

# Глава 2. Построение базы данных

# План

- 2.1. Основные принципы проектирования БД
- 2.2. Создание новой базы данных
- 2.3. Создание таблиц
- 2.4. Сохранение таблицы
- 2.5. Определение связей между таблицами
- 2.6. Модификация БД

# 2.1. Основные принципы проектирования БД

---

План

Создание БД должно начинаться с ее проектирования. Процесс проектирования БД включает следующие основные этапы:

1. Определение назначения БД. На первом этапе проектирования БД необходимо определить список задач, решаемых с ее помощью, и какие данные для этого нужны.
2. Определение структуры таблиц. Это один из наиболее сложных этапов в процессе создания БД. Правильная структуризация БД позволяет быстро извлечь нужные данные, исключает их дублирование и обеспечивает целостность хранящейся информации. Процедура разделения сложных данных на несколько таблиц называется нормализацией.

- Специалистами по проектированию БД была разработана теория нормализации БД . Она рекомендует при проектировании таблиц руководствоваться следующими основными принципами:
- Каждая из таблиц должна содержать информацию о наборе однотипных объектов, например, сведения о студентах или итоги сдачи сессии.
- Каждому из таких наборов данных должна соответствовать отдельная таблица. Например, сведения о студентах и о полученных ими оценках в сессию должны храниться в разных таблицах. Тогда при удалении сведений об оценках студента информация о нем останется в БД.
- Информация в таблице не должна дублироваться. Не должно быть повторений и между таблицами. Это исключает возможность несовпадения информации в разных таблицах и делает работу с БД более эффективной.

3. Определение полей. Каждая таблица содержит информацию о наборе объектов одного типа. При разработке полей для таблицы необходимо помнить следующее:

- В таблице должна присутствовать вся необходимая информация о данном наборе объектов.
- Каждое поле должно содержать сведения о том или ином свойстве именно этого, а не других наборов объектов. Исключением могут являться поля, используемые для связи с другими таблицами. Не рекомендуется включать в таблицу данные, которые являются результатом вычисления значения некоторого выражения, так называемые вычисляемые поля.

- Информацию следует разбивать на наименьшие логические единицы. Например, в таблицу со сведениями об адресах студентов лучше включить поля Общежитие и Комната, а не общее поле Адрес. Это даст возможность осуществить в БД поиск студентов, живущих в данной комнате, или произвести сортировку записей по номерам комнат.

- 4. Определение ключевых полей. Для того чтобы Access мог связать данные из разных таблиц, каждая таблица должна содержать первичный ключ, или просто ключ. Это одно или несколько полей, совокупность значений которых однозначно определяет каждую запись в таблице. Наличие ключа в таблице исключает возможность появления в ней двух одинаковых записей. При создании новой таблицы Access предлагает построить ключ путем добавления дополнительного поля с уникальными значениями.



- 5. Определение связей между таблицами. После распределения данных по таблицам и определения ключевых полей необходимо выбрать схему для связи данных в разных таблицах. Для этого нужно определить связи между таблицами. Эти связи Access будет использовать при создании много табличных запросов, форм и отчетов.

## 2.2. Создание новой базы данных

---

- Для создания БД нужно выбрать в меню Файл команду Создать или нажать кнопку Создать базу данных на панели инструментов. Будет открыто диалоговое окно Создание. В этом окне следует выбрать диск и каталог для сохранения БД и ввести имя файла новой БД. Access автоматически добавит к нему расширение `mdb`. В этом файле хранятся данные, а также описания структуры таблиц, запросов, форм, отчетов и других объектов создаваемой БД. На экране появится окно новой «пустой» базы данных. Она постепенно заполняется содержимым по мере создания с помощью соответствующих мастеров или «вручную» нужных таблиц, а затем и других объектов.

## 2.3. Создание таблиц

---

План

- Для создания таблицы нужно щелкнуть по корешку вкладки Таблица окна БД, а затем — по кнопке Создать. Access открывает окно Новая таблица и предлагает несколько способов создания таблицы.

## 2.3.1. Режим таблицы

- При выборе варианта Режим таблицы появляется заготовка таблицы, содержащая 30 строк (записей) и 20 столбцов (полей) со стандартными именами Поле1, Поле2 и т.д. После заполнения таблицы Access автоматически определит тип полей в зависимости от внесенной в них информации. Для изменения имени поля нужно сделать двойной щелчок по заголовку соответствующего столбца, ввести новое имя и нажать клавишу Enter.

Страницы : таблица		
Имя поля	Тип данных	Описание
Код студента	Текстовый	Номер расчетной книжки
Группа	Текстовый	Номер учебной группы
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Курс	Числовой	Номер курса
Пол	Текстовый	
Дата рождения	Дат/время	

Свойства поля

Общие Подстановка

Размер поля 6

Формат поля

Маска ввода

Подпись

Значение по умолчанию

Условия на значение

Сообщение об ошибке

Обязательное поле Да

Пустые строки Нет

Недекорированное поле Да (Совпадения не допускаются)

Формат вывода значений данного поля. Выберите стандартный формат или создайте новый. Для справки по функциям нажмите клавишу F1.

- После завершения ввода информации в таблицу нужно щелкнуть по кнопке Сохранить и дать имя созданной таблице. Затем Access предложит добавить в нее ключевое поле. Если это предложение будет принято, то в таблицу будет добавлено поле Код типа Счетчик, содержащее номера введенных записей. Использование режима таблицы не позволяет установить нужные свойства полей. Для этого следует перейти в режим конструктора таблицы, вызываемый нажатием кнопки Конструктор окна БД.



## 2.3.2. Режим конструктора таблицы

- При создании таблицы в этом режиме открывается окно таблицы в режиме конструктора (см. рис. 2.1). В верхней части окна находится пустой бланк таблицы, содержащей три графы, и для создания таблицы нужно заполнить, по крайней мере, две из них: Имя поля и Тип данных. Для каждого из полей будущей таблицы следует выполнить следующие действия.

## 1. Определить имя поля таблицы

- Чтобы определить поле таблицы, нужно ввести в ячейку столбца Имя Поля имя создаваемого поля (до 64 алфавитно-цифровых символов, включая пробелы).

## 2. Определить тип данных поля.

- Для этого следует нажать клавишу Tab и перейти в столбец Тип данных. В этом столбце можно оставить выводящееся по умолчанию значение Текстовый или, раскрыв список, выбрать нужный тип данных.

- Тип данных определяет вид информации, сохраняемой в поле.
- Например, если поле имеет тип Числовой, то Access не позволит ввести в это поле текст. По типу данных поля Access определяет не только, какие данные могут сохраняться в этом поле, но и сколько места для них надо зарезервировать. Для хранения данного типа Дата/время требуется 8 байт памяти, текстовое поле требует 1 байт для каждого символа.

# Основные типы данных:

- текстовый — алфавитно-цифровые данные, до 255 байт;
- поле MEMO — комментарии и пояснения, до 64 Кбайт;
- числовой — числовые данные;
- дата/время — значения даты и времени;
- денежный — денежные значения;
- счетчик — автоматически вставляющиеся последовательные номера;
- логический — логические значения типа Да/Нет, Истина/Ложь или Вкл/Выкл;
- объект OLE — рисунок, таблица Excel, документ Word или другие данные в двоичном формате;
- гиперссылка — строка, состоящая из букв и цифр и представляющая адрес гиперссылки.

- 3. Ввести описание поля (необязательно)

Для более подробного описания полей можно ввести пояснительный текст о содержащихся в поле данных в ячейку столбца Описание. Этот текст будет появляться в строке состояния при выборе данного поля в режиме таблицы или формы.

- 4. Задать свойства поля (необязательно). Каждое поле имеет набор характеристик, называемых свойствами, которые задают условия хранения и отображения данных. Этот набор определяется типом данных поля. После указания типа данных Access выводит список свойств в нижней части окна конструктора и дает возможность задать нужные значения свойств поля. Настройка свойств поля позволяет, в частности, проконтролировать правильность вводимых данных, а нередко упростить сам процесс ввода.

- Ниже приводится перечень и краткое описание свойств поля. Для получения подробной справки нужно щелкнуть по интересующему свойству и нажать F1.
- Размер поля. Задает максимальный размер данных, которые будут храниться в поле. Текстовое поле может иметь размер от 1 до 255 символов (по умолчанию — 50 символов). Размер числового поля зависит от выбранного типа числовых данных. Access допускает следующие диапазоны значений для числовых данных.

- Байт — целые числа в пределах от 0 до 255.
- Целое (2 байта) — целые числа в пределах от -32768 до 32767.
- Длинное целое (4 байта) — целые числа в пределах от -2 147 483 648 до 2 147 483 647.
- С плавающей точкой (4 байта) — числа в пределах от  $-3,4 \cdot 10^{38}$  до  $3,4 \cdot 10^{38}$  с точностью до 6 разрядов.
- С плавающей точкой (8 байт) — числа в пределах от  $-1,8 \cdot 10^{308}$  до  $1,8 \cdot 10^{308}$  с точностью до 10 разрядов.



- **Формат поля.** Это свойство поля задает формат представления его значений при выводе на экран или печать. Формат, установленный для поля при создании таблицы, используется по умолчанию в содержащей это поле форме или запросе.
- Для числовых, денежных, логических полей, а также поля типа дата/время Access предоставляет список стандартных форматов. Этот список вызывается щелчком мыши (см. рис. 2.2). Пользователь может создать свой собственный формат поля (см. справку Access).

Формат поля	Основной	
Число десятичных знаков	Основной	3456,789
Маска ввода	Денежный	3 457р.
Подпись	Фиксированный	3456,79
Значение по умолчанию	С разделителями разрядов	3 456,79
Условие на значение	Процентный	123,00%
Сообщение об ошибке	Экспоненциальный	3,46E+03

<i>Символ</i>	<i>Назначение</i>
0	цифра от 0 до 9
9	может быть цифра или пробел
#	может быть цифра, пробел или символы плюс и минус
L	буква
?	может быть буква
A	буква или цифра
A	может быть буква или цифра
&	любой символ или пробел
C	может быть любой символ или пробел
<	перевод всех последующих символов на нижний регистр
>	перевод всех последующих символов на верхний регистр
.,:;- /	разделители, сохраняющие свой вид в строке данных
!	указывает, что маску нужно заполнять справа налево
\	указывает, что следующий символ надо воспринимать не как символ маски, а буквально
«текст»	ВВОД ПОСТОЯННЫХ ТЕКСТОВЫХ СИМВОЛОВ

- Маска ввода. Устанавливает символы форматирования для их автоматического заполнения во время ввода данных, например, добавления скобок и дефисов в полях номеров телефонов. Маску ввода можно использовать для преобразования вводимых символов к нужному регистру. В нее можно также включить строку символов, которая будет сохраняться в этом поле или использоваться для форматирования при выводе на экран. Описание маски ввода состоит из трех разделов, разделенных точкой с запятой.

- Первый раздел — это собственно маска ввода, состоящая из символов маски и постоянных символов.
- Второй раздел указывает, следует ли сохранять постоянные текстовые символы маски в поле. Если постоянные символы нужно включить в значение поля, то в этом разделе следует указать 0; в противном случае в этом разделе указывается 1. Если этот раздел отсутствует, то сохраняются только введенные символы.
- Третий раздел задает символ — указатель позиций, заполняемых при вводе. Допускается использование любого символа. Если этот раздел отсутствует, то для представления пустых символов используется символ подчеркивания ( \_ ).

- Ниже приводятся описания некоторых масок ввода и примеры значений, которые могут быть в них введены (см. табл. 2.2).
- Для создания маски ввода можно воспользоваться услугами мастера, который вызывается при нажатии кнопки Построить .

<i>Описание маски</i>	<i>Примеры значений</i>
<p>&gt;L&lt;???????????????</p> <p>(999) 900-00-00;0</p> <p>(\0\95) 000-00-00</p>	<p>Смирнов; Мария (маска для ввода слова длиной до 15 букв; первая буква становится заглавной).</p> <p>(381) 23-45-67; (812) 321-43-55 (маска для ввода телефонных номеров; в поле хранятся все символы маски).</p> <p>(095) 123-54-87; (095) 321-55-88 (маска для ввода московских телефонных номеров; в поле хранятся только семь цифр номера без кода и дефисов).</p>

- Подпись. Задает альтернативное имя, которое будет появляться в качестве заголовка поля при просмотре таблицы или созданных на ее основе запросах, формах или отчетах. Например, если установить значение № зачетки для подписи поля Код студента таблицы Студенты, то при переходе в режим таблицы это название будет выведено в качестве заголовка данного поля.



- **Значение по умолчанию.** Задает значение, автоматически вводящееся в поле при создании новой записи. Для числового поля значение по умолчанию равно 0, а для текстового или Memo поля — значение Null (данные в поле неизвестны).
- **Условие на значение.** Задает логическое выражение, определяющее условие для ввода или редактирования данных. Выражение принимает значение «Истина» (True), если значение в поле удовлетворяет заданному условию, и «Ложь» (False) — в противном случае. Access не позволит ввести в поле значение, не удовлетворяющее условию, и выдаст сообщение об ошибке.

<i>Условие</i>	<i>Комментарий</i>
>= 0	значение должно быть больше или равно 0
< #10.03.98#	дата, предшествующая 10 марта 1998 года
>10 Or Is Null	значение должно быть больше 10 или пусто
<= 20 And >= 10	значения должны находиться между 10 и 20
Not «Москва»	значение, отличное от слова «Москва»
Between 0 And 100	значения должны находиться между 0 и 100
"Омск" Or "Томск"	любое значение из списка городов
In ("Омск"; "Томск")	любое значение из списка городов
In (1; 3; 5; 11; 17)	любое значение из списка чисел
Like "A*"	любой текст, начинающийся с буквы А

- Условие на значение задается выражением, состоящим из операторов сравнения ( $>$ ,  $>=$ ,  $<$ ,  $<=$  и т.д.) и операндов — значений, используемых для сравнения. Если выражение не содержит оператора, то Access будет использовать оператор «равно» ( $=$ ). Условие может содержать несколько сравнений, разделенных логическими операторами Or (логическое сложение) или And (логическое умножение). Для отрицания некоторого условия используется логический оператор Not (логическое отрицание).

- Текстовые значения должны заключаться в кавычки. Если в качестве операнда используется дата, то ее нужно заключить в символы номера (#). Для проверки, попадает ли значение в заданный интервал, можно использовать логический оператор **And** или оператор **Between**. Чтобы проверить, содержится ли значение в заданном списке, нужно использовать логический оператор **Or** или оператор **In**. Проверку соответствия вводимого значения текстового поля или поля **Memo** некоторому шаблону можно осуществить с помощью оператора **Like**.

- Для создания условия на значение поля можно использовать построитель выражений, который вызывается при нажатии кнопки Построить . Он содержит перечень операторов, функций и констант, которые можно добавлять в создаваемое выражение.
- Сообщение об ошибке. Содержит текст сообщения, которое будет выведено на экран при вводе в поле данных, нарушающих условие на значение. Например, если задать для поля Оценка условие  $In (2; 3; 4; 5)$  и ввести текст сообщения: Введена неверная оценка, то при попытке ввода любого символа, отличного от вышеперечисленных цифр, Access откажется занести его в таблицу Сессия и выдаст соответствующее сообщение. От значений следующих двух свойств поля зависит, как Access будет интерпретировать отсутствие значения в поле.

## Пустые (Null) значения и пустые строки

При вводе информации в таблицу некоторые поля могут остаться незаполненными из-за отсутствия соответствующих данных. В Access различаются следующие ситуации:

- *Значение в поле существует, но неизвестно.* В этом случае в поле ничего не вводится, и Access будет хранить в нем так называемое *пустое значение*, обозначаемое *Null*. Значение *Null* может использоваться в выражениях и вводиться в качестве критерия отбора в запросы. Ключевые поля не могут содержать значений *Null*.
- *Известно, что значение в поле отсутствует.* В этом случае в поле вводятся две кавычки и его значение — *пустая строка* (""). Пустую строку можно ввести только в текстовое и Memo поле.

- **Обязательное поле.** Позволяет указать, обязателен ли ввод данных в это поле. Если задать для этого свойства значение Нет и оставить поле пустым (ввести пробелы или сразу нажать клавишу Enter), то Access будет считать, что в это поле введено значение Null. Если же это свойство имеет значение Да, то значения Null в этом поле не допускаются и при вводе новой записи в это поле обязательно должно быть введено значение.

- Пустые строки. Если установить значение этого свойства равным Да, то для текстового или Memo поля будет разрешен ввод пустых строк. В противном случае Access преобразует введенные пустые строки, а также строки, состоящие только из пробелов, в строки, содержащие значение Null (при условии, что свойство Обязательное поле имеет значение Нет).



- Отметим, что если свойство Пустые строки имеет значение Да, то пустые строки являются допустимыми значениями данного поля вне зависимости от значения свойства Обязательное поле. Если же для свойства Обязательное поле задано значение Да, а для свойства Пустые строки значение Нет, то Access не разрешит ввод в поле пустой строки и потребует ввести непустое значение.

- **Индексированное поле.** Создается индекс по данному полю. Индекс указывает местоположение записей таблицы на диске и помогает Access быстро находить нужные данные. Если таблица содержит много записей и часто проводится операция поиска или сортировки по определенным полям, то ее выполнение можно значительно ускорить, создав индекс по этим полям. Access также использует индексы для установления связей между таблицами. Ключевые поля таблицы индексируются автоматически.

- При индексировании поля есть две возможности. Выбор варианта Да (Совпадения не допускаются) означает, что создается уникальный индекс. В этом случае таблица не может иметь в этом поле повторяющиеся значения. При выборе варианта Да (Совпадения допускаются) создается индекс, учитывающий возможность повторения значений в этом поле.

- Для создания индекса, основанного на нескольких полях (составного индекса) нужно щелкнуть по кнопке Индексы панели инструментов и ввести в столбце Индекс окна диалога имя индекса. Затем в столбце Имя поля следует указать первое поле индекса и добавить дополнительные поля в расположенные ниже строки, не вводя других имен индексов. В столбце Порядок сортировки можно изменить порядок сортировки для любого поля, входящего в создаваемый индекс. Для удаления индекса нужно щелкнуть по кнопке Индексы, выделить строки, определяющие удаляемый индекс, и нажать клавишу **Delete**.

# Задание

# свойства

# поля

Для задания свойства поля нужно щелкнуть мышью по изменяемому свойству и выполнить одно из следующих действий:

- если в ячейке появится кнопка раскрытия списка, нажать эту кнопку и
- выбрать значение из списка.
- если рядом с ячейкой появится кнопка Построить , нажать эту кнопку. После открытия окна строителя выражений ввести нужное значение или выражение.

Установленные свойства поля таблицы автоматически переносятся на использующие это поле запросы, формы и отчеты.

## Создание ключевого поля

Как уже говорилось, обычно таблица содержит ключевое поле (ключ).

Использование ключа дает следующие преимущества:

- уникальность записей — Access не позволяет вводить и хранить в таблице записи, имеющие одинаковое значение в ключевом поле;
- связи — используя ключ, Access устанавливает связи между таблицами;
- скорость — Access создает индекс по ключевому полю, ускоряющий поиск нужных записей и выполнение запросов;
- упорядочение — Access автоматически сортирует и отображает записи таблицы в порядке возрастания или убывания значений в ключевом поле.

- Простейший способ создания ключевого поля заключается в создании поля типа Счетчик и объявлении его ключевым. Если до сохранения созданной таблицы ключ не был определен, то Access создает его именно таким способом.
- Если в таблице имеется поле, содержащее значения, уникальные для каждой записи, то это поле можно объявить ключевым. Пример поля такого типа — поле Код студента в таблице Студенты, содержащее номера зачетных книжек студентов. Такое поле называется простым ключом.

- В том случае, когда нельзя гарантировать уникальность значений ни одного из полей, можно создать ключ, состоящий из нескольких полей. Такое ключевое поле называется составным ключом. Пример составного ключа — совокупность полей Код студента и Код экзамена в таблице
- Сессия. Ни одно из этих полей по отдельности не может использоваться в этой таблице в качестве ключевого, так как каждое из них содержит повторяющиеся значения. Однако комбинация значений этих полей уникальна, поэтому их совокупность может служить ключом.



- Чтобы объявить одно или несколько полей ключевыми, нужно выделить эти поля, щелкнув по ним мышью. Если полей несколько, то их выделение следует производить, держа нажатой клавишу **Ctrl**. Затем нужно щелкнуть по кнопке Ключевое поле панели инструментов.
- Для удаления ключа достаточно выделить составляющие его поля и повторно щелкнуть по кнопке Ключевое поле. Другой способ — щелкнуть по кнопке Индексы и затем удалить индекс **PrimaryKey**.

# Подстановка данных

- Подстановка данных является очень полезной возможностью в Access. Ее использование во многих случаях позволяет существенно упростить процедуру ввода данных. Фактически подстановка сводится к созданию столбца подстановки — списка, обычно содержащего несколько столбцов. Значения одного из них, так называемого присоединенного столбца, заносятся в поле таблицы, называемое полем подстановки. Остальные столбцы служат комментариями, обеспечивая выбор нужного элемента списка.

- Пусть, например, нужно ввести в таблицу Сессия результаты сдачи какого-либо экзамена. Экзаменационная ведомость содержит следующую информацию: фамилию, имя и отчество студента, его оценку и некоторые другие данные. Так как в таблице Сессия студент представлен не фамилией, а кодом, то при вводе информации приходится всякий раз по фамилии студента отыскивать его код в таблице Студенты и затем вводить найденное значение в БД.

- Ясно, что при такой организации ввода значительная часть времени и усилий затрачивается не на саму операцию ввода, а на поиск нужных кодов, причем вполне возможно занесение ошибочного кода. С помощью подстановки удастся заменить поиск и ввод кода студента следующей существенно более удобной процедурой: из общего списка студентов, созданного Access на базе таблицы Студенты, выбирается нужный студент, после чего его код автоматически заносится в таблицу Сессия.

# Мастер подстановок

- Чтобы создать подстановку для поля таблицы, проще всего использовать соответствующий мастер. Для этого нужно выбрать в качестве типа данных поля значение: Мастер подстановок. Опишем работу мастера на примере создания подстановки для поля Код студента в таблице Сессия.

1. На первом шаге мастер предлагает указать источник данных для столбца подстановки:

- значения из таблицы или запроса;
- фиксированный список значений.

По умолчанию используется первый пункт. Поэтому, так как мы собираемся использовать в качестве источника данных таблицу Студенты, нужно просто щелкнуть по кнопке Далее.

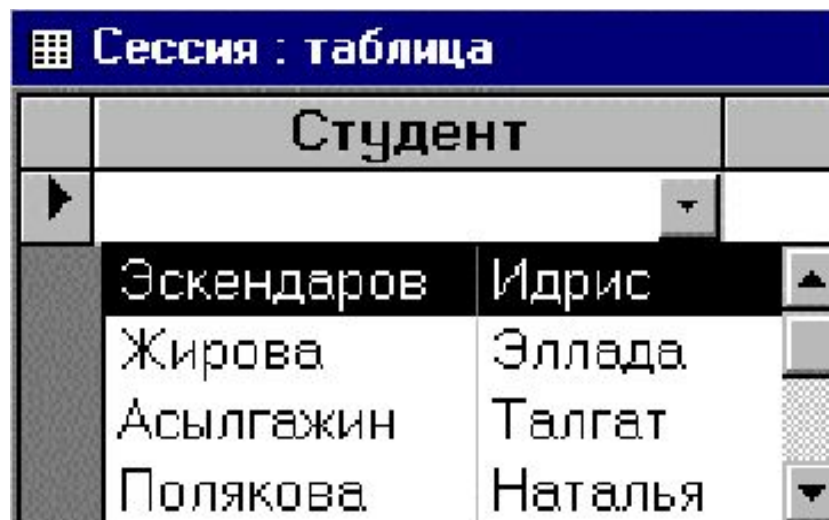
2. Мастер просит указать таблицу или запрос, содержащие столбец подстановки. В списке таблиц нужно выбрать таблицу Студенты и щелкнуть по кнопке Далее.

3. Затем нужно двойным щелчком мыши отобразить поля, используемые в подстановке: Код студента, Фамилия и Имя. Последнее поле добавлено, чтобы иметь возможность различать студентов-однофамильцев.

4. Так как Код студента — ключевое поле, то Access автоматически выбирает его в качестве источника данных (присоединенного столбца) для поля подстановки. На этом шаге можно изменить ширину полей, используемых в подстановке, и указать, надо ли скрыть присоединенный столбец. Оставим без изменения установку по умолчанию: Скрыть ключевой столбец.



5. На последнем шаге зададим подпись для столбца подстановки: Студент и нажмем кнопку Готово. Access попросит сохранить таблицу, и операция создания подстановки завершена.



The screenshot shows a table in Microsoft Access. The table is titled 'Сессия : таблица' (Session : table). The column header is 'Студент' (Student). The data in the column is as follows:

Студент
Эскендаров Идрис
Жирова Эллада
Асылгажин Талгат
Полякова Наталья

- Для просмотра полученных результатов перейдем в режим таблицы, щелкнув по кнопке Вид на панели инструментов. В первом столбце Студент вместо кодов студентов появятся их фамилии. Перейдем на последнюю (пустую) запись таблицы, щелкнем по ячейке в поле Студент, а затем по появившейся справа кнопке. Откроется список, содержащий фамилии и имена студентов (см. рис. 2.3). Если выбрать щелчком мыши одного из студентов, то в ячейке появится его фамилия, а в таблицу Сессия будет занесен его код.

- В качестве источника данных для столбца подстановки может использоваться фиксированный список значений. Например, можно создать столбец подстановки для поля Оценка, указав на первом шаге работы мастера, что значения берутся не из таблицы, а из списка. Затем следует создать список, содержащий перечень возможных оценок: 2, 3, 4 и 5. После завершения создания столбца подстановки значения в поле Оценка можно будет не вводить с клавиатуры, а выбирать из списка.

Общие

Подстановка

Тип элемента управления	Поле со списком
Тип источника строк	Таблица/запрос
Источник строк	SELECT DISTINCTROW [Студенты].[Код сту
Присоединенный столбец	1
Число столбцов	3
Заглавия столбцов	Нет
Ширина столбцов	0см;2,54см;2,54см
Число строк списка	4
Ширина списка	5,079см
Ограничиться списком	Да

Чтобы изменить свойства поля подстановки, созданного с помощью мастера, нужно щелкнуть по ярлычку Подстановка. На рис. 2.4 видны свойства поля Код Студента. Ниже дается их краткое описание:

- Тип элемента управления — задается представление этого поля в форме.
- Тип источника строк — указывается тип источника данных для поля. Им может быть таблица/запрос (по умолчанию) или список значений.
- Источник строк — определяет источник данных. Если тип источника строк — таблица/запрос, то здесь указывается имя таблицы/запроса или инструкция SQL. Если тип источника строк — список значений, то указывается список элементов, разделяемых точкой с запятой.
- Присоединенный столбец — указывается номер столбца, значения из которого заносятся в поле.

- Число столбцов — задает число выводящихся столбцов.
- Заглавия столбцов — выводятся (Да) или нет (Нет) в качестве заголовков столбцов имена полей или первые элементы списка значений.
- Ширина столбцов — задается ширина выводимых столбцов (через точку с запятой). Чтобы скрыть столбец, нужно установить его ширину, равной 0.
- Число строк списка — задает максимальное число строк, выводящихся в раскрывающемся списке.
- Ширина списка — задает ширину раскрывающегося списка.
- Ограничиться списком — указывает, что в поле вводятся только значения, принадлежащие списку (Да), или разрешен ввод любых других значений (Нет). Access разрешит вводить в поле значения, не принадлежащие списку, лишь в том случае, если присоединенный столбец является первым отображаемым столбцом в списке.



## Свойства таблицы



### Общие

Описание . . . . .	Заказы на книги
Условие на значение . . . . .	[Дата получения] <=[Дата отправки]
Сообщение об ошибке . . . . .	Попытка ввода ошибочной даты
Фильтр . . . . .	
Порядок сортировки . . . . .	

## ● Установка свойств таблицы

Кроме задания свойств полей пользователь имеет возможность задать свойства таблицы. Эти свойства относятся ко всей таблице и всем ее записям. В частности, можно задать условия на значения в записях, относящиеся не к одному, а нескольким полям таблицы, и указать текст выводимого сообщения при нарушении этих условий.



- Для установки свойств таблицы нужно щелкнуть правой кнопкой мыши в окне конструктора и выбрать в появившемся контекстном меню пункт Свойства. В появившемся диалоговом окне Свойства таблицы

# Мастер таблиц

- Чтобы создать таблицу с помощью мастера, нужно в окне диалога Создание таблицы нажать кнопку Мастер. Мастер Создание таблиц предлагает на выбор различные образцы таблиц, каждая из которых содержит набор полей. Пользователь должен отобрать те поля, которые он собирается включить в свою таблицу. После этого мастер автоматически создает таблицу, предоставляя на завершающем этапе возможность пользователю создать ключевое поле и определить связи с уже существующими таблицами.

- После завершения работы мастера можно перейти в режим конструктора и произвести необходимые изменения в структуре таблицы и свойствах ее полей.

# Импорт таблиц

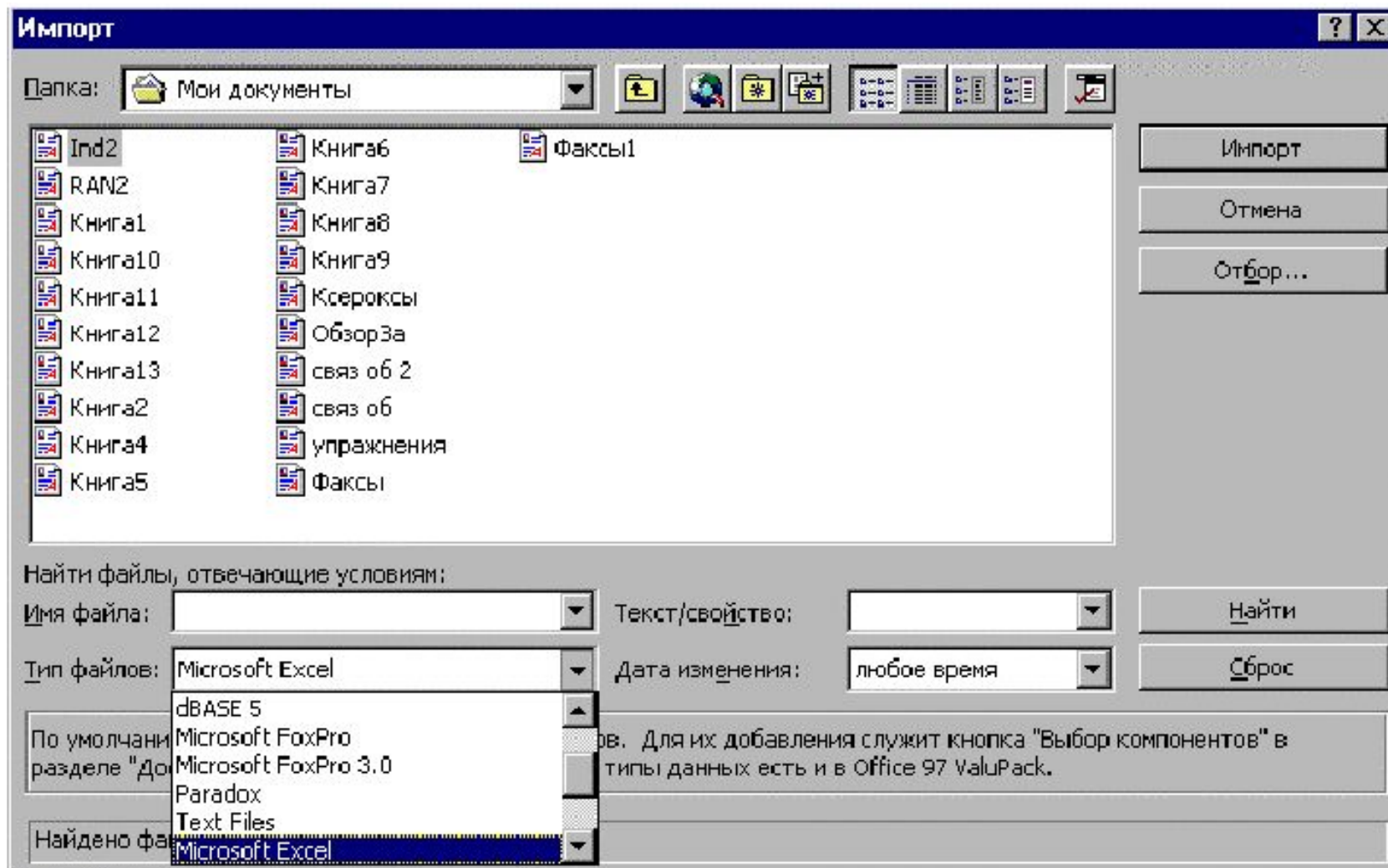
- Таблицу можно создать, импортируя данные из файлов других форматов. Если выбран этот способ создания таблицы, то открывается окно диалога Импорт. Пользователь должен указать тип файла, из которого будут импортированы данные, и найти этот файл. Проще всего производится операция импорта таблицы из другой БД Access, а также таблиц, созданных в формате других СУБД (dBASE, FoxPro, Paradox). Она осуществляется автоматически без участия пользователя.

- При импорте электронных таблиц (файла Excel или Lotus 1-2-3) можно прочитать целый лист или именованный диапазон. При этом импортируемая таблица должна иметь стандартный формат баз данных, когда столбцы являются полями, и каждая строка представляет собой отдельную запись. После выбора файла электронной таблицы вызывается мастер Импорт электронной таблицы и пользователю предлагается ответить на ряд вопросов.

- Сначала он должен выбрать вариант импорта данных: лист или именованный диапазон. Затем указать, содержит ли первая строка источника данных имена полей, и выбрать вариант сохранения данных: в новой или существующей таблице. На следующем шаге пользователь имеет возможность изменить названия полей, создать индексы и отметить поля, не включаемые в таблицу. В заключение мастер предлагает создать ключевое поле и дать таблице имя.

- По сходной схеме производится импорт данных из текстового файла.
- Для осуществления операции импорта можно также вызвать щелчком правой кнопки мыши по окну базы данных контекстное меню и выбрать в нем пункт Импорт.

# Окно для выбора файла при импорте данных





# Связывание таблиц

- Альтернативным способом использования в Access данных из других источников (баз данных, электронных таблиц или текстовых файлов) является установление связи с внешними данными — связывание данных. При импорте данных создается их копия в новой таблице текущей БД. После этого всякая связь с внешними данными теряется. Связывание позволяет использовать данные из внешнего файла, не импортируя их в Access, и в этом случае пользователь всегда имеет дело с самой «свежей» информацией. Формат данных файла-источника не меняется и его можно продолжать использовать в приложении, в котором он был создан.

- Поэтому в тех случаях, когда внешний файл часто изменяется или используется в режиме коллективного доступа, целесообразно вместо операции импорта использовать связывание. Правда, это может привести к определенному снижению быстродействия при работе с данными, так как Access максимально эффективно работает со своей копией данных и в «родном» формате.

- Операция связывания производится при помощи мастера Связь, который начинает работу при выборе пункта Связь с таблицами контекстного меню окна БД. Операция связывания таблицы из другой БД осуществляется автоматически без участия пользователя. При связывании листа или именованного диапазона электронной таблицы мастер запрашивает, содержит ли первая строка источника данных имена полей, и затем устанавливает связь.

- Для связанных таблиц изменение некоторых свойств полей становится невозможным. В связанную таблицу нельзя добавить новое поле или удалить существующее поле, но можно добавить записи, а также изменить значения отдельных полей. Если связь установлена с таблицей Access, то доступна также операция удаления записей.

- В Access для обозначения связанных таблиц и таблиц, хранящихся в текущей БД, используются разные значки. Для удаления связи нужно щелкнуть по значку связанной таблицы и нажать клавишу Delete.

## 2.4. Сохранение таблицы

---

План

- После определения всех полей таблицы нужно нажать кнопку Сохранить на панели инструментов или выбрать в меню Файл команду Сохранить для сохранения макета таблицы.
- Выводится окно диалога Сохранение, в которое следует ввести имя таблицы и нажать кнопку ОК.

## 2.5. Определение связей между таблицами

---



- После того как в БД созданы основные таблицы, следует указать, как они связаны друг с другом. Эти связи Access будет использовать в запросах, формах и отчетах при отборе информации из нескольких таблиц. Задание связей между таблицами позволяет также обеспечить защиту целостности данных в БД.

- Связь между двумя таблицами, одна из которых является главной, а другая подчиненной, устанавливает соответствие между записями этих таблиц. Для установления связи нужно, чтобы в главной таблице существовало поле или группа полей, совокупность значений которых однозначно определяла бы запись (была уникальной). Обычно в качестве поля (группы полей) связи выбирается ключевое поле таблицы, но достаточно, чтобы оно имело уникальный индекс.

- В соответствие ему ставится поле (группа полей) подчиненной таблицы, называемое внешним ключом. Связь между записями устанавливается по совпадению значений в полях связи. Access анализирует значение поля связи (ключа) любой записи главной таблицы и связывает ее со всеми записями подчиненной таблицы, имеющими такое же значение внешнего ключа.

- Поля связи могут иметь разные имена, но они должны иметь один тип данных и иметь однотипное содержимое. Исключение из этого правила: поле типа Счетчик можно связывать с числовым полем, имеющим в свойстве Размер поля значение «Длинное целое». Кроме того, связываемые поля числового типа должны иметь одинаковые значения свойства Размер поля.

- Например, для получения информации о студентах и полученных ими оценках следует определить связь по полям Код Студента в таблицах Студенты и Сессия. Главной в этой связи будет таблица Студенты, а подчиненной — таблица Сессия. Каждой записи в главной таблице — данным о студенте — соответствует несколько (или ни одной) записей из подчиненной таблицы — информация об его оценках. Поле Код Студента в таблице Сессия является внешним ключом.

# таблицами

## 1. Связь типа «один-ко-многим»

- Описанная выше связь между таблицами Студенты и Сессия — пример связи типа «один-ко-многим». Это наиболее распространенный тип связи. При таком типе связи каждой записи в главной таблице могут соответствовать одна, несколько или ни одной записи в подчиненной таблице, а каждая запись в подчиненной таблице не может иметь более одной соответствующей ей записи в главной таблице. Если запись в подчиненной таблице не имеет соответствующей ей записи в главной таблице, то значение поля связи в такой записи должно быть пусто.

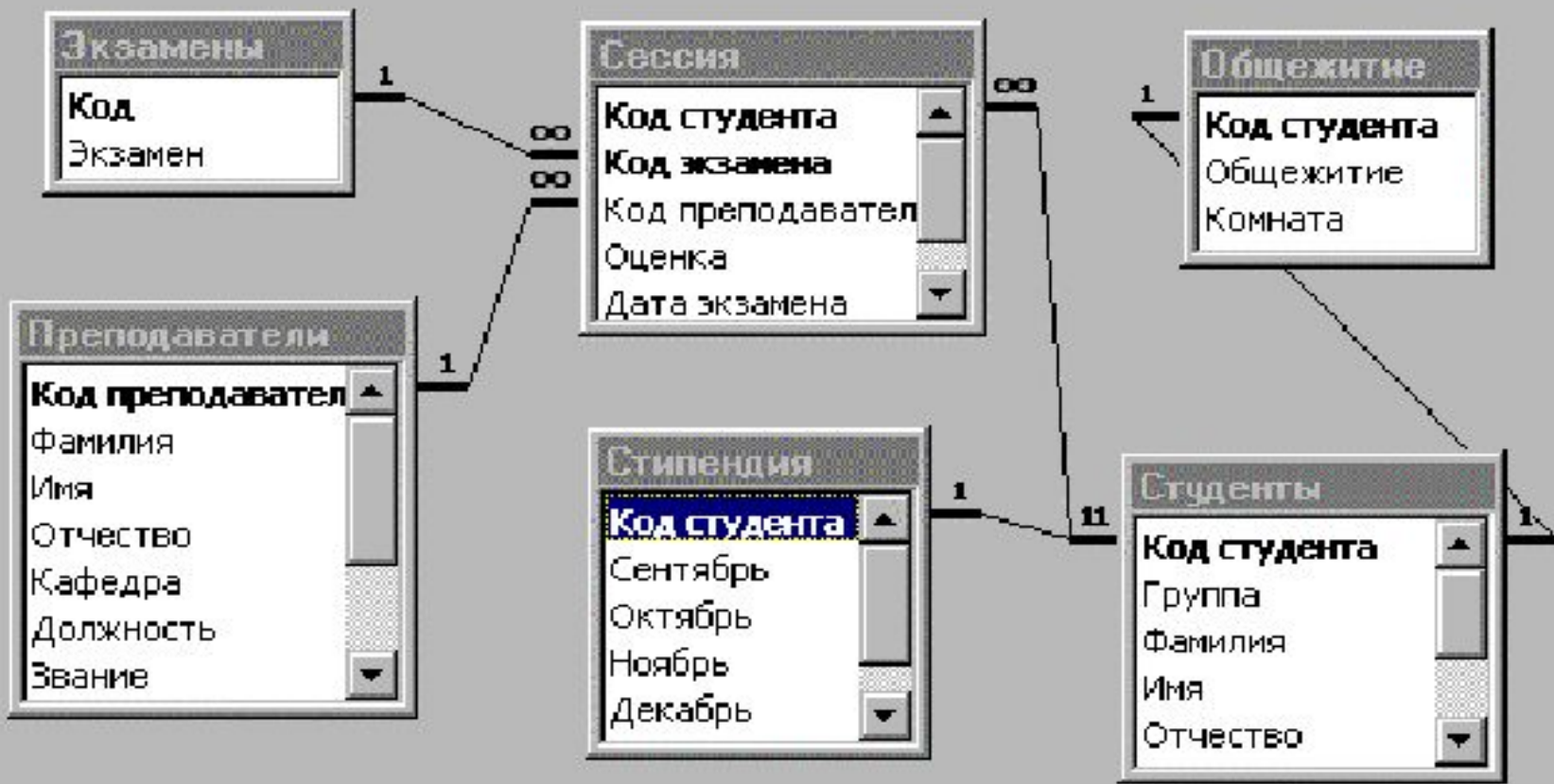
## 2. Связь типа «один-к-одному»

- При связи типа «один-к-одному» каждой записи в главной таблице может соответствовать не более одной записи в подчиненной таблице, и наоборот, каждая запись в подчиненной таблице не может иметь более одной соответствующей ей записи в главной таблице. Как и в предыдущем случае, если запись в подчиненной таблице не имеет соответствующей ей записи в главной таблице, то значение поля связи в такой записи должно быть пусто.

- Этот тип связи применяется реже, так как такие данные могут быть помещены в одну таблицу. Связь типа «один-к-одному» обычно используют для разделения таблиц, имеющих много полей, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице. Например, такой тип связи использован при установлении связей между таблицами Студенты и Общежитие.



Схема данных



- 3. Связь типа «многие-ко-многим»

- При установлении связи между таблицами возможна ситуация, когда между ними нельзя установить отношение «главная-подчиненная» из-за того, что любой записи в одной таблице может соответствовать несколько записей из другой таблицы. Примером могут служить таблицы Студенты и Преподаватели, так как каждый студент сдавал экзамены нескольким преподавателям, а каждый преподаватель принимал экзамен у нескольких студентов. Поэтому между этими таблицами нельзя установить ни одну из описанных выше связей.

- Это пример связи типа «многие-ко-многим». Access непосредственно не поддерживает такой тип связи. Но ее можно реализовать в виде двух связей типа «один-ко-многим» с помощью третьей (связующей) таблицы. В качестве такой связующей таблицы в БД Деканат используется таблица Сессия (см. рис. 2.8). Она связана как с таблицей Студенты по полю Код студента, так и с таблицей Преподаватели по полю Код преподавателя. После того как эти связи установлены, можно легко определить с помощью соответствующих запросов, у каких студентов принимал экзамены данный преподаватель или кому сдавал экзамены данный студент.

## 2.5.2.

## Создание

## связей

- Для создания связей между таблицами нужно вернуться в окно БД, закрыть все открытые таблицы и щелкнуть по кнопке Схема данных панели инструментов либо вызвать щелчком правой кнопки мыши контекстное меню и выбрать в нем пункт Схема данных. Если связи в БД определяются впервые, то будет открыто пустое окно Схема данных. В это окно нужно добавить таблицы, между которыми устанавливается связь.

- Для добавления таблиц следует вызвать щелчком правой кнопки мыши контекстное меню окна Схема данных и выбрать в нем пункт Добавить таблицу. Откроется диалоговое окно Добавление таблицы, содержащее список таблиц БД (см. рис. 2.9). Для добавления таблицы нужно щелкнуть по ее имени, а затем — по кнопке Добавить. После того как все таблицы отобраны, нужно закрыть это окно и вернуться в окно Схема данных.

# Добавление таблицы



Таблицы



Запросы



Таблицы и запросы

Добавить

Закрыть

## Общежитие

Преподаватели

Сессия

Стипендия

Студенты

Экзамены

- Чтобы определить связь между таблицами, находящимися в окне Схема данных, следует перенести с помощью мыши поле связи главной таблицы и поместить его на поле связи подчиненной таблицы. Откроется диалоговое окно Связи

Связи

Таблица/запрос:	Связанная таблица/запрос:
Студенты	Сессия
Код студента	Код студента

Обеспечение целостности данных

каскадное обновление связанных полей

каскадное удаление связанных записей

Тип отношения: один-ко-многим

ОК

Отмена

Объединение...

- В левом столбце выводятся имена главной таблицы и ключа, используемого для связи, а в правом столбце — имена подчиненной таблицы и внешнего ключа. Для изменения поля следует открыть список полей справа от его имени. Если связь производится по нескольким полям, то их имена можно добавить, используя пустые строки.



- Обычно Access сам определяет тип создаваемой связи, проводя анализ полей, для которых определяется связь. Если только одно из полей является ключевым или имеет уникальный индекс, создается связь «один-ко-многим». Связь «один-к-одному» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы.

- Создавая связь, нужно настроить режим обеспечения целостности данных. Обеспечение целостности позволяет избежать ситуации, когда в подчиненной таблице имеются записи, не связанные с записями главной таблицы. Если этот режим включен, то Access не разрешит добавить в подчиненную таблицу запись, для которой не найдется связанной с ней записи из главной таблицы. Нельзя будет также удалить из главной таблицы запись, имеющую связанные с ней записи в подчиненной таблице.

- Например, нельзя будет добавить в таблицу Сессия запись с кодом студента, отсутствующим в таблице Студенты. Соответственно, из таблицы Студенты нельзя удалить запись о студенте, пока в таблице Сессия содержатся сведения о его оценках. Если для связи включен этот режим, то можно дополнительно указать, следует ли автоматически выполнять для связанных записей операции каскадного обновления и каскадного удаления. Если включить режим Каскадное обновление связанных полей, то при изменении значения ключа в главной таблице будут автоматически обновлены соответствующие значения в связанных записях подчиненной таблицы. При включении режима Каскадное удаление связанных записей при удалении записи из главной таблицы будут автоматически удалены связанные с ней записи в подчиненной таблице.

- В том случае, когда эти режимы не включены, а режим обеспечения целостности данных включен, Access не позволит изменить значение в ключевом поле главной таблицы, а также удалить запись в главной таблице, если в подчиненной таблице имеются данные, связанные с этой записью. После завершения операции создания связи нужно нажать кнопку ОК. Связь отображается в виде линии, соединяющей две таблицы. Если включен режим обеспечения целостности данных, то Access изобразит на конце линии, соответствующей главной таблице, цифру 1. На другом конце линии, соответствующем подчиненной таблице, будет изображен символ бесконечности  $\infty$  для связи типа «один-ко-многим» и цифра 1 для связи типа «один-к-одному».

- Любую связь можно изменить или удалить. Для изменения связи нужно сделать двойной щелчок по линии связи, и отредактировать ее в открывшемся окне Связи. Чтобы удалить связь, следует щелкнуть по ней и нажать клавишу Delete.

2.6.

Модификация

БД

План

- Access позволяет достаточно просто внести изменения в БД. Однако, прежде чем вносить в БД изменения, нужно иметь в виду следующее:
  - Access не вносит автоматически сделанные в таблицах изменения в использующие эти таблицы объекты (запросы, формы и т.д.).
  - Нельзя изменять тип данных для поля, которое используется при определении связи между таблицами. Сначала нужно удалить соответствующую связь.
  - Прежде чем открывать таблицу в режиме конструктора для внесения изменений, нужно закрыть все объекты, которые на нее ссылаются.

## 2.6.1. Операции над таблицами

- Копирование таблицы

Чтобы создать копию таблицы в БД, нужно вызвать ее контекстное меню и выбрать пункт Копировать (другой способ — выделить щелчком мыши таблицу и нажать клавиши **Ctrl+C** или щелкнуть по кнопке Копировать панели инструментов). Затем нужно нажать клавиши **Ctrl+V** или щелкнуть по кнопке Вставить панели инструментов. Access откроет диалоговое окно Вставка таблицы и предложит ввести имя новой таблицы; выбрать режим копирования. В зависимости от того, какой режим выбран, Access создаст пустую таблицу, имеющую такую же структуру, что и исходная таблица; создаст точную копию исходной таблицы (этот режим задан по умолчанию) или добавит данные из исходной таблицы в указанную таблицу.



- **Переименование таблицы**

Для переименования таблицы нужно вызвать ее контекстное меню и выбрать пункт Переименовать. Затем нужно ввести новое имя таблицы и щелкнуть по кнопке ОК

- Удаление таблицы

Чтобы удалить ставшую ненужной таблицу в БД, нужно вызвать ее контекстное меню и выбрать пункт Удалить (другой способ — выделить щелчком мыши таблицу и нажать клавишу **Delete**). Access предложит подтвердить ваше желание удалить таблицу и в случае получения положительного ответа удалит ее.

## ● Экспорт таблицы

Для экспорта таблицы Access в формат Excel или Word проще всего выполнить такие действия:

1. Щелчком мыши выделить нужную таблицу.
2. Выбрать пункт меню Сервис, а затем — Связи с Office или щелкнуть мышью по правому краю кнопки Связи с Office .
3. Выбрать команду Анализ в MS Excel или соответственно команду Публикация в MS Word. Access скопирует таблицу в электронную таблицу Excel (таблицу Word в формате RTF) с таким же именем. Ее первая строка будет содержать названия (подписи) полей экспортируемой таблицы. Если файл с таким именем уже существует, то Access предложит заменить этот файл или изменить имя нового файла.

Можно использовать более универсальный способ экспорта, позволяющий экспортировать таблицу Access в другую БД (Access, dBase, FoxPro, Paradox) или файл другого формата (текстовый, документ Word, таблица Excel, документ HTML). Для этого нужно выполнить следующие действия:

1. Вызвать контекстное меню таблицы или открыть меню Файл и выбрать пункт Сохранить как/ экспорт...
2. В диалоговом окне Сохранение объекта следует выбрать параметр Во внешнем файле или базе данных и нажать кнопку ОК.
3. В поле со списком Тип файла выбрать формат БД или файла.
4. В поле со списком Папка выбрать диск и папку, в которую следует поместить экспортируемую таблицу.
5. В поле Имя файла нужно указать имя, которое получит экспортируемая таблица, и нажать кнопку Экспорт.

- Экспорт в другую БД или в формат HTML Access осуществляется автоматически, создавая файл в заданном формате. Если этот формат не поддерживает длинные имена таблиц или кириллицу в названиях полей (FoxPro), то перед осуществлением операции следует внести изменения в структуру исходной таблицы. Если задать экспорт таблицы в текстовый файл, то будет вызван мастер Экспорт текста, который позволит выбрать формат текста (Windows или DOS), настроить формат вывода дат и чисел и указать, какие поля нужно экспортировать. Мастер дает возможность создать текстовый файл с полями фиксированной ширины или с разделителями полей.

## 2.6.2. Модификация структуры таблицы

- Все описанные ниже операции по модификации структуры таблицы производятся в режиме конструктора таблицы.

- Вставка полей
- Чтобы вставить в таблицу новое поле, нужно установить курсор на строку определения поля, перед которым вы хотите вставить новое поле. Затем следует щелкнуть по кнопке **Добавить строки** или выбрать команды **Вставка/Строки**. **Access** вставит пустую строку, которую можно использовать для определения нового поля. Введите его имя, укажите тип и задайте нужные свойства.

- Удаление полей
- Для удаления поля нужно выделить строку его определения, щелкнув по области маркировки строки. Область выделения можно расширить, используя клавиши **Ctrl** и **Shift**. Затем нужно нажать клавишу **Delete** или щелкнуть по кнопке Удалить строки . Если таблица содержит данные в удаляемом поле (полях), то **Access** и их потерю. Поэтому перед такой операцией имеет смысл на всякий случай создать копию таблицы.



- Перемещение полей

Чтобы переместить поле, нужно выделить строку его определения и, держа нажатой мышью, переместить эту строку в другое место области определения полей.

Можно поступить иначе: с помощью клавиш **Ctrl+F8** включить режим Сдвиг (в строке состояния появится индикатор СДВ) и затем для перемещения поля использовать клавиши **↓** и **↑**. Для выключения этого режима нужно нажать на клавишу **Esc**. Аналогично можно переместить выделенную группу полей.

- Копирование полей
- Если несколько полей таблицы имеют близкие определения, то можно создать одно поле, а затем скопировать его нужное число раз. Для выполнения операции копирования поля нужно выделить целиком строку определения поля и скопировать в буфер обмена, используя клавиши **Ctrl+C** или кнопку Копировать панели инструментов. Затем вставить пустую строку (см. пункт Вставка полей) в то место определения таблицы, где должна находиться копия поля, и выполнить операцию вставки из буфера обмена, используя клавиши **Ctrl+V** или кнопку Вставить панели инструментов. После этого вносятся необходимые изменения в название и свойства нового поля.

- **Изменение типа данных и размера поля**

Для изменения типа данных нужно щелкнуть по ячейке типа данных изменяемого поля. Затем щелкнуть по кнопке справа, открывающей список возможных типов данных, и выбрать новый тип данных. С некоторыми ограничениями Access может успешно преобразовать любой тип данных (кроме типа поле объекта OLE). Например, при преобразовании из текстового в числовой формат поле должно содержать только цифры и допустимые разделители.

- Для текстового и числового поля задается его размер. Установленный размер поля можно затем изменить. Если он будет уменьшен, то Access выдаст предупреждающее сообщение о возможной потере некоторых данных. Если вы подтвердите свое желание сохранить сделанные изменения в определении таблицы, то операция будет выполнена, но это может повлечь за собой ошибки в преобразовании данных и их потерю. Поэтому перед такой операцией имеет смысл на всякий случай создать копию таблицы.