ввода пользователя. Следовательно, вводить новые данные следует в первой строке в пустом столбце (в последующем после счетчика).*

*Заполненные поля/столбцы автоматически получают имена: «Поле1, Поле2 и тд».*

*Перейдите в режим Конструктора, выбрав в главном меню соответствующую иконку (рис.А.1.1). Удостоверьтесь, что система верно распознала вводимые данные и назначила типы полям/столбцам. Удалите Таблицу1, для дальнейшего выполнения лабораторной она не потребуется.*

*Перед удалением Таблицы1 ее следует закрыть. Для этого нажмите крестик в верхней правой части таблицы. Как только закроется окно, вызовите консольное меню Таблицы1 и выберите «Удалить».*

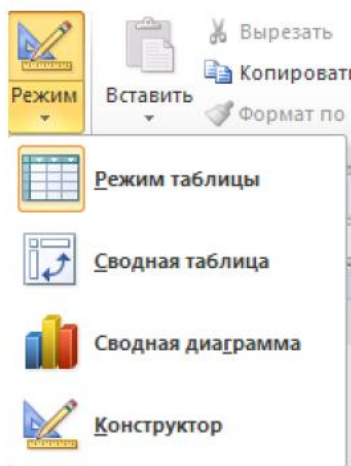


Рис.А.1.1 Переход в режим Конструктора

В дальнейшем мы будем работать по правилам: если предстоит работа со структурой таблиц – то стоит обратиться к режиму Конструктора, а работа с данными – в режиме Таблицы.

Как уже отмечалось, в режиме Конструктора таблицы создаются путем задания имен полей, их типов и свойств. При задании имен полей нужно учитывать, что:

- имена полей должны быть уникальны в пределах таблицы;

- имена полей могут содержать не более 64 символов, включая пробелы;
- нежелательно, чтобы имена полей совпадали с именами встроенных функций или свойств Access, операторов языка SQL, так как в некоторых случаях это может привести к неправильной интерпретации запросов к БД;
- имена полей не должны начинаться с пробела или управляющего символа;
- имена полей могут содержать любые буквы, цифры, знаки за исключением точки ".", восклицательного знака "!", апострофа " ' ", квадратных скобок " [ ", " ] ".

Аналогичные правила используются при задании имен таблиц и других объектов Access.

Познакомившись с правилами именования полей таблиц, перейдем к рассмотрению типов данных. Ниже приведено перечисление основных типов с названиями.

**Текстовый** (Text) – символьные строки переменной длины до 255 символов. Это тип данных «по умолчанию». В свойстве «Размер поля» можно задать максимальную длину строки. Access не заполняет пробелами неиспользованную часть текстового поля, что снижает общий размер базы данных. С помощью этого типа данных иногда хранят и числовые значения, с которыми не требуется производить вычисления (например, номера телефонов).

**Поле МЕМО** (Memo) – это поле для ввода текстовой информации большого объема. Оно может содержать до 63 999 символов. В отличие от поля типа Текстовый, в данном случае в таблице находятся не сами данные, а только ссылки на блоки данных, которые хранятся отдельно (но в том же файле). Поле МЕМО не может быть ключевым или проиндексированным. Для символьных данных небольшого объема более экономно использовать тип Текстовый.

**Числовой** (Number) – тип, используемый для хранения числовых данных. Имеет несколько подтипов, определяющих, целочисленные или дробные данные будут храниться, а также размер элемента. Выбор осуществляется путем редактирования свойства «Размер данных». По умолчанию используется тип **Длинное целое** (Long Integer), занимающий 4 байта и принимающий целочисленные значения из диапазона от -2 147 483 648 до 2 147 483 647. Другие варианты: **Байт** (Byte), **Целое** (Integer), **Действительное** (Decimal), **Одинарное с плавающей точкой** (Single),

**Двойное с плавающей точкой** (Double), **Код репликации** (Replication ID) – 16-ти байтное значение, также называемое Globally unique identifier (GUID). Размер и диапазоны значений для каждого из указанных числовых типов всегда можно уточнить в справочной системе Access (вызывается по нажатию клавиши F1).

**Дата/Время** (Date/Time) – специальный тип данных для представления даты и времени. Занимает 8 байт. Это число, целая часть которого задает дату, а дробная – время. Могут быть представлены даты, начиная с 100 по 9999 годы.

**Денежный** (Currency) – специальный тип, предназначенный для хранения чисел с точностью от 1 до 4 знаков после запятой: арифметические операции с числами, обозначающими денежные суммы, принято проводить с точностью до 4-го знака после запятой. Целая часть значения данного типа может содержать до 15 десятичных цифр.

**Счетчик** (AutoNumber) – тип данных, используемый, если требуется получить уникальное в рамках таблицы числовое значение. Подобные поля часто используются для задания ключа таблицы. По умолчанию, счетчик основывается на типе данных **Длинное целое**, и последовательно выдает 4-х байтные значения, автоматически увеличивая на 1 предыдущее. В свойствах поля типа Счетчик можно указать, чтобы значения брались не последовательно, а в случайном порядке. Значения поля типа Счетчик вводить вручную и обновлять нельзя – значения автоматически генерируются для каждой новой строки. Другая важная особенность этого типа – генерируемые числа не повторяются. Если, например, вы внесли в таблицу строку со значением последовательно увеличивающегося счетчика равным 1, а потом ее удалили, следующая строка все равно получит значение 2 (хотя теперь в таблице она будет первая).

**Логический** (Yes/No) – логический тип, который может содержать только два значения. В зависимости от настроек в свойстве поля, называемом «Формат ввода» они могут отображаться как «Истина/Ложь», «Да/Нет» или «Вкл/Выкл». Поля данного типа не могут быть ключевыми, но их можно индексировать.

**Поле объекта OLE** (OLE object) – содержит ссылку на OLE объект (звук, изображение, электронная таблица Excel и т.д.). Тип объекта не указывается в свойствах поля, он зависит от приложения, использованного для создания объекта OLE. Таким образом, в один столбец таблицы можно помещать данные разного типа, но этого делать не рекомендуется.

**Гиперссылка** (Hyperlink) – позволяет вставить в поле гиперссылку, указывающую на произвольный фрагмент данных внутри файла или «во вне» (в том числе и в Интернет).

**Вложение** (Attachment) – позволяет добавить файл произвольного типа. Работа с данными этого типа напоминает работу с типом Поле объекта OLE, но как отмечается в справочной системе, использование типа Вложение позволяет работать с файлами более гибко и экономно с точки зрения использования дискового пространства.

Для большинства полей (столбцов таблицы) можно задать дополнительные свойства. Для этого служат вкладки внизу окна конструктора. Ниже перечислены некоторые из этих свойств, но в зависимости от выбранного типа данных список доступных свойств может меняться.

**Подпись** – псевдоним имени поля. Выводится в заголовке столбца в режиме таблицы, в формах и т.д. Например, вы хотите называть столбец «ID», а при отображении таблицы показывать его под заголовком «Табельный номер сотрудника». В этом случае удобно будет использовать подпись.

**Обязательное поле** – определяет, является ли ввод данных в это поле обязательным или допустимо присутствие записей, у которых соответствующее значение не указано. Значение по умолчанию – «Нет», если поставлено «Да», то это эквивалентно ограничению NOT NULL в SQL.

**Формат поля** – позволяет определить, в каком формате отображаются данные в режиме таблицы. Конкретный перечень форматов зависит от типа данных, который выбран для этого поля. Очень часто predetermined формат указывают для данных с типом **Дата/Время**.

**Значение по умолчанию** – позволяет задать значение по умолчанию для поля. Это значение будет присваиваться в случаях, когда при внесении записи в таблицу явно не указано другое.

**Индексированное поле** – свойство, позволяющее указать, будет ли создаваться индекс для данного поля таблицы. Индексы используются для ускорения поиска записей в БД, но для их создания и поддержания требуется дополнительное время и дисковое пространство. Поле может быть не индексированным, индексированным и допускать совпадение значений, индексированным и не допускать совпадения значений. Уникальный индекс (совпадение значений не допускается) также является ограничением целостности. Он автоматически создается для первичного ключа таблицы, и можно его создать вручную для альтернативных ключей. По первичным и альтернативным ключам будет посвящена отдельная тема. В данной лабораторной работе Вам предстоит лишь поверхностное знакомство.

**Условие на значение** – позволяет задать ограничения на возможные значения поля, если такие ограничения присутствуют в предметной

области. Например, дата рождения человека не может быть позже текущей даты.

**Сообщение об ошибке** – позволяет определить текст сообщения, выводимого Access в случае ввода в поле недопустимых значений.

В режиме конструктора можно задать ключевое поле таблицы. Его наличие необязательно, но желательно, так как это является важным ограничением целостности и позволяет повысить скорость обработки БД. Часто в качестве ключевого поля используется специально вводимое поле типа Счетчик. Но можно использовать и поле другого типа (кроме Логического, МЕМО, Поля объекта OLE или Вложения) или несколько полей (для составного ключа). Для этого в режиме конструктора надо выделить необходимое поле (или несколько, удерживая клавишу Shift или Ctrl) и нажать кнопку с изображением ключа на панели инструментов (вкладка Конструктор).

Теперь перейдем к созданию таблиц в базе данных. В качестве примера рассмотрим условную БД, созданную для учета книг в кафедральной библиотеке. Мы хотим хранить данные о том, какие издания у нас есть, сколько экземпляров каждого издания, и каков текущий «статус» у каждого экземпляра (в библиотеке, в учебном классе, выдан на руки).

*Задание. Откройте базу lib0.accdb (ссылка на скачивание оформлена на странице сайта). В этой базе данных три таблицы, ознакомьтесь с их структурой. В ходе выполнения лабораторной работы нужно будет в новой базе воспроизвести созданную в образце структуру таблиц и поупражняться в экспорте и импорте данных из БД в Access.*

Начнем с таблицы Book, в ней хранится информация об имеющихся в библиотеке изданиях. В качестве ключа в таблице используется поле BookID типа Счетчик. Это так называемый суррогатный ключ: реальной характеристикой издания это значение не является, а введено специально, чтобы можно было легко выбрать первичный ключ.

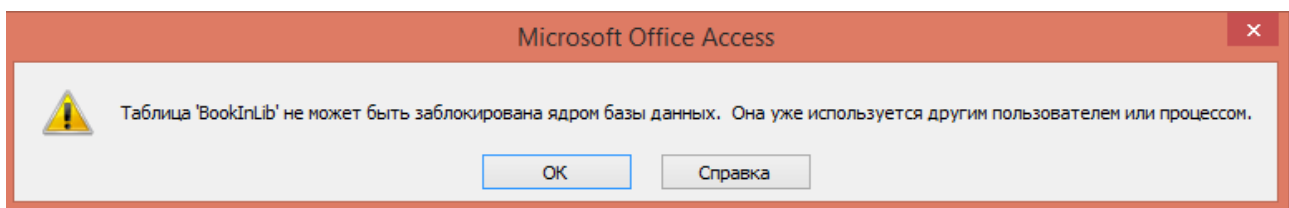
Другие поля таблицы Book содержат данные об авторе (Author), названии (Title), издательстве (Publisher) и годе издания книги (BookYear).

*Задание. Вернитесь к недавно созданной пустой базе данных. На вкладке меню Создание нажмите пиктограмму Конструктор таблиц. Введите названия полей и их типы, как в образце. Определите первичный ключ таблицы. Обратите внимание, что поле Title является обязательным. Сохраните таблицу под именем Book.*

Необходимо проверить, чтобы год издания книги находился в диапазоне от 1000-го до текущего. Функция Now() позволяет получить текущее системное время и дату. Выделить из полученного значения год можно с помощью функции Year(). Выражение Between позволяет проверить принадлежность значения заданному интервалу. Таким образом, для проверки нашего условия можно использовать Between 1000 And Year(Now()) .

*Задание. Задайте ограничение на значения поля BookYear и сообщение, указывающее на причину ошибки при вводе недопустимого года. В созданную таблицу внесите какие-нибудь значения и проверьте правильность работы ограничения.*

*Если в ходе выполнения задания вы видите следующее предупреждение, то Вы недавно работали с другой таблицей и не сохранили изменения. Во избежание ошибок следует таблицу, указанную в предупреждении, закрыть, так как на нее ссылается таблица, с которой Вы в данный момент работаете.*



Следующая работа будет проходить с таблицей BookStatus, в которой перечисляются возможные состояния экземпляра издания – он может находиться в библиотеке, в учебном классе или выдан на руки.

*Задание. Создайте таблицу BookStatus. В качестве первичного ключа используйте поле с типом Счетчик. Название статуса сделайте обязательным и потребуйте его уникальности.*

И наконец – третья таблица. Ее структура самая интересная, если первые две описывали справочную информацию (об издании или о возможных состояниях экземпляра книги из библиотеки), то третья таблица описывает конкретный объект – книгу, которая, с одной стороны, является экземпляром издания, а с другой – находится в одном из возможных состояний.

*Задание. Создайте таблицу BookInLib и определите в ней поле LibID – уникальный номер книги, тип поля – Числовой, размер поля – Длинное*

целое (т.к. это значение будем задавать явно, тип Счетчик здесь не подойдет). Далее нам нужны два поля с тем же типом данных, в одном из которых будет храниться идентификатор, указывающий на издание, в другом – идентификатор, указывающий на состояние.

Но вводить числовой идентификатор, соответствующий тому или иному изданию, для пользователя сложнее, чем выбрать нужный пункт в списке. Подстановка в Access позволяет сделать так, чтобы у пользователя была возможность работать с выпадающим списком, а в таблице хранились числовые значения. Для задания подстановки нужно сохранить таблицу, после чего в выпадающем списке Тип данных, одного из созданных полей выбрать пункт Мастер подстановок (рис. А.1.2).

Имя поля	Тип данных
LibID	Числовой
BookID	Числовой
StatusID	Текстовый
	Поле МЕМО
	Числовой
	Дата/время
	Денежный
	Счетчик
	Логический
	Поле объекта OLE
	Гиперссылка
	Вложение
	Вычисляемый
	Мастер подстановок.

Рис.А.1.2. Запуск мастера подстановок

С помощью мастера можно легко определить нужные параметры. В нашем примере, данные для подстановки мы берем из таблиц (этот пункт надо выбрать в первом диалоговом окне мастера), во втором окне указываем, что эта таблица Book, в третьем – те столбцы, которые надо включить в подстановку (рис. А.1.3).



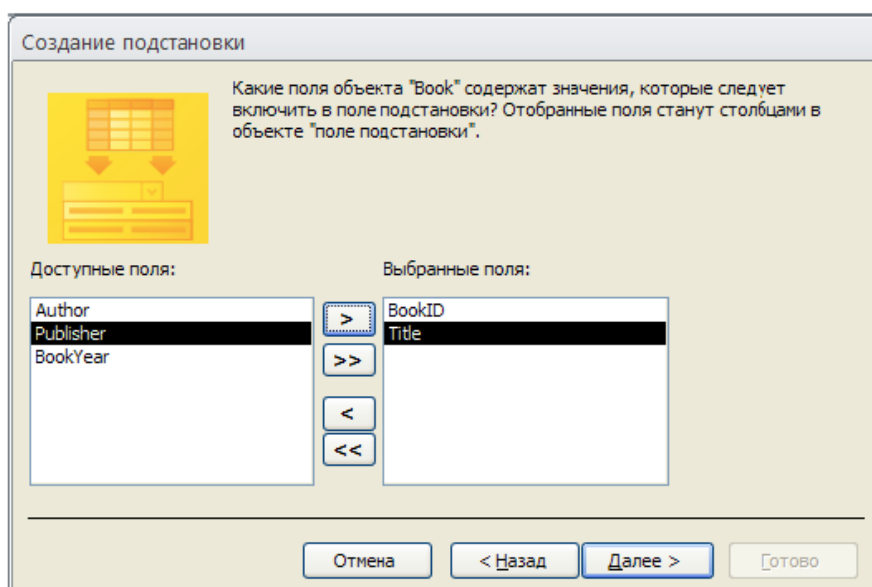


Рис.А.1.3. Выбор столбцов для подстановки

Здесь надо пояснить, как мы выбираем столбцы. В таблице BookInLib мы хотим хранить числовое значение идентификатора книги, взятое из столбца BookID, а пользователю будем показывать название из столбца Title, поэтому данные столбцы и участвуют в подстановке.

Далее можно выбрать порядок сортировки записей в списке подстановки или пропустить этот пункт. Ключевой столбец (т.е. столбец BookID с идентификаторами) пользователю показывать совершенно не обязательно, поэтому в следующем окне надо согласиться с настройкой Мастера подстановок «скрывать этот столбец». В последнем окне Мастера можно включить проверку целостности данных (рис. А.1.4).

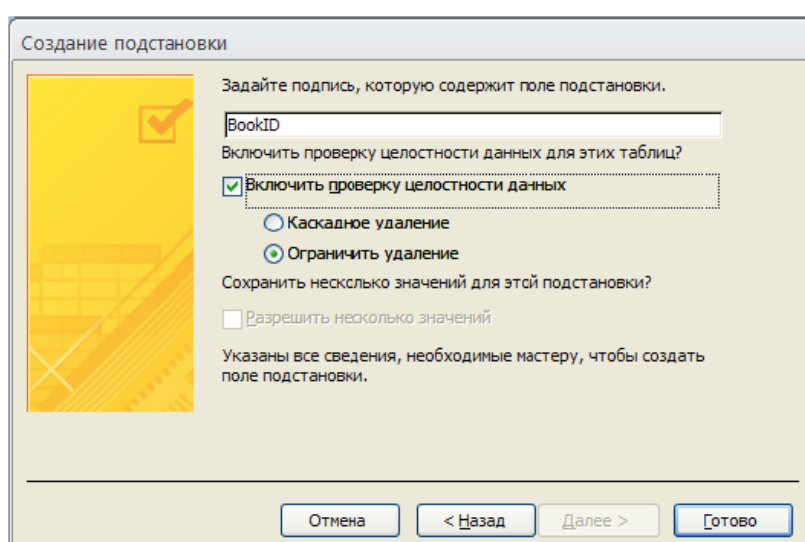


Рис.А.1.4. Определение проверки целостности данных

В некоторых версиях Access мастер не предлагает последний пункт. Не переживайте, в таком случае проверка целостности данных будет определена позже – при работе связки таблиц.

По сути, эта настройка (проверка целостности данных) создает в таблице BookInLib внешний ключ, ссылающийся на таблицу Book. При попытке удаления записи из таблицы Book, для которой есть ссылающиеся на нее записи в таблице BookInLib, в зависимости от настроек с которыми он был создан, внешний ключ может или ограничить (запретить) удаление, или каскадировать его. В последнем случае, вместе с записью из «родительской» таблицы будут удалены и все связанные с ней записи «дочерней» таблицы. Но с этой настройкой надо быть аккуратнее, т.к. в этом случае пользователь базы данных может, удаляя одну запись, неожиданно для себя удалить и множество связанных с ней.

*Задание. В соответствии с приведенным описанием создайте подстановку для поля BookID в таблице BookInLib. После окончания работы Мастера обратите внимание на вкладку Подстановка в свойствах этого столбца (рис.А.1.5). Сделанные настройки можно просмотреть и при необходимости подправить.*

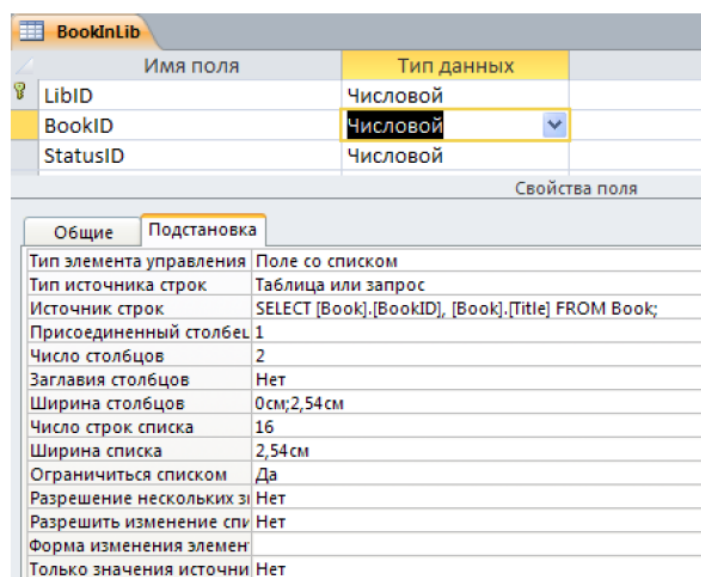


Рис.А.1.5. Вкладка Подстановка после окончания работы Мастера подстановок

*Задание. Самостоятельно определите подстановку с проверкой целостности данных для другого столбца таблицы (указывающего состояние книги) – он должен ссылаться на ключ таблицы BookStatus, а в выпадающем списке должно отображаться название статуса книги. Добавьте в таблицу BookInLib описание нескольких книг: проверьте работу созданных постановок.*

Здесь стоит отметить, что если подстановка названия статуса дает пользователю полную информацию о статусе книги, то вывод в столбце подстановки только названия книги без прочих атрибутов, может привести к неоднозначности выбора, если у нас есть несколько изданий одной и той же книги. В этом случае для создания пользовательского интерфейса правильнее будет использовать составные формы, о которых пойдет речь во второй лабораторной работе

### Схема данных

Она позволяет наглядно отобразить связи между таблицами БД. Если выбрать вкладку Работа с базами данных → Схема данных, то можно увидеть диаграмму с таблицами и связями между ними (рис. А.1.6).

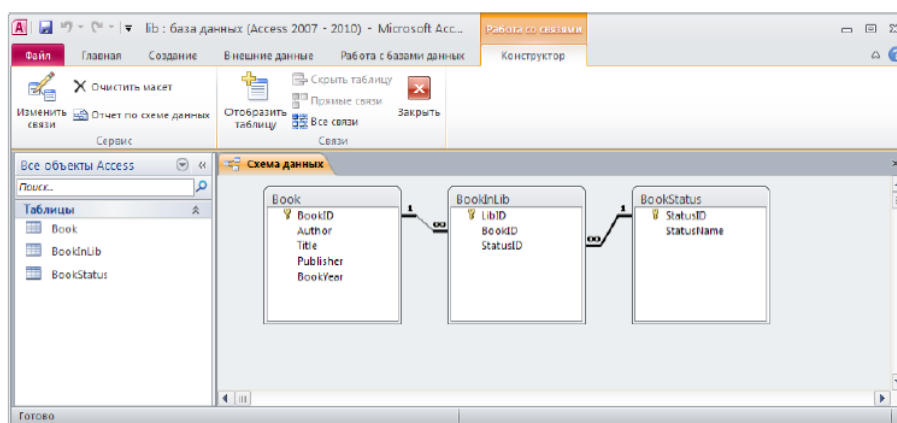


Рис.А.1.6. Схема данных

Если какие-то таблицы на схеме не представлены или она вообще пустая, нужно выбрать в контекстном меню «Добавить таблицу» и из списка выбрать нужные таблицы.

Схема данных Access явно поддерживает два типа отношений между таблицами: «один – ко – многим» и «один – к – одному».

Если таблицы связаны отношением «один – ко – многим», то запись в одной таблице (где поле – ключевое) может быть связана по значению поля с несколькими записями в другой таблице.

При отношении «один – к – одному» запись в одной таблице может иметь не более одной связанной записи в другой таблице и наоборот. Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением «один – к – одному» используют для разделения очень широких таблиц или для отделения части таблицы по соображениям защиты.

Отношение «многие – ко – многим» можно реализовать в Access через создание третьей связующей таблицы, с которой две начальных таблицы связаны «один – ко – многим».

Мастер подстановок, кроме собственно определения подстановки, определил и связь между таблицами. Работая со схемой данных, ее параметры можно отредактировать. Для этого нужно щелчком мыши выделить связь на диаграмме и в контекстном меню выбрать пункт Изменить связь...

*Задание. Откройте схему данных. Убедитесь, что созданные Мастером постановок связи между таблицами обеспечивают контроль целостности данных. Если связь отличается от примера – то настройте ее самостоятельно через настройки.*

### **Импорт и экспорт данных**

Access может конвертировать данные из файлов ряда других форматов в свой и наоборот. Воспользуемся функцией экспорта данных для того, чтобы сохранить данные из таблицы БД Access в электронную таблицу Excel.

*Задание. Откройте базу, которая шла вместе с описанием работы. Откройте таблицу Book и в меню выберите Внешние данные → Экспорт в таблицу Excel. Сохраните таблицу в виде файла Excel. Посмотрите особенности структуры созданного файла Excel.*

*Вернитесь к своей базе данных и импортируйте в нее таблицу Excel. Для этого в меню Внешние данные выберите Импорт электронной таблицы Excel. Поместите в базу ранее экспортированные данные. При этом сохраните в качестве первичного ключа таблицы поле BookID (в процессе импорта надо будет уточнить тип данных этого поля). А имя задайте отличное от Book (так как таблица с таким именем уже существует в Вашей базе данных.)*