

2. Цели занятия:

- формирование представления о связях между таблицами и контроль данных.

- изучение основополагающих принципов разработки и использования современных информационных технологий.

3. Учебные вопросы:

1. Создание и модификация структуры таблиц данных

2. Задание и модификации связей между таблицами данных

3. Создание экранных форм, использующих одну таблицу базы данных

Вопрос 1. Создание и модификация структуры таблиц данных

Реляционная модель данных (РМД) некоторой предметной области представляет собой набор отношений, изменяющихся во времени. При создании информационной системы совокупность отношений позволяет хранить данные об объектах предметной области и моделировать связи между ними. Элементы РМД и формы их представления приведены в табл. 1.1.

Элемент реляционной модели	Форма представления
Отношение	Таблица
Схема отношения	Строка заголовков столбцов таблицы (заголовок таблицы)
Кортеж	Строка таблицы
Сущность	Описание свойств объекта
Атрибут	Заголовок столбца таблицы
Домен	Множество допустимых значений атрибута
Значение атрибута	Значение поля в записи
Первичный ключ	Один или несколько атрибутов
Тип данных	Тип значений элементов таблицы

Таблица 1.1 Элементы реляционной модели

Отношение является важнейшим понятием и представляет собой двумерную таблицу, содержащую некоторые данные.

Сущность есть объект любой природы, данные о котором хранятся в базе данных. Данные о сущности хранятся в отношении.

Атрибуты представляют собой свойства, характеризующие сущность. В структуре таблицы каждый атрибут именуется и ему соответствует заголовок некоторого столбца таблицы.

Математически отношение можно описать следующим образом. Пусть даны n множеств $D_1, D_2, D_3, \dots, D_n$, тогда отношение R есть множество упорядоченных **кортежей**, где $dk \in D_k$, dk - **атрибут**, а D_k - **домен** отношения R . На рис. 1.1 приведен пример представления отношения СОТРУДНИК.

В общем случае порядок кортежей в отношении, как и в любом множестве, не определен. Однако в реляционных СУБД для удобства кортежи все же упорядочивают. Чаще всего для этого выбирают некоторый атрибут, по которому система автоматически сортирует кортежи по возрастанию или убыванию. Если пользователь не назначает атрибута упорядочения, система автоматически присваивает номер кортежам в порядке их ввода.

Формально, если переставить атрибуты в отношении, то получается новое отношение. Однако в реляционных БД перестановка атрибутов не приводит к образованию нового отношения.

Домен представляет собой множество всех возможных значений определенного атрибута отношения. Отношение СОТРУДНИК включает 4 домена. Домен 1 содержит фамилии всех сотрудников, домен 2 - номера всех отделов фирмы, домен 3 - названия всех должностей, домен 4 - даты рождения всех сотрудников. Каждый домен образует значения одного типа данных, например, числовые или символьные.

Отношение СОТРУДНИК содержит 3 кортежа. Кортеж рассматриваемого отношения состоит из 4-х элементов, каждый из которых выбирается из соответствующего домена. Каждому кортежу соответствует строка таблицы (рис. 1.1).

Схема отношения (заголовок отношения) представляет собой список имен атрибутов. Например, для приведенного примера схема отношения имеет вид СОТРУДНИК (ФИО, Отдел, Должность, Д_Рождения). Множество собственно кортежей отношения часто называют **содержимым (телом) отношения**.

Первичным ключом (ключом отношения, ключевым атрибутом) называется атрибут отношения, однозначно идентифицирующий каждый из его кортежей. Например, в отношении СОТРУДНИК (ФИО, Отдел, Должность, Д_Рождения) ключевым является атрибут "ФИО". Ключ может быть **составным (сложным)**, т. е. состоять из нескольких атрибутов.

Каждое отношение обязательно имеет комбинацию атрибутов, которая может служить ключом. Ее существование гарантируется тем, что отношение - это множество, которое не содержит одинаковых элементов - кортежей. Т. е. в отношении нет повторяющихся кортежей, а это значит, что, по крайней мере, вся совокупность атрибутов обладает свойством однозначной идентификации кортежей отношения. Во многих СУБД допускается создавать отношения, не определяя ключи.

Возможны случаи, когда отношение имеет несколько комбинаций атрибутов, каждая из которых однозначно определяет все кортежи отношения. Все эти комбинации атрибутов являются **возможными ключами** отношения. Любой из возможных ключей может быть выбран как первичный.

Если выбранный первичный ключ состоит из минимально необходимого набора атрибутов, говорят, что он является **не избыточным**.

Ключи обычно используют для достижения следующих целей: 1) исключения дублирования значений в ключевых атрибутах (остальные атрибуты в расчет не принимаются);

- 2) упорядочения кортежей. Возможно упорядочение по, возрастанию или убыванию значений всех ключевых атрибутов, а также смешанное упорядочение (по одним - возрастание, а по другим - убывание);
- 3) ускорения работы к кортежами отношения ([подраздел 3.2](#));
- 4) организации связывания таблиц ([подраздел 3.3](#)).

Пусть в отношении R1 имеется не ключевой атрибут А, значения которого являются значениями ключевого атрибута В другого отношения R2. Тогда говорят, что атрибут А отношения R1 есть **внешний ключ**.

С помощью внешних ключей устанавливаются связи между отношениями. Например, имеются два отношения СТУДЕНТ (ФИО, Группа, Специальность) и ПРЕДМЕТ (Назв.Пр., Часы), которые связаны отношением СТУДЕНТ_ПРЕДМЕТ (ФИО, . Назв.Пр. Оценка) (рис. 3.2). В связующем отношении атрибуты ФИО и Назв.Пр образуют составной ключ. Эти атрибуты представляют собой внешние ключи, являющиеся первичными ключами других отношений.

Реляционная модель накладывает на внешние ключи ограничение для обеспечения целостности данных, называемое ссылочной целостностью. Это означает, что каждому значению внешнего ключа должны соответствовать строки в связываемых отношениях.

Поскольку не всякой таблице можно поставить в соответствие отношение, приведем условия, выполнение которых позволяет таблицу считать отношением.

1. Все строки таблицы должны быть уникальны, т. е. не может быть строк с одинаковыми первичными ключами.
2. Имена столбцов таблицы должны быть различны, а значения их простыми, т. е. недопустима группа значений в одном столбце одной строки.
3. Все строки одной таблицы должны иметь одну структуру, соответствующую именам и типам столбцов.
4. Порядок размещения строк в таблице может быть произвольным.

Наиболее часто таблица с отношением размещается в отдельном файле. В некоторых СУБД одна отдельная таблица (отношение) считается базой данных. В других СУБД база данных может содержать несколько таблиц.

В общем случае можно считать, что БД включает одну или несколько таблиц, объединенных смысловым содержанием, а также процедурами контроля целостности и обработки информации в интересах решения некоторой прикладной задачи. Например, при использовании СУБД Microsoft Access в файле БД наряду с таблицами хранятся и другие объекты базы: запросы, отчеты, формы, макросы и модули.

Таблица данных обычно хранится на магнитном диске в отдельном файле операционной системы, поэтому по ее именованию могут существовать ограничения. Имена полей хранятся внутри таблиц. Правила их формирования определяются СУБД, которые, как правило, на длину полей и используемый алфавит серьезных ограничений не накладывают.

Если задаваемое таблицей отношение имеет ключ, то считается, что таблица тоже имеет ключ, и ее называют **ключевой** или **таблицей с ключевыми полями**.

У большинства СУБД файл таблицы включает управляющую часть (описание типов полей, имена полей и другая информация) и область размещения записей.

К отношениям можно применять систему операций, позволяющую получать одни отношения из других. Например, результатом запроса к реляционной БД может быть новое отношение, вычисленное на основе имеющихся отношений. Поэтому можно разделить обрабатываемые данные на хранимую и вычисляемую части.

Основной единицей обработки данных в реляционных БД является отношение, а не отдельные его кортежи (записи).

Вопрос 2. Задание и модификации связей между таблицами данных

При проектировании реальных БД информацию обычно размещают в нескольких таблицах. Таблицы при этом связаны семантикой информации. В реляционных СУБД для указания связей таблиц производят операцию их связывания.

Укажем выигрыш, обеспечиваемый в результате связывания таблиц. Многие СУБД при связывании таблиц автоматически выполняют контроль целостности вводимых в базу данных в соответствии с установленными связями. В конечном итоге это повышает достоверность хранимой в БД информации.

Кроме того, установление связи между таблицами облегчает доступ к данным. Связывание таблиц при выполнении таких операций как поиск, просмотр, редактирование, выборка и подготовка отчетов обычно обеспечивает возможность обращения к, произвольным полям связанных записей. Это уменьшает количество явных обращений к таблицам данных и число манипуляций в каждой из них.

Основные виды связи таблиц

Между таблицами могут устанавливаться бинарные (между двумя таблицами), тернарные (между тремя таблицами) и, в общем случае, n-арные связи. Рассмотрим наиболее часто встречающиеся бинарные связи.

При связывании двух таблиц выделяют основную и дополнительную (подчиненную) таблицы. Логическое связывание таблиц производится с помощью **ключа связи**.

Ключ связи, по аналогии с обычным ключом таблицы, состоит из одного или нескольких полей, которые в данном случае называют **полями связи** (ПС).

Суть связывания состоит в установлении соответствия полей связи основной и дополнительной таблиц. Поля связи основной таблицы могут быть обычными и ключевыми. В качестве полей связи подчиненной таблицы чаще всего используют ключевые поля.

В зависимости от того, как определены поля связи основной и дополнительной таблиц (как соотносятся ключевые поля с полями связи), между двумя таблицами в общем случае могут устанавливаться следующие четыре основные вида связи (табл. 3.2):

- один - один (1:1);
- один - много (1:M);
- много - один (M:1);
- много - много (M:M или M:N).

Характеристика полей связи по видам	1:1	1:M	M:1	M:M
Поля связи основной таблицы	являются ключом	являются ключом	не являются ключом	не являются ключом
Поля связи дополнительной таблицы	являются ключом	не являются ключом	являются ключом	не являются ключом

Таблица 3.2 Характеристика видов связей таблиц

Дадим характеристику названным видам связи между двумя таблицами и приведем примеры их использования.

Связь вида 1:1

Связь вида 1:1 образуется в случае, когда все поля связи основной и дополнительной таблиц являются ключевыми. Поскольку значения в ключевых полях обеих таблиц не повторяются, обеспечивается взаимнооднозначное соответствие записей из этих таблиц. Сами таблицы, по сути, здесь становятся равноправными.

Пример 1. Пусть имеются основная 01 и дополнительная Д1 таблицы. Ключевые поля обозначим символом "*", используемые для связи поля обозначим символом "+".

В приведенных таблицах установлена связь между записью (а, 10) таблицы 01 и записью (а, стол) таблицы Д1. Основанием этого является совпадение значений в полях связи. Аналогичная связь существует и между записями (в, 3) и (в, книга) этих же таблиц. В таблицах записи отсортированы по значениям в ключевых полях.

Сопоставление записей двух таблиц по существу означает образование новых "виртуальных записей" (псевдозаписей). Так, первую пару записей логически можно считать новой псевдозаписью вида (а, 10, стол), а вторую пару - псевдозаписью вида (в,3,книга).

На практике связи вида 1:1 используются сравнительно редко, так как хранимую в двух таблицах информацию легко объединить в одну таблицу, которая занимает гораздо меньше места в памяти ЭВМ. Возможны случаи, когда удобнее иметь не одну, а две и более таблицы. Причинами этого может быть необходимость ускорить обработку, повысить удобство работы нескольких пользователей с общей информацией, обеспечить более высокую

степень защиты информации и т. д. Приведем пример, иллюстрирующий последнюю из приведенных причин.

Пример 2. Пусть имеются сведения о выполняемых в некоторой организации научно-исследовательских работах. Эти данные включают в себя следующую информацию по каждой из работ- тему (девиз и полное наименование работ), шифр (код), даты начала и завершения работы, количество этапов, головного исполнителя и другую дополнительную информацию. Все работы имеют гриф "Для служебного пользования" или "секретно".

В такой ситуации всю информацию целесообразно хранить в двух таблицах: в одной из них - всю секретную информацию (например, шифр, полное наименование работы и головной исполнитель), а в другой - всю оставшуюся несекретную информацию. Обе таблицы можно связать по шифру работы. Первую из таблиц целесообразно защитить от несанкционированного доступа.

Связь вида 1:М

Связь 1:М имеет место в случае, когда одной записи основной таблицы соответствует несколько записей вспомогательной таблицы.

Пример 3. Пусть имеются две связанные таблицы 02 и Д2. В таблице 02 содержится информация о видах мультимедиа-устройств ПЭВМ, а в таблице Д2 - сведения о фирмах-производителях этих устройств, а также о наличии на складе хотя бы одного устройства.

Таблица Д2 имеет два ключевых поля, так как одна и та же фирма может производить устройства различных видов. В примере фирма Sony производит устройства считывания и перезаписи с компакт-дисков.

Сопоставление записей обеих таблиц по полю "Код" порождает псевдозаписи вида: (а, CD-ROM, Acer, да), (а, CD-ROM, Mitsumi, нет), (а, CD-ROM, NEC, да), (а, CD-ROM, Panasonic, да), (а, CD-ROM, Sony, да), (б, CD-Recorder, Philips, нет), (б, CD-Recorder, Sony, да) и т. д.

Если свести псевдозаписи в новую таблицу, то получим полную информацию обо всех видах мультимедиа-устройств ПЭВМ, фирмах их производящих, а также сведения о наличии конкретных видов устройств на складе.

Связь вида М:1

Связь М:1 имеет место в случае, когда одной или нескольким записям основной таблицы ставится в соответствие одна запись дополнительной таблицы.

Пример 4. Рассмотрим связь таблиц 03 и Д3. В основной таблице 03 содержится информация о названиях деталей (Поле11), видах материалов, из которого детали можно изготовить (Поле12), и марках материала (Поле13). В дополнительной таблице Д3 содержатся сведения о названиях деталей (Поле21), планируемых сроках изготовления (Поле22) и стоимости заказов (Поле23).

Связывание этих таблиц обеспечивает такое установление соответствия между записями, которое эквивалентно образованию следующих

псевдозаписей: (деталь!, чугу́н, марка!, 4.03.98, 90), (деталь!, чугу́н, марка2, 4.03.98, 90), (деталь2, сталь, марка!, 3.01.98, 35), (деталь2, сталь, марка2, 3.01.98, 35), (деталь2, сталь, марка3, 3.01.98, 35), (деталь3, алюминий, - , 17.02.98, 90), (деталь4, чугу́н, марка2, 6.05.98, 240).

Полученная псевдотаблица может быть полезна при планировании или принятии управленческих решений, когда необходимо иметь все возможные варианты исполнения заказов по каждому изделию. Отметим, что таблица 03 не имеет ключей и в ней возможно повторение записей. Если таблицу ДЗ сделать основной, а таблицу 03 - дополнительной, получим связь вида 1:М. Поступив аналогично с таблицами 02 и Д2, можно получить связь вида М:1. Отсюда следует, что вид связи (1:М или М:1) зависит от того, какая таблица является главной, а какая дополнительной.

Связь вида М:М

Самый общий вид связи М:М возникает в случаях, когда нескольким записям основной таблицы соответствует несколько записей дополнительной таблицы.

Пример 5. Пусть в основной таблице 04 содержится информация о том, на каких станках могут работать рабочие некоторой бригады. Таблица Д4 содержит сведения о том, кто из бригады ремонтников какие станки обслуживает.

Первой и третьей записям таблицы 04 соответствует первая запись таблицы Д4 (у всех этих записей значение второго поля - "станок!"). Четвертой записи таблицы 04 соответствуют вторая и четвертая записи таблицы Д4 (во втором поле этих записей содержится "станок3").

Исходя из определения полей связи этих таблиц можно составить новую таблицу с именем "04+Д4", записями которой будут псевдозаписи. Записям полученной таблицы можно придать смысл возможных смен, составляемых при планировании работы. Для удобства, поля новой таблицы переименованы (кстати, такую операцию предлагают многие из современных СУБД).

Работа	Станок	Обслуживание
Иванов А.В.	станок1	Голубев Б.С.
Иванов А.В.	станок2	Зыков А.Ф.
Петров Н.Г.	станок1	Голубев Б.С.
Петров Н.Г.	станок3	Голубев Б.С.
Петров Н.Г.	станок3	Зыков А.Ф.
Сидоров В.К.	станок2	Зыков А.Ф.

Таблица "04+Д4"

Приведенную таблицу можно использовать, например, для получения ответа на вопрос: "Кто обслуживает станки, на которых трудится Петров Н.Г?".

Очевидно, аналогично связи 1:1, связь М:М, не устанавливает подчиненности таблиц. Для проверки этого можно основную и

дополнительную таблицу поменять местами и выполнить объединение информации путем связывания. Результирующие таблицы "04+Д4" и "Д4+04" будут отличаться порядком следования первого и третьего полей, а также порядком расположения записей.

Вопрос 3. Создание экранных форм, использующих одну таблицу базы данных

В 7.2.4 был рассмотрен режим непосредственного ввода данных в таблицу. Очевидно, что он имеет весьма ограниченное применение. Это обуславливается как тем, что длина записи может оказаться достаточно большой и вводить информацию в нее в табличной форме будет технически неудобно, так и соображениями более принципиального характера:

о во-первых, структура таблицы должна строиться на основе логики задач хранения информации, которая, вообще говоря, может существенно отличаться от логики ее накопления и ввода;

о во-вторых, важным показателем качества автоматизированной системы является организация ее системы ввода/вывода в виде, максимально приближенном к традиционным формам представления информации на немашинных носителях. Такие формы, как правило, делают программное обеспечение привлекательным для конечного пользователя, уменьшают период его адаптации ко вновь внедряемой системе и позволяют быстро сосредоточиться на решении основных профессиональных задач;

о в-третьих, в сложной и развитой автоматизированной информационной системе должно обеспечиваться разделение доступа к различным группам полей и записей для различных категорий пользователей в зависимости от выполняемых ими функций. Также в определенных ситуациях требуется представить одну и ту же информацию либо в различных видах и разрезах, либо в различных сочетаниях с другой информацией.

Для решения как этих, так и многих других проблем организации интерфейса ввода/вывода данных в Access служит механизм электронных форм. Выберем вкладку Формы главного окна базы данных и нажмем кнопку Создать. Появляющееся диалоговое окно позволяет выбрать как таблицу или запрос, для работы с данными которых составляется форма, так и режим ее создания. В зависимости от квалификации пользователя и, естественно, сложности разрабатываемой формы можно либо воспользоваться встроенными программными надстройками-мастерами, либо сразу начать ее создание с нуля в режиме Конструктора. Весьма плодотворным также оказывается комбинированный подход: сначала используется соответствующий мастер, а затем полученная форма дополнительно дорабатывается в "ручном режиме". Проиллюстрируем сказанное на примере. Создадим форму для работы с таблицей Бумаги, воспользовавшись надстройкой Автоформа: в столбец. В результате получим окно следующего вида.

КодБум	
НаимБум	ОАО "Автомат"
Номинал	1 000,00р.
СуммОбъем	2000
Дата эмиссии	10.01.1993
ТипБум	Акция

Запись: 1 из 5

По умолчанию форме было предложено присвоить такое же имя, как и у таблицы, на основе которой она была создана, то есть Бумаги. Как видно из рис. 7.17, при создании подписей полей программная настройка использовала их соответствующие атрибуты, заданные при конструировании таблицы. Последнее не всегда бывает удобным с точки зрения интерфейса пользователя. Для устранения этих и подобных недостатков нам придется вернуться в режим изменения макета формы (кнопка Конструктор либо пиктограмма Вид на панели инструментов).

На рис. 7.18 показана та же форма в режиме Конструктор. Технология процесса проектирования форм в среде Access сводится к добавлению управляющих элементов и изменению их свойств. В связи с этим при переходе в режим Конструктор >>Ш Экране по умолчанию появляются два дополнительных окна:

Окно Панель элементов, которое предназначено для выбора очередного добавляемого к проектируемой форме управляющего элемента. В конструктор форм Access встроены такие элементы управления, как надпись, поле, кнопка, флажок, переключатель, список, набор вкладок и др. Помимо этого к форме можно подключать специальные (дополнительные) элементы управления OLE, что значительно расширяет возможности развития интерфейса управления данными.

Окно Свойств текущего элемента управления, предназначенное для изменения его атрибутов и настроек, например, цвета, шрифта, размера и т. п.

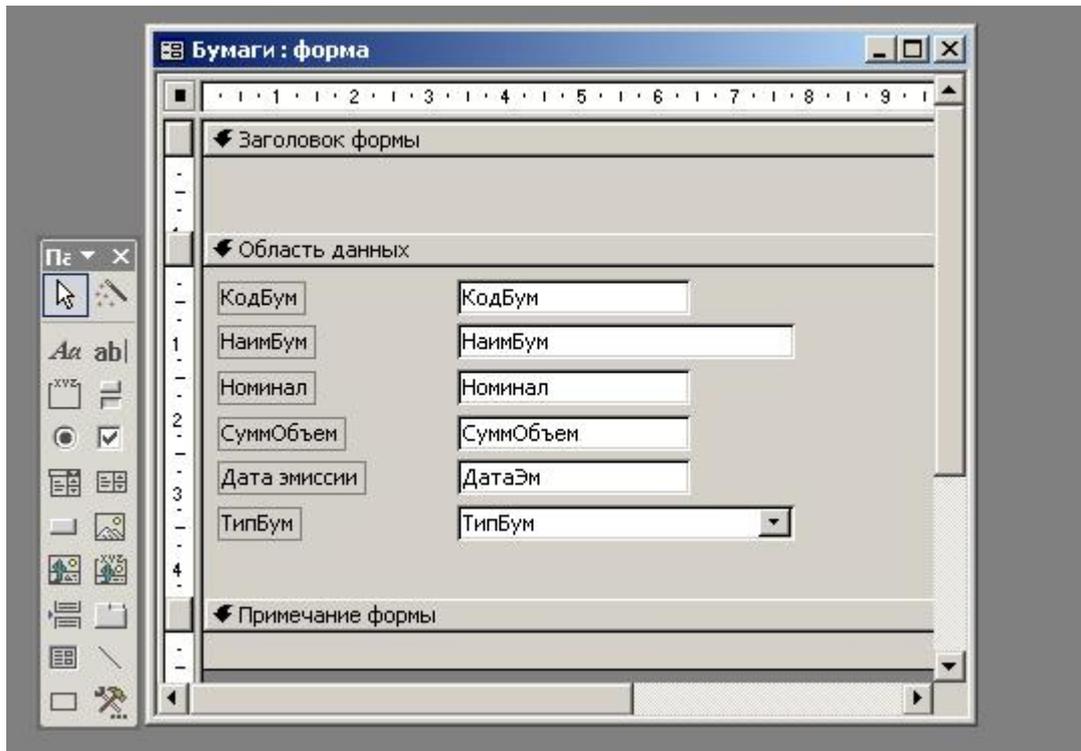


Рис. 7.18. Форма Бумаги в режиме конструктора

В режиме Конструктор явно видна структура формы. Она состоит из трех частей: Заголовок формы, Область данных и Примечание формы. Как нетрудно догадаться, такая структура в первую очередь ориентирована на возможности представления таблично организованных данных. Заметим, что как сама форма, так и ее разделы также рассматриваются как элементы управления, обладающие некоторыми настраиваемыми наборами свойств. В качестве иллюстрации возможностей конструктора по изменению интерфейса ввода/вывода проведем следующие манипуляции над формой Бумаги:

1. Удалим фоновый рисунок: очистим свойство Рисунок, когда текущим выбранным элементом является вся форма.

2. Изменим цвет фона: выберем элемент ОбластьДанных и изменим у нее атрибут Цвет фона (рис. 7.19).

3. Изменим внешний вид полей: выделим группу полей (поля выбираются с помощью мыши при нажатой клавише Shift) и в окне свойств изменим значение атрибута Оформление на Утопленное.

4. Отредактируем подписи полей и несколько изменим их расположение друг относительно друга: для этого достаточно воспользоваться возможностями визуального редактирования элементов.

5. Добавим разделительную линию после поля НаимБум (наименование бумаги): для этого следует воспользоваться элементом Линия.

6. Добавим кнопку завершения работы с формой: в большинстве ситуаций эту и подобные операции проще и удобнее делать в режиме мастера (нажата соответствующая кнопка на панели Элементы управления). В этом случае от пользователя требуется лишь ввести минимальное количество

параметров для добавляемого программного компонента. Добавленную кнопку поместим в область Примечания формы.

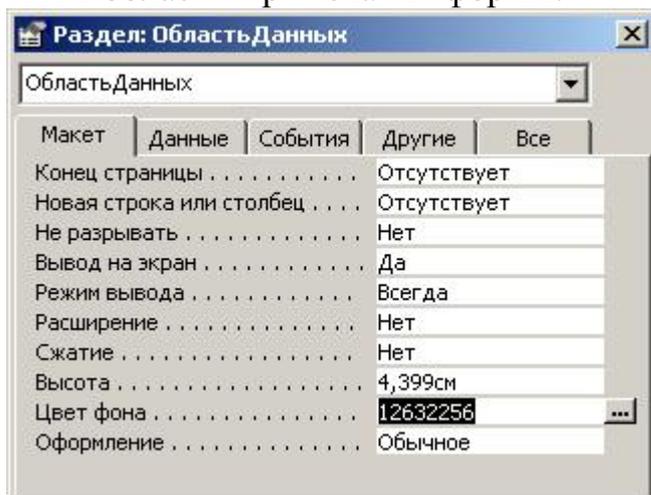


Рис. 7.19. Окно свойств элемента управления

В результате отредактированная форма Бумаги примет вид, показанный на рис 7.20.

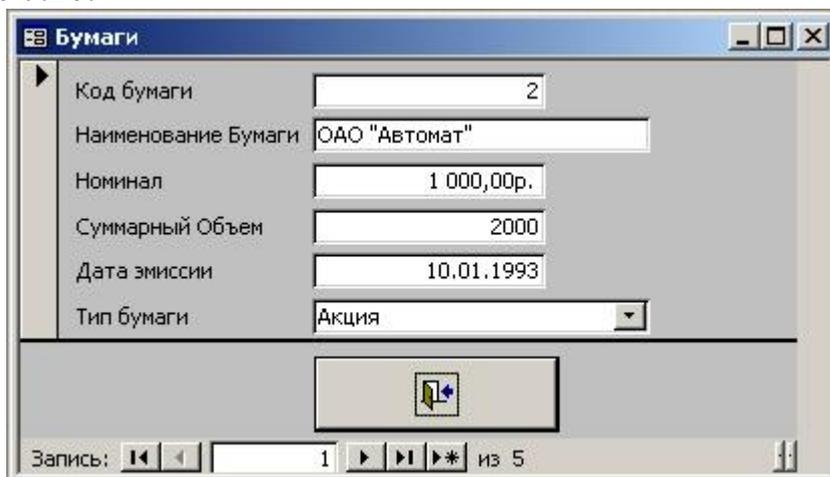


Рис. 7.20. Форма Бумаги после редактирования

Пример организации ввода/вывода данных в таблицу Бумаги с помощью одноименной формы носит в некотором смысле вырожденный характер: в нем структура полей в форме однозначно соответствует их структуре в таблице. Однако, как правило, при создании реальных приложений приходится решать задачу управления Данными, находящимися в системе взаимосвязанных таблиц, из единой формы. В качестве примера рассмотрим задачу построения формы, в которой для каждой данной бумаги одновременно выводится информация по заявкам на ее покупку и продажу. Ее внешний вид приведен на рис. 7.11. Верхняя (заголовочная) часть формы соответствует текущей строке таблицы Бумаги и меняется при переходе от записи к записи, который может производиться с помощью стрелок, расположенных в нижней части окна. Одновременно должны меняться строки таблиц Заявки на продажу и Заявки на покупку, в которые выводится только информация, относящаяся к текущей бумаге.

Рассмотрим более подробно те средства Access, с помощью которых может быть получен такой результат. Это так называемая сложная/или

составная форма (Заявки по бумагам). Процесс ее создания состоит из двух принципиальных этапов:

- создание основной (главной) формы. Для этого осуществляются действия, аналогичные тем, которые выполнялись при создании формы Бумаги;
- создание подчиненных форм. Для этого в созданную главную форму добавляется элемент управления Подчиненная форма. При создании подчиненной формы в Access существует две принципиальные возможности:
 - создать новую форму на базе некоторой таблицы или запроса;
 - воспользоваться уже существующей формой, сделав ее подчиненной.

В данном случае созданы две новые подчиненные формы. Причем созданы они на базе специальных запросов. Такое решение позволяет выделить по отдельности из общей таблицы Заявки записи с заявками на продажу и на покупку. В частности, запрос ЗаявПрод, возвращающий выборку из заявок на продажу ценных бумаг, имеет структуру, показанную на рис. 7.22. В качестве преимуществ такого подхода к организации источника данных для подчиненной формы следует отметить следующие моменты:

о во вспомогательном запросе достаточно просто обработать условие, идентифицирующее тип заявки (если объем заявки меньше нуля, то это заявка на продажу). Более того, для конечного пользователя в качестве объема заявки вместо отрицательных величин выводятся выглядящие более естественно положительные значения $-1 * [\text{ОбъемЗаявки}]$;

о данные выводятся отсортированными по возрастанию предлагаемых цен, что несомненно упрощает процесс работы с ними в экранной форме.

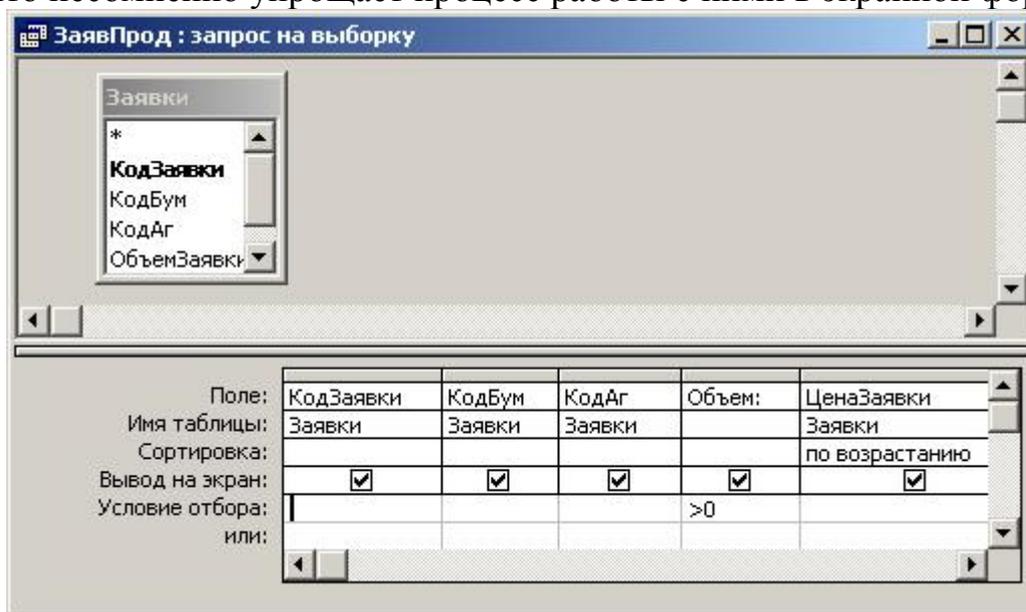


Рис. 7.22. Структура запроса ЗаявПрод (заявки на продажу)

Запрос, возвращающий записи с заявками на покупку, создается аналогично с учетом модификации условия отбор.

Наиболее существенным моментом в процессе внедрения подчиненной формы в главную является правильное задание условия связи между ними. Во многих случаях с этим корректно справляются программные надстройки мастеров. При этом они используют информацию из схемы данных и описаний структуры таблиц. В то же время, не следует забывать и о возможностях

изменения условий связи между ведущей и подчиненной формами в ручном режиме. Для этого необходимо изменить атрибуты в элементе управления Подчиненная форма, находясь в режиме Конструктор

Контрольные вопросы:

1. Создание структуры таблиц данных.
2. Модификация связей между таблицами данных.
3. Описание полей, таблиц и баз данных, связей между таблицами на языке СУБД.
4. Возможности системного меню и экранных форм, реализующих функции создания и модификации структуры таблиц данных

Литература

1. Попов В.Б. Основы информационных и телекоммуникационных технологий.- Финансы и статистика, 2007 г. – 336 с.
2. Угринович Н.Д. Информационные технологии и компьютеризация делопроизводства. – М.:БИНОМ ЛЗ.,2007.-394с.
3. Шамраев А.В. Правовое регулирование информационных технологий (анализ проблем и основные документы). – Статут, 2003. – 1013 с.
4. Хроленко А.Т., Денисов А.В. Современные информационные технологии для гуманитария. – Флинта, 2007. – 128 с.
5. Федотова Е.Л. Информационные технологии в профессиональной деятельности. – Форум, 2008. – 368 с.
6. Емельянова Н.З., Партыка Т.Л., Попов И.И. Основы построения автоматизированных информационных систем. – Форум, 2007. – 416 с.
7. Беяева Т. М., Важнов С. А., Вешняков В. В., Кудинов А. Т., Пальянова Н. В. Основы информатики и математики для юристов. – Элит, 2007. – 368 с.
8. Казанцев С.Я. Информатика и математика для юристов. Учебник. – Юнити, 2008. – 560 с.
9. Строганов М.П., Щербаков М.А. Информационные сети и телекоммуникации. – Высшая школа, 2008. – 151 с.
10. Микрюков В.Ю. Информация, информатика, компьютер, информационные системы, сети. – Феникс, 2007. – 448 с.