МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Пензенский Государственный Университет

Кафедра «Маркетинг»

Пояснительная записка

по дисциплине: «Информационные системы маркетинга»

на тему:

**«Создание базы данных в среде Microsoft Access 2002»**

Выполнила:

студентка гр. 04ЭС-1

Муртазина л. Ф.

Проверил: Убиенных А. Г.

Пенза, 2007

**Введение**

В данном курсовом проекте я познакомилась с основными принципами организации и обработки данных об объектах и явлениях реального мира. Данные вместе с программно-аппаратными средствами для их обработки называют информационными системами (ИС).

В курсовом проекте использовались фактологические ИС, где основа, объект их обработки - база данных (БД). БД - система специальным образом организованных данных, программных, технических, языковых организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования этих данных. Базой данных называется поименованная структурированная совокупность взаимосвязанных данных, относящихся к конкретной предметной области и находящихся под централизованным программным управлением.

Различают централизованные и распределённый БД. *Централизованная* *БД* хранится в памяти одной вычислительной системы. Если такая система является компонентом вычислительной сети, то возможен распределённый доступ к этой базе данных - доступ к ней пользователей различных узлов сети. *Распределённая БД* состоит из нескольких, возможно, пересекающих или даже дублирующих друг друга частей. Однако пользователь распределённой базы данных получает возможность работать с такой базой данных как с единым информационным массивом с помощью СУБД. При разработке информационной системы обычно стремятся, чтобы её БД была интегрированной, что очень важно в маркетинговой деятельности на предприятии.

Базы данных имеют свои преимущества и недостатки. Основными преимуществами БД являются:

1. Наличие единого целостного отображения определённой части реального мира. Оно сокращает избыточность хранимых данных, обеспечивает выполнение новых приложений, обеспечивает реализацию нерегламентированных запросов;

2. Увеличение мобильности и гибкости информационной системы;

3. Легче осуществляется контроль за целостностью и непротиворечивостью данных;

4. Наличие интегрированной БД сокращает затраты не только на создание и хранение данных, но и на их поддержание в актуальном состоянии, а также уменьшает потоки данных циркулирующих в системе.

Основными недостатками БД являются:

1. Увеличение сложности создаваемых информационных систем;

2. Использование интегрированных БД может привести к потере эффективности отдельных приложений;

3. Применение сложных структур данных увеличивая долю службы информации в общем объёме хранимых данных.

В курсовом проекте ставится задача - разработка БД в конкретной предметной области.

Практику работы с ИС я рассмотрю на примере современной системы управления базами данных (СУБД) Microsoft Access.

**1. Описание предметной области**

Если подходить к сбору маркетинговой информации как к случайному, редкому событию, которое необходимо только тогда, когда нужно получить данные по конкретному вопросу, можно столкнуться с рядом проблем.

Например, может возникнуть ситуация, когда результаты предыдущих исследований хранятся в неудобном для использования виде; незаметны изменения в окружающей среде и действиях конкурентов; проводится несистематизированный сбор информации; возникают задержки при необходимости проведения нового исследования; по ряду временных периодов отсутствуют данные, необходимые для анализа; маркетинговые планы и решения анализируются неэффективно; действия представляют собой лишь реакцию, а не предвидение.

Маркетинговые исследования надо рассматривать как часть постоянно действующего интегрированного информационного процесса. Необходимо, чтобы фирма разрабатывала и использовала систему постоянного слежения за окружающей средой и хранения данных с тем, чтобы они могли анализироваться в будущем. Маркетинговую информационную систему можно определить как совокупность процедур и методов, разработанных для создания, анализа и распространения информации для опережающих маркетинговых решений на регулярной постоянной основе.

Сначала фирма устанавливает цели компании, определяющие общие направления планирования маркетинга. На эти цели воздействуют факторы окружающей среды (конкуренция, правительство, экономика). Планы маркетинга включают контролируемые факторы, определенные в предыдущих разделах, включая выбор целевого рынка, целевого маркетинга, тип организации маркетинга, маркетинговую стратегию (товар или услуга, распределение, продвижение и цена) и управление.

Когда план маркетинга определен, с помощью информационной сети, которая включает исследования, постоянное наблюдение и сбор данных, можно конкретизировать и удовлетворять общие потребности маркетинговых служб в информации. Маркетинговое исследование дает точную информацию для решения исследовательских проблем. Для него может понадобиться хранящаяся информация (внутренние вторичные данные) или сбор внешней вторичной и/или первичной информации. Постоянное наблюдение - это процедура, посредством которой регулярно анализируется меняющаяся окружающая среда. Оно может включать изучение бюллетеней новостей, регулярное получение информации от сотрудников и потребителей, присутствие на отраслевых заседаниях и наблюдение за действиями конкурентов. Хранение данных - это накопление всех видов значимой внутрифирменной информации (такой, как объем продаж, издержки, работа персонала и т.д.), а также информации, собранной через маркетинговые исследования и постоянное наблюдение. Эти данные помогают принимать решения и хранятся для дальнейшего использования.

В целом маркетинговая информационная система дает множество преимуществ:

-организованный сбор информации;

-избежание кризисов;

-координация плана маркетинга;

-скорость;

-результаты, выражаемые в количественном виде;

-анализ издержек и прибыли.

Предметной областью называется часть реального мира, представляющая интерес для данного исследования и отражаемая в информационной системе. В автоматизированных информационных системах отображение предметной области представляется моделями нескольких уровней. Инфологическое моделирование имеет специфические особенности, вызванные с одной стороны идеологией организации данных, с другой стороны особенностями СУБД.

Предметной областью моего КП является - маркетинговая среда предприятия. Моя база данных – данные о постоянных клиентах фирмы о взаимодействии с ними. Моя БД предоставляет:

1.Информацию о клиентах, их название, место расположение, контактный телефон, должность и ФИО контактного лица.

2.Информацию о заключенных договорах.

3.Информацию о выплатах, задолжностях и о сроках сделки.

4.Иоформация о времени отгрузки.

**2. Инфологическое проектирование (ER-модель)**

В настоящее время разработано значительное количество разнообразных моделей БД. В большинстве случаев в маркетинговой деятельности используется реляционная модель, когда данные представляются в виде совокупности таблиц, над которыми могут выполняться операции.

Проектирование БД—одна из наиболее ответственных и трудных задач, связанных с созданием информационной системы маркетинговой деятельности. В результате её решения должны быть определены и содержание БД. И эффективный способ её организации, и инструментальные средства управления данными, которые будут применяться в создаваемой системе.

*Процесс проектирования БД должен включать следующие этапы:*

1.Инфологическое проектирование, т.е. определение предметной области системы, позволяющее изучить информационные потребности будущих пользователей:

- определение требований к операционной обстановке, в которой будет функционировать информационная система;

- выбор СУБД и других инструментальных и программных средств её реализации;

- логическое проектирование БД;

- физическое проектирование БД.

Задача этапа логического проектирования БД состоит в разработке её «логической» структуры в соответствии с инфологической моделью предметной области. На это этапе создаются схемы БД на языках определения данных.

Этап физического проектирования БД требует поиска проектных решений, обеспечивающих эффективную поддержку построения «логической» структуры БД в среде хранения БД. На этом этапе решаются вопросы построения структуры хранимых данных, размещения хранимых данных в памяти, выбора эффективных методов доступа к различным компонентам «физической» БД. Описывается также отображение «логической» структуры БД в структуре хранения.

Итак, в процессе этапа инфологического проектирования можно выделить следующие сущности - связи, объекты, атрибуты. Взаимосвязь объектов и связей между ними представлена в виде ER - диаграммы.

Объект - абстракция класса предметов реального мира. Каждый объект обладает присущими ему атрибутами. Каждый атрибут должен иметь имя и должен получать значение из некоторого множества допустимых значений (домен). У каждого объекта должен быть идентификатор, который однозначно определяет экземпляр объекта. Атрибуты бывают простые и ключевые.

В моей БД ключевым атрибутом является «название организации» и «адрес», т.е. по данным атрибутам происходит взаимосвязь всех четырех таблиц.

Объекты бывают статистические и динамические, поэтому надо учитывать количество меняющихся экземпляров объекта от общего числа.

*Описание атрибутов сущности*

1.Договора.

|  |  |  |  |
| --- | --- | --- | --- |
| Название атрибута | Тип поля | Свойства поля | Описание |
| Название организации | Текстовый | 50 | Название организации |
| Номер договора | Текстовый | 50 | Номер договора |
| Адрес | Текстовый | Текстовый | Адрес |

2.Задолжность

|  |  |  |  |
| --- | --- | --- | --- |
| Название атрибута | Тип поля | Свойства поля | Описание |
| Название организации | Текстовый | 50 | Название организации |
| Сумма задолжноти | Денежный | 50 | Сумма задолжности |

3.Клиент

|  |  |  |  |
| --- | --- | --- | --- |
| Название атрибута | Тип поля | Свойства поля | Описание |
| Название организации | Текстовый | 50 | Название организации |
| Адрес | Числовой | 50 | Адрес организации |
| ФИО директора | Текстовый | 50 | ФИО директора |
| Контакты | Числовой | Длинное целое |  |

4.Объем заказа.

|  |  |  |  |
| --- | --- | --- | --- |
| Название атрибута | Тип поля | Свойства поля | Описание |
| Название организации | Текстовый | 50 | Название организации |
| Срок отгрузки | Текстовый | 50 | Время отгрузки |
| Срок приема | Текстовый |  | Время приема |
| Выплата | Денежный | 50 | Сумма платежа |
| Объем заказа | Числовой | Длинное целое | Количество |

*Описание связей*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № связи | Название | Тип | Главный объект | Связан, объект |
| 1 | Договора-клиент | 1:∞ | Адрес | Адрес |
| 2 | Задолжность-клиент | 1:∞ | Название организации | Название организации |
| 3 | Объем заказа-клиент | 1:∞ | Название организации | Название организации |

**3. Разработка базы данных (БД)**

*Основные элементы базы данных Access*

В файле базы данных Access могут храниться объекты шести основных типов. Вместе эти объекты выполняют все основные функции СУБД и составляют законченное приложение.

*Таблицы* содержат элементы данных в табличном формате (строка-столбец), подобном тому, который используется в электронных таблицах. База данных Access может включать до 32768 объектов (общее количество таблиц, форм, отчетов, запросов и т.д.), при этом одновременно можно открыть до 1024 таблиц, при наличии, конечно, достаточных ресурсов. Таблицы можно импортировать из баз данных других приложений (таких как xBase и Paradox), баз данных архитектуры клиент-сервер (таких как Microsoft SQL Server) или из электронных таблиц (например, Microsoft Excel или Lotus 1-2-3). Базы данных Access можно связывать с таблицами баз данных других типов (dBase, FoxPro, Paradox) форматированными файлами (такими, как текстовые файлы в формате ASCII и рабочие листы Excel) и с другими базами данных Access.

*Запросы* связывают между собой данные нескольких таблиц (до 16 таблиц) и отображают их в виде некоторой виртуальной таблицы. Кроме того, с помощью запросов можно указать, какие именно поля выбранных таблиц нужно отобрать (всего до 255 полей) и как на их основании сформировать поля виртуальной таблицы. Критерии отбора определяют не только поля, но и строки отбираемых данных.

*Формы* отображают данные, содержащиеся в таблицах и запросах, и позволяют добавлять, удалять или редактировать информацию. В формы можно вставлять рисунки и диаграммы, а при наличии звуковой карты еще и комментарии и музыку. Далее в работе описано создание форм и добавление в них графических объектов. В Access 97 в формы можно включать собственные процедуры на языке УВА (хранящиеся в модуле класса), с помощью которых можно управлять интерактивным режимом ввода, просмотра и корректировки данных.

*Отчеты* дают возможность печатать данные таблиц и запросов практически в любом формате. Access позволяет включать в отчет графические объекты, благодаря чему можно, например, распечатать полный иллюстрированный каталог продукции фирмы. Отчеты Access предоставляют пользователям более гибкие возможности, чем аналогичные функции большинства других систем управления реляционными базами данных, включая системы, разработанные для мини-компьютеров. В отчеты Access можно включать такие же управляющие VBA-процедуры, как и в формы.

*Создание новой базы данных*

Итак, чтобы создать новую базу данных, выполните следующие действия.

1. Если Access еще не загружена, загрузите ее и перейдите к шагу 3.

2.Если Access уже загружена и открыта одна из баз данных, щелкните на заголовке окна базы данных, чтобы сделать его активным. Если окна базы данных не видно, щелкните на кнопке Окно базы данных панели инструментов или выберите команду Окно1, или нажмите клавишу <F11>.

3.Теперь щелкните на кнопке панели инструментов Создать базу данных или выберите из меню команду Файл, а потом Создать базу данных. Перед вами появится диалоговое окно Создание.

4. В нашем примере нужно выбрать значок Новая база данных и щёлкнуть на кнопке ОК. Появится следующее диалоговое окно Файл новой базы данных.

5. В текстовое поле Имя файла введите имя новой базы данных. Расширение имени файла вводить не нужно. Access добавит его сама.

6. Щёлкните на кнопке Создать или нажмите Enter.

*Свойства таблиц и их полей*

Прежде чем добавить в новую базу данных первую таблицу, следует познакомиться с основными понятиями и соглашениями, которые в Access применяются для описания структуры таблиц и их полей. Основным из этих понятий является понятие свойств таблиц и полей.

Свойства таблицы описывают таблицу в целом. Задавать их не обязательно. Если же вы хотите определить какие-либо из свойств, это можно сделать, введя их значения в текстовые поля диалогового окна свойств таблицы. Для этого нужно в режиме конструктора таблицы щелкнуть на кнопке панели инструментов Свойства. Свойств у таблиц пять.

•Свойство Описание позволяет ввести дополнительное пояснение о назначении таблицы. Если в окне базы данных выбрать Вид, потом Таблица, рядом с названием каждой таблицы появится ее описание. Это свойство удобно использовать также при создании с помощью архивариуса описаний объектов базы данных.

•Свойство Условие на значение позволяет в случае необходимости указать условие для проверки правильности вводимых в таблицу данных. Вводимое в это поле выражение может включать несколько полей таблицы. Это условие относится не конкретному полю, а к таблице целиком и проверяется каждый раз при вводе или изменении данных.

•Свойство Сообщение об ошибке предназначено для задания сообщения, которое будет выводиться каждый раз, когда пользователь попытается ввести в таблицу данные, не соответствующие условию свойства Условие на значение.

• Свойство Фильтр может содержать определение фильтра, накладываемого на таблицу сразу после ее открытия.

• Свойство Порядок сортировки позволяет задать порядок сортировки записей таблицы после ее открытия.

Еще одно важное диалоговое окно Access 97, доступное в режиме конструктора таблиц, окно Индексы, позволяет указать первичный ключ таблицы и набор индексов для сортировки данных.

Каждое поле таблицы тоже обладает набором свойств. Первые четыре из них задаются в верхней части окна конструктора — в таблице с перечнем полей.

Ниже перечислены четыре основных свойства полей таблицы.

- Имя поля. Каждое поле таблицы обязательно должно иметь имя. Имена могут иметь длину до 64 символов, могут содержать пробелы (но не ведущие) и знаки пунктуации. Точки (.), восклицательные знаки (!) и квадратные скобки ([]) в именах полей не допускаются. Имена полей в таблице должны быть уникальными: два поля одной таблицы не могут иметь одинаковые имена. Хорошей практикой в программировании считается не включать в имена полей пробелов. Их обычно заменяют символами подчеркивания (\_) или используют прописные и заглавные буквы, чтобы длинные имена полей легче читались.

- Тип данных. Тип данных каждого поля тоже нужно указать обязательно. Происходит это в результате выбора значения второго столбца таблицы полей из раскрывающегося списка. Перечень допустимых типов данных фиксирован и включает следующие варианты: Текстовый, Поле MEMO, Числовой, Дата/время, Денежный, Счетчик, Логический, Поле объекта OLE, Гиперссылка, Мастер подстановок. (Тип данных Гиперссылка впервые появился в Access 97.)

- Описание. Значение этого свойства задавать не обязательно. Введенный здесь текст будет появляться в строке состояния, после выбора пользователем соответствующего поля в режиме таблицы.

- Ключевое поле. Чтобы задать свойство Ключевое поле, выделите поля, которые вместе будут составлять первичный ключ, и щелкните на кнопке Ключевое поле панели инструментов. Слева от названий выбранных полей появятся изображения ключа.

Кроме этих четырех, у каждого поля есть еще ряд дополнительных свойств, они перечислены в нижней части окна конструктора. Некоторые из этих дополнительных свойств зависят от типа данных выбранного поля. Их значения или вводятся вручную, или выбираются из раскрывающихся списков. (Во вкладке Подстановка для полей подстановок можно указать, какие элементы управления будут соответствовать этим полям в формах, — поля, списки или поля со списками.) Вкладка Общие позволяет указать значения следующих свойств полей таблиц.

• *Размер поля*. Для текстовых полей это свойство определят их длину и вводится вручную. Размер числовых полей выбирается из раскрывающегося списка. У полей типа Поле MEMO, Дата/время, Денежный, Логический, Поле объекта OLE, Гиперссылка, Мастер подстановок этого свойства нет.

• *Формат поля*. Это свойство определяет, в каком формате данные поля будут выводиться на экран. Значение свойства можно выбрать из стандартного набора форматов. Можно задать и свой собственный формат, введя его в текстовое поле. У полей типа Поле объекта OLE этого свойства нет.

• *Число десятичных знаков*. Для этого свойства числовых полей и полей счётчика можно выбрать из раскрывающегося списка значение Авто или фиксированное количество знаков дробной части, а можно ввести значение и вручную. Поля других типов этого свойства не имеют. От указанного здесь значения, как и от значения свойства Формат поля, зависит то, как данные будут отображаться на экране (на сами данные значение этого свойства не влияет).

• *Маска ввода*. Значением этого свойства может быть символьная строка, подобная той, которая определяет формат поля. Только маска ввода определяет, как данные будут вводиться или редактироваться. Для создания масок ввода полей типа Текстовый, Числовой, Дата/время и Денежный есть специальный мастер, которого можно запустить с помощью кнопки с троеточием (она появится справа, если щелкнуть мышкой на нужном свойстве). Этот мастер предложит вам целый список готовых масок ввода, подходящих, например, для ввода номеров телефона, времени, дат, индексов и т.п.

• *Подпись*. Если вы хотите, чтобы в режиме таблицы в качестве заголовка поля выводилось не его имя, а более удобная или понятная надпись, именно сюда ее и нужно ввести. Для подписей никаких ограничений не существует — они могут содержать любые символы, включая точки, восклицательные знаки и квадратные скобки.

• *Значение по умолчанию*. Значение, которое вы введете для этого свойства, Access будет автоматически вносить в указанное поле каждой добавляемой в таблицу записи. Очень часто в качестве значения по умолчанию для полей типа Дата/время используетсятекущая дата. Для полей типа Поле объекта OLE и Счетчик значений по умолчанию быть не может.

-*Условие на значение*. Здесь вводится условие (в виде выражения Ассеss), которое будет проверяться каждый раз, когда значение поля будет изменяться пользователем. Свойство Условие на значение у полей типа Поле MEMO, Поле объекта OLE и Счетчик отсутствует.

Сообщение об ошибке. Если введенное в таблицу значение не соответствует условию проверки (определенному в свойстве Условие на значение), пользователь увидит в строке состояния указанное здесь сообщение.

• *Обязательное поле*. Если для этого свойства задать значение Да, пользователь обязательно должен ввести значение этого поля для каждой добавляемой им в таблицу записи. Такая установка эквивалентна заданию для свойства Условие на значение значения Is Not Null. Для полей, являющихся частью первичного ключа, задавать это свойство не обязательно — пустые значения (Null -значения) в них не допускаются по определению.

• *Пустые строки*. Это свойство есть только у текстовых, МЕМО-полей и гиперссылок. Его значение Нет означает, что в поле должен быть введен хотя бы один символ. В Ассе ss между пустой строкой и значением поля Null существует определенная разница, поэтому свойства Пустые строки и Обязательное поле используются независимо.

• *Индексированное поле*. Свойство определяет индекс, создаваемый по одному полю. Это свойство может иметь одно из трех значений, выбираемых из раскрывающегося списка: Нет, Да (допускаются совпадения), Да (совпадения не допускаются). Последнее означает, что никакие две записи таблицы не могут иметь одинаковые значения данного поля. Для удаления индекса, созданного по одному полю, достаточно свойству Индексированное поле присвоить значение Нет. Нельзя индексировать таблицы по полям типа Поле MEMO, Поле объекта OLE и Гиперссылка.

• *Новые значения*. Это свойство есть только у полей-счетчиков. Оно определяет способ вычисления новых значений для добавляемых в таблицу записей: последовательные (максимальное из значений поля +1) или случайные числа. Случайные значения счетчика, как правило, используются при репликации данных, чтобы при добавлении записей в реплики одной и той же таблицы значения их ключевых полей не совпадали.

Итак, чтобы добавить в базу данных первую таблицу, нужно ввести имена ее полей, указать их типы и размеры, а также формат отображения данных.

Одним из главных принципов построения реляционных баз данных является то, что в каждом поле таблицы должны содержаться данные только одного типа. Для данных разных типов нужно создавать отдельные поля. При добавлении в таблицу новых полей Ассеss автоматически назначает их текстовыми. Если поле должно быть другим, разработчик базы данных выбирает тип его данных из раскрывающегося списка. Тип данных поля должен быть определен обязательно. Кроме того, можно определить его размер, формат отображения данных на экране и другие характеристики поля.

*Межтабличные связи, ключевые поля*

Последнее, что нужно сделать, прежде чем начать ввод данных в новую таблицу, — определить связи таблицы. Личные дела с другими таблицами базы данных, а также назначить ключевое поле и создать индексы для сортировки и поиска данных.

*Создание межтабличных связей*

От того, как новая таблица связана с остальными, зависит, какое из ее полей следует назначить первичным ключом. В теории СУБД известны четыре варианта связей между двумя таблицами (традиционно называемых отношениями). Все их мы рассматриваем с точки зрения того, как новая таблица связана с одной из существующих (как часто говорят, мы смотрим на связь со стороны новой таблицы):

• *Отношение один-к-одному*. При таком отношении каждой записи первой таблицы соответствует не более одной записи второй таблицы, и наоборот. Соответствие записей устанавливается в результате поиска в поле, являющемся первичным ключом одной из таблиц, значения поля, называемого внешним ключом второй таблицы. В этом случае ключевое поле новой таблицы, в котором ищутся соответствующие значения поля существующей таблицы, не должно содержать повторяющихся значений. Можно сказать, что две таблицы, связанные отношением один-к-одному, представляют собой одну таблицу, разделенную надвое вертикальной чертой, так что часть полей оказалась в одной таблице, а часть — в другой.

• *Отношение многие-к-одному*. При таком отношении любой записи той таблицы базы данных, связь с которой мы рассматриваем, может соответствовать любое количество записей созданной нами новой таблицы, но не наоборот. В этом случае ключевое поле новой таблицы, по которому устанавливается соответствие записей, будет внешним ключом, повторяющиеся значения в нем допускаются.

Этот тип отношений наиболее распространен. Именно возможность создавать межтабличные связи типа многие-к-одному и отличает реляционные СУБД от обычных процессоров плоских файлов.

• *Отношение один-ко-многим*. В этом случае первичный ключ новой таблицы (поле, содержащее уникальные значения) связывается с внешним ключом старой (значения поля могут повторяться). При этом каждой записи новой таблицы может соответствовать несколько записей существующей. Фактически это отношение в точности такое же, как отношение многие-к-одному. Можно сказать, что отношение многие-к-одному — это отношение один-ко-многим наоборот.

Отношение многие-ко-многим. В этом последнем случае каждой записи одной из таблиц может соответствовать любое количество записей другой таблицы, и наоборот. Соответственно поля обеих таблиц, по которым осуществляется связь, являются внешними ключами и могут содержать повторяющиеся значения.

Итак, все эти четыре вида отношений говорят о том, как новая таблица связана с существующей.

Access требует, чтобы те поля двух таблиц, по которым они связываются между собой, имели одинаковые типы данных. В случае числовых полей это требование означает, что их подтипы, т.е. значения свойства Размер поля, должны совпадать. А вот текстовые поля разной длины связывать можно. Хотя и не стоит, потому что запросы, основанные на таблицах, связанных по текстовым полям разной длины ведут себя довольно странно. Для того чтобы создавать связи между таблицами было легко и удобно, Access предлагает специальный режим: в окне Схема данных вы найдете наглядное графическое представление межтабличных связей. Чтобы перейти в этот режим и установить связь между двумя таблицами, выполните перечисленные ниже действия.

1. Закройте таблицу Клиент (с помощью кнопки закрытия окна). Если открыта таблица Объем заказа, закройте и ее. Изменять связи между открытыми таблицами нельзя.

2.Чтобы на панели инструментов присутствовала кнопка Схема данных, окно базы данных должно быть активным. Если это не так, активизируйте его с помощью кнопки панели инструментов или команды меню Окно1. В списке меню Окно может присутствовать до девяти названий открытых окон объектов базы данных, но окно базы данных всегда стоит первым.

3. Щелкните на кнопке панели инструментов Схема данных или выберите команду Сервис, потом Схема данных. Откроется окно Схема данных базы данных Анкетный опрос, в котором будут представлены все межтабличные связи этой базы данных.

4.Теперь щелкните на кнопке панели инструментов Добавить таблицу или выберите команду Связи, потом Добавить таблицу из меню Access. Перед вами появится диалоговое окно Добавление таблицы.

5.Выберите из списка предложенных таблиц Клиент и дважды щелкните мышью или с помощью кнопки Добавить добавьте ее в окно Схема данных. Щелкните на кнопке Закрыть.

6.Связь между таблицами Объемом заказа и Клиент основывается на значениях их полей Название организации. Щелкните мышью на поле Название организации таблицы Клиент и перетащите его на поле Название организации Объем заказа. Перед вами появится диалоговое окно Связи.

7. В диалоговом окне Связи есть очень полезная опция — Обеспечение целостности данных. Если установить соответствующий флажок, то при вводе пользователем в таблицу Клиент значения поля Название организации, Access проверит, есть ли в таблице Объем заказа запись с таким кодом (названием). Если такой записи не окажется, Access выдаст предупреждающее сообщение и откажется сохранять в таблице Клиент записи.. Такая процедура называется обеспечением (или поддержкой) целостности данных. Подробнее о целостности данных рассказывается в следующем разделе.

8. Щелкните на кнопке Создать, чтобы новая связь появилась в окне Схема данных.

9. Теперь щелкните на кнопке закрытия окна Схема данных и подтвердите сохранение изменений.

Созданная межтабличная связь будет использоваться при создании запросов, форм и отчетов.

*Обеспечение целостности данных*

Способность Access автоматически поддерживать целостность данных исключительно важна. Немногие из настольных реляционных СУБД умеют это делать. Сохранение целостности данных позволяет избежать создания в связанной таблице записей-сирот, не имеющих связи с главной таблицей. Например, если названиям организаций в таблице Клиент присвоены коды от 1 до 9, запись таблицы Арендная плата со значением 10 будет считаться сиротой, она не относится ни к одному из существующих сотрудников. Однако когда в таблицу Клиент будет добавлена новая запись с кодом 10, запись-сирота автоматически окажется связанной с ней, в результате чего в новой анкете не появится явно неправильная информация.

*Способы обеспечения целостности данных*

Чтобы обеспечить целостность данных, нужно сделать так, чтобы данные главной таблицы, с которыми связана информация других таблиц, не были изменены или удалены. Если вы попытаетесь удалить запись из таблицы Клиент, Access не позволит этого сделать. Программа предупредит вас о том, что прежде чем удалять запись главной таблицы, нужно удалить все связанные с ней записи подчиненных таблиц. Точно также в таблице Арендная плата не удастся изменить значение поля название организации. Значение поля типа Счетчик вообще нельзя изменять, но даже если бы поле название организации имело другой тип, Access все равно не позволила бы внести изменения и выдала бы предупреждающее сообщение.

Аналогичным образом, при попытке ввести в поле название организации таблицы Клиент код несуществующей организации, Access выдаст предупреждающее сообщение и откажется сохранить неверные данные. Благодаря этому разработчику базы данных не нужно прибегать к свойству поля таблицы Условие на значение, чтобы обеспечить целостность данных. При сохранении новой или измененной записи таблицы Access автоматически проверит значения критических полей.

***Выбор первичного ключа***

Таблицам, которые никогда не используются в качестве главных таблиц связи, первичный ключ не нужен. Главная таблица обычно содержит информацию, описывающую объекты реального мира, такие как сотрудники, клиенты или заказы. При этом для описания одного объекта используется одна-единственная запись. В этом смысле таблица Клиент может быть главной. Однако в нашей базе данных ни для одной межтабличной связи таблица Клиент главной не будет. Использование первичного ключа является простейшим способом предотвращения дублирования записей таблицы. Access требует, чтобы при создании межтабличной связи типа один-к-одному или при одновременном изменении записей нескольких таблиц первичный ключ был определен обязательно.

Хотя в Access это и не является обязательным требованием, главная таблица должна иметь первичный ключ. При его наличии пользователь просто не сможет ввести в таблицу записи с одинаковыми значениями ключевого поля. Часто связанные таблицы тоже имеют первичные ключи, хотя это и не обязательно. Предыдущие версии Access рассматривали таблицу без первичного ключа как нечто странное, и каждый раз при выходе из режима конструктора предлагали создать для таблицы первичный ключ. Access делает это только один раз.

Первичный ключ может состоять и из нескольких полей, тогда он называется составным ключом.

*Изменение полей и межтабличных связей*

Очень часто в процессе разработки базы данных обнаруживается, что нужно поменять местами поля таблицы, изменить их типы или межтабличные связи. Это одна из причин, по которой готовую базу данных нужно сначала протестировать, введя в нее некоторое количество пробных записей, проверить, все ли правильно и удобно ли будет пользователю работать, и только когда все необходимые изменения внесены, запускать базу данных.

Изменять можно практически все: форматы полей, их типы и размеры, условия проверки вводимых значений и т.д. Для этого нужно просто переключиться в режим конструктора и внести необходимые изменения свойств полей. Однако нужно иметь в виду, что изменения типов полей могут повлечь за собой потерю данных, поэтому, прежде чем предпринимать подобные действия с таблицами, содержащими нужную информацию, следует внимательно прочитать соответствующий раздел этой главы. Не менее разрушительным может оказаться и изменение межтабличных связей, поэтому этот вопрос тоже требует предварительного детального изучения.

*Изменение порядка полей таблицы*

Бывает, что создав таблицу и открыв ее в режиме таблицы, разработчик решает, что порядок ее полей не удобен для ввода и просмотра данных. В этом случае поля можно поменять местами, причем сделать это можно с помощью мыши. Затем можно решить, сохранить сделанные изменения навсегда или аннулировать их после закрытия таблицы. Итак, изменить положение поля в таблице можно сделать следующим образом.

1. Откройте таблицу в режиме таблицы. Изменение порядка следования полей — это единственное изменение структуры таблицы, которое можно выполнить в этом режиме.

2. Щелкните на кнопке с именем того поля, которое нужно переместить. При этом столбец поля окажется выделенным целиком.

3. Укажите курсором мыши на заголовок поля и нажмите левую кнопку мыши — вид курсора изменится, указывая на то, что выполняется операция перетаскивания объекта. На новую позицию левого края поля указывает при перетаскивании толстая вертикальная черта.

4. Переместите вертикальную линию в нужное положение поля и отпустите левую кнопку мыши. Поле будет перемещено.

5. Закрывая таблицу Клиент, вы получите обычный запрос на подтверждение сохранения изменений. Если поля должны остаться там, куда вы их переместили, подтвердите сохранение.

Порядок следования полей можно изменять и в режиме конструктора. Процедура будет похожей: щелкните на кнопке выбора поля, выделив его, а потом перетащите в новое положение. Изменение порядка следования полей никак не затрагивает их свойств.

***Применение бланка запроса***

Запросы — важнейший инструмент любой системы управления базами данных. Запросы служат для выборки записей, обновления таблиц и включения в них новых записей. Чаще всего запросы применяют для выборки конкретных групп записей, удовлетворяющих заданному условию. Кроме того, запросы позволяют комбинировать информацию, содержащуюся в различных таблицах, обеспечивая связанным элементам данных таблиц унифицированный вид. В данной главе рассматривается создание запросов, а также обсуждается задание в запросе условий отбора данных и применение результатов запроса.

*Что такое запрос*

Бланк запроса, называвшийся в ранних версиях бланком запроса по образцу (Query By Example — QBE), первоначально был создан, чтобы позволить пользователям компьютерных приложений баз данных, незнакомых с языками программирования, находить и отображать требуемые данные. Бланки запроса в той или иной форме применяются во многих системах управления базами данных.

По мере развития компьютерной техники совершенствовалось и средство пользовательского интерфейса, называемое графическим бланком запроса. Графический бланк запроса содержит отображение имен полей одной или нескольких таблиц базы данных в виде заголовков столбцов. Чтобы создать запрос, пользователи могут вводить в эти столбцы составные компоненты, или выражения. Поскольку в те времена, когда разрабатывался графический бланк запроса, редко использовались терминалы, которые могли работать в подлинном графическом режиме, применялся стандартный текстовый режим, позволяющий отобразить 25 строк по 80 символов. Применение графического бланка запроса принесло успех приложению Paradox RDBMS фирмы Ansa Software на рынке настольных баз данных RDBMS (система управления рв-щюнной базой данных) для персональных компьютеров, на котором в то время преобладали приложения dBASE II и dBASE III.

Windows предлагает пользователям графические средства, идеально подходящие для применения графического бланка запроса в системах управления базами данных.

*Создание первого реального запроса*

К данным, содержащимся в таблицах, применялась сортировка в заданном порядке и фильтрование по указанному условию отбора. Однако диалоговое окно запроса в режиме конструктора обеспечивает большую гибкость, чем окно фильтра, поскольку позволяет выбирать поля, появляющиеся в итоговом наборе данных запроса. Кроме того, объединяя основную и связанную таблицы, можно создавать более сложные запросы.

В данной главе рассматривается применение бланка запроса.

Чтобы создать простой запрос:

1. Щелкните на кнопке со стрелкой возле кнопки Новый объект панели инструментов База данных и выберите в раскрывающемся списке команду Запрос. Можно также щелкнуть во вкладке Запросы окна базы данных, затем — на кнопке Создать. Откроется диалоговое окно Новый запрос. Здесь можно выбрать самостоятельное создание запроса (в режиме Конструктор) либо применить один из мастеров создания специализированных запросов. Выделите пункт Конструктор и щелкните на кнопке ОК, чтобы отобразить окно запроса в режиме конструктора.

Поверх окна запроса в режиме конструктора появится окно Добавление таблицы. Вкладки диалогового окна Добавление таблицы позволяют выбрать все существующие таблицы или запросы, или их комбинацию. Новый запрос можно создать на основе одной или нескольких созданных таблиц или запросов.

2. В данном примере для создания запроса применяются только таблицы, поэтому оставайтесь в открывшейся по умолчанию вкладке Таблицы. Выделите таблицу Арендная плата и щелкните на кнопке Добавить. Чтобы включить таблицу в запрос, можно также дважды щелкнуть на ней. Запрос можно создать на основе нескольких таблиц, выбирая в списке имена этих таблиц и повторно щелкая на кнопке Добавить. Однако этот пример ограничивается одной таблицей. Щелкните на кнопке Закрыть. Диалоговое окно Добавление таблицы закроется.

В верхней области окна Добавление таблицы отобразится список полей таблицы Арендная плата, а в нижней области окна — пустой бланк запроса. Список полей содержит имена всех полей таблицы Арендная плата.

По умолчанию в окне запроса в режиме конструктора создается запрос на выборку, возвращающий отобранные данные из таблиц или итоговый набор данных других запросов на выборку. Также по умолчанию в этом режиме первому создаваемому (но еще не сохранявшемуся) запросу Access присваивает имя Запрос 1. Имя запросу можно присвоить при сохранении. Для последующих запросов, после сохранения первого из них, Access предлагает использовать для сохранения стандартные имена Запрос 2, Запрос 3 и т.д.

*Выбор полей для запроса*

После выбора таблицы в диалоговом окне Добавление таблицы, выберите поля, включаемые в запрос. Создаваемый запрос предназначен для создания списка предпочтений покупателей, в него следует включить поля, необходимые для записи персональных мнений.

В первой строке любого графического бланка запроса содержатся имена всех полей, включенных в запрос. (Иногда имена полей называют заголовками полей, или названиями полей.) Следовательно, в первой строке создаваемого образца запроса должны находиться имена всех полей, составляющих почтовый адрес.

Чтобы выбрать поля для бланка запроса, выполните следующие действия.

1. После открытия диалогового окна запроса в режиме конструктора, курсор помещается в первом столбце строки Поле. Щелкните на кнопке раскрывающегося списка в правом углу этой ячейки или нажмите клавишу F4 чтобы открыть список имен полей.

2. Щелкните на поле Название организации, чтобы выбрать его в качестве первого заголовка поля в запросе, либо выделите имя этого поля и нажмите клавишу <Еп1ег>. Список полей в нижней области окна закроется.

3. Переместите курсор во второй столбец, нажав клавишу *<→>* или <ТаЬ>. (Обратите внимание, кнопка раскрывающегося списка также перейдет во второй столбец.) Щелкните дважды на в списке имен полей таблицы, чтобы использовать Название как второе поле запроса. Дважды щелкнуть на имени поля в списке полей таблицы — еще один способ добавления полей в запрос Access.

В Access существует и третий способ добавления полей в запрос: метод ''перетащить и опустить". Чтобы применить его для вставки полей Название организации, прежде всего выделите эти поля в верхней части окна запроса. В списке полей таблицы Арендная плата щелкните на поле Арендная плата, затем нажмите клавишу <С1г1> или <Shift> и, удерживая ее нажатой, выделите поля Название организации и Вид автомата. Можно также после выделения поля Арендная плата, нажав клавишу < Shift>, дважды нажать клавишу < ↓ >. Теперь поля выделены.

5. Поместите указатель мыши на выделенные поля, щелкните и удерживайте нажатой левую кнопку мыши. Указатель мыши превратится в символ, условно представляющий имена трех выбранных полей. Перетащите его к третьему столбцу строки Поле бланка запроса и отпустите кнопку мыши.

Access добавит имена трех полей в запрос, начиная от столбца, на котором была отпущена кнопка мыши.

6. Теперь в бланке запроса отображаются несколько столбцов стандартной ширины. В данном запросе применяется семь полей, поэтому необходимо перетащить.границы окна запроса, чтобы отобразить дополнительно два пустых поля. Для уменьшения ширины столбцов можно переместить влево разделители полей. Чтобы отобразить два пустых поля, щелкните дважды на кнопке горизонтальной прокрутки вправо либо перетащите вправо бегунок полосы прокрутки.

7. Щелкните на кнопке Вид панели инструментов, чтобы перейти в режим выполнения запроса. Подождите некоторое время, пока Access выполнит запрос, основанный на таблице. Чтобы запустить запрос, можно также щелкнуть на кнопке Запуск панели инструментов Конструктор запросов.

Поскольку в бланке запроса в строку Условия отбора не введено никакого условия отбора, результат выполнения запроса в таблице Колбасные предпочтения отображает все записи.

***Создание и применение форм***

В Access формы служат для организации пользовательского интерфейса с таблицами данных. Многие задачи можно решить, просмотрев таблицы и результаты запросов в режиме таблицы. Например, можно расположить поля формы так, чтобы ввод и редактирование записи выполнялся слева направо и сверху вниз. Формы позволяют создать средства многовариантного выбора значений полей, в которых для представления допустимых значений применяются сокращенные кодовые обозначения. Правильно сконструированная форма ускоряет ввод данных и сводит до.минимума ошибки ввода оператора.

Формы создаются из набора отдельных конструктивных элементов, называемых элементами управления, или объектами элементов управления. Элементы управления — это компоненты экрана и диалоговых окон как Access, так и прочих приложений Windows. Для ввода и редактирования данных применяют поля, для наименования полей применяют надписи (также называемые подписями), а для вставки рисунков — рамки объектов. В окне формы размещают элементы управления двух типов: для отображения данных таблиц и для отображения статических данных — подписей и логотипов.

*Типы форм*

Содержимое и внешний вид формы зависит от ее назначения в конкретном приложении баз данных. Существует три типа приложений баз данных.

Приложения для обработки транзакций. Они предназначены для добавления в таблицы новых записей и редактирования существующих. Эти приложения требуют доступа (разрешения) на запись в таблицы, связанные с формами.

• Приложения для поддержки принятия решений. В таких приложениях информация предоставлена в виде графиков, таблиц или отдельных элементов данных. Возможность ввода или редактирования данных отсутствует. Эти приложения требуют доступ только для чтения таблиц, связанных с формами.

• Приложения поддержки баз данных. Они предназначены для выполнения административных функций, таких как создание баз данных или таблиц баз данных, контроль доступа к базе данных, защита данных с помощью пароля, периодическое уплотнение и копирование базы данных. Приложения поддержки баз данных требуют полного доступа ко всем объектам базы данных.

Формы — основные элементы приложений для обработки транзакций и поддержки принятия решений, описываемых в следующих разделах. Обычные операции поддержки баз данных не требуют применения форм, однако формы удобно использовать при вводе и просмотре записей.

*Создание главной формы с помощью мастера форм*

Создать форму и подчиненную форму проще всего с помощью мастера форм Access. Он позволяет создавать формы (с подчиненными формами и без них), содержащие поля из одной или нескольких таблиц или запросов. Мастер форм создает основную структуру формы и добавляет элементы управления Поле для отображения и редактирования значений элементов данных.

Чтобы создать форму Клиент с помощью мастера форм, выполните следующие действия.

1. Выберите вкладку Формы окна базы данных, затем щелкните на кнопке Создать. Отобразится диалоговое окно Новая форма.

2. Выберите Мастер форм в списке диалогового окна Новая форма. Мастер форм позволяет создать как простую форму (без подчиненной формы), так и форму с данными, содержащую подчиненную форму. После выбора опции Конструктор открывается пустая форма в режиме конструктора, а после выбора одной из опций Автоформа— автоматически создаются формы с особой структурой: в столбец, ленточная и табличная. Щелкнув на опции Диаграмма, вы запустите мастер диаграмм, позволяющего вставить в форму диаграмму или график. Наконец, с помощью опции Сводная таблица можно ускорить создание формы на основе сводных таблиц Excel.

3. В раскрывающемся списке в нижней части диалогового окна Новая форма перечисляются все существующие таблицы или запросы, доступные в качестве источника данных формы. Щелкните на стрелке, чтобы открыть список, затем выберите таблицу Клиент. Щелкните на кнопке ОК. Access отобразит первое диалоговое окно мастера Создание форм.

4. Выделите Название организации в списке Доступные поля, затем щелкните на кнопке с символом >, чтобы перенести поле Номер анкеты из списка Доступные поля в список Выбранные поля. Для переноса поля можно также дважды щелкнуть на имени этого поля.

Повторите эти действия для полей Адрес, Телефон, ФИО КЛ таблицы Клиент. (Включение этих полей в форму позволяет редактировать данные этих полей в таблице Клиент.)

5. Откройте раскрывающийся список Таблицы/запросы и выберите таблицу Клиент. В списке Доступные поля отобразятся доступные поля таблицы Клиент.

6. Щелкните на кнопке », чтобы включить все поля списка Клиент в список Выбранные поля. (Таким образом вы будете иметь возможность редактировать все поля таблицы Клиент.)

7. Поскольку поле Название организации таблицы Арендная плата включено в список Выбранные поля, нет необходимости включать в форму поле Название организации таблицы Клиент. Выделите поле Название организации в списке Выбранные поля, затем щелкните на кнопке <, чтобы убрать это поле из списка Выбранные поля и поместить обратно в список Доступные поля. Наконец, щелкните на кнопке Далее, чтобы отобразить второе диалоговое окно мастера форм.

8. Поскольку поля, выбранные для отображения с помощью формы, находятся в разных таблицах, мастер форм предлагает выбрать тип представления данных.

9. Теперь Мастер форм предлагает выбрать внешний вид подчиненной формы. Выберите опцию ленточный. В результате подчиненная форма будет отображать данные таблицы Клиент в ленточном формате, подобном отображению в режиме таблицы, однако допускающем изменение форматирования (цвета, заголовков столбцов и т.д.). Щелкните на кнопке Далее, чтобы перейти в следующее диалоговое окно Создание форм.

10.Мастер форм отображает диалоговое окно выбора стиля отображения формы. Создаваемая форма применяется для ввода данных — в ней не требуется специальных эффектов выделения или оформления полей. Поэтому щелкните на опции Обычный, а затем — на кнопке Далее, чтобы перейти в последнее диалоговое окно мастера.

Главную форму, созданную мастером форм, можно уже использовать. Однако как главная, так и подчиненная формы, нуждаются в доработке. В последующих примерах этой главы показано, как модифицировать формы, создаваемые мастером форм. Полученные навыки редактирования форм можно применить при создании собственных форм "с нуля" (как это сделать, вы узнаете в следующей главе). Независимо от опыта создания форм Access, применение мастера для создания основного макета формы значительно экономит время.

*Применение конструктора формы*

Чтобы модифицировать макет новой формы, щелкните на кнопке Вид панели инструментов. Откроется окно конструктора формы. Перемещаемая Панель элементов позволяет вставлять в форму новые элементы управления.

*Элементы окна конструктора форм*

Форма в окне конструктора состоит из трех частей: Заголовок формы, Область данных и Примечание формы. Применение колонтитулов (с помощью команды Вид Колонтитулы) не обязательно. В окне конструктора форм расположены следующие элементы.

• Панель инструментов Конструктор форм с кнопками быстрого доступа к командам меню в режиме конструктора.

• Панель инструментов Формат (форма/отчет) с кнопками опций форматирования цветом, шрифтами, границами и т.д.

• Вертикальная и горизонтальная линейки помогают определять размер и положение объектов формы. Линейки калиброваны в сантиметрах (в версиях Access для стран, применяющих метрическую систему измерений).

• Вертикальную линию, указывающую положение правого поля формы, можно переместить в любое нужное место.

• Горизонтальную линию, указывающую положение нижнего поля формы, можноподчиненной формы, чтобы вместить ее в прямоугольник заданного размера в главной форме.

• Полосы вертикальной и горизонтальной прокрутки позволяют просматривать части формы, выходящие за пределы экрана.

• Полоса раздела Заголовок формы определяет высоту раздела заголовка формы. Она применяется только в случае, если в форму добавляют заголовок и примечание. Раздел заголовка формы содержит не изменяющийся текст, графические изображения и другие элементы управления, отображаемые в верхней части формы. В многостраничной форме заголовок отображается только на первой странице.

• Полоса Область данных отделяет заголовок от остального содержимого формы. В области данных находятся элементы управления формы, отображающие данные таблиц и запросов, а также статические элементы, такие как надписи и логотипы.

• Раздел Примечание формы по функциям подобен разделу Заголовок формы. При выводе на печать многостраничной формы Примечание формы печатается только внизу последней страницы.

***Печать отчетов***

Отчет — конечный продукт большинства приложений баз данных. Отчет Access — это ленточная форма особого рода, сконструированная специально для печати. Для его создания Access комбинирует данные таблиц, запросов и даже форм. Отчет предназначен для печати и широкого распространения (это может быть и печатная версия формы). Такие отчеты часто применяют для поддержки принятия решений. Напечатав ленточную форму, можно создать отчет, отображающий все или часть значений полей таблицы или запроса.

В данной главе обсуждается создание относительно простых отчетов с помощью мастера отчетов. Кроме того, рассматриваются методы модификации конструкции отчетов, созданных мастером, для более полного соответствия требованиям пользователя.

*Отчеты и формы: различие и сходство*

Большинство методов создания форм для обработки транзакций применимо и к отчетам. Ниже перечислены принципиальные различия между формами и отчетами.

• Отчеты предназначены только для печати, но не отображения данных на экране. В окне предварительного просмотра содержимое отчета отображается в неудобном для чтения виде. В увеличенном же масштабе можно просмотреть только часть отчета.

• В отличие от форм, данные, на базе которых создан отчет, невозможно изменять с помощью элементов управления, установленных на панели элементов. В отчетах Access не разрешает пользователю вводить данные с помощью переключателей, флажков и т.д. Однако эти элементы управления можно применять для указания состояния Да/Нет переключателей, флажков и полей со значениями, производными от списков, предназначенных для многовариантного выбора.

• Отчеты невозможно просматривать в режиме таблицы. Доступен только предварительный просмотр, перед печатью и режим конструктора.

• Можно создать свободный отчет, не связанный с источником данных. Свободные отчеты служат "контейнерами" индивидуальных подчиненных отчетов, использующих собственные источники данных.

• С помощью специальных диалоговых элементов управления, служащих для установки параметра принтера, можно задать минимально допустимые левые, правые, верхние и нижние печатные пол отчета. Если отчет меньше ширины листа бумаги, положение правого поля определяется конструкцией отчета. Передвигая печатные поля вправо от отображаемого левого поля, можно установил ширину левого поля, превышающую стандартную.

• В многостолбцовых отчетах число столбцов, ширина столбцов и расстояние между ними задаются установками параметров принтера, но не элементами управления или установками свойств в режиме конструктора.

С другой стороны, характеристики отчетов и форм Access имеют много общего.

• Мастера отчетов. С их помощью могут создаваться три основных типа отчетов: в столбец, ленточный и почтовые наклейки. Отчеты, созданные мастерами, можно

Разделы отчета. Они включают в себя заголовок, примечание и колонтитулы. Заголовок и примечание печатаются в начале и конце отчета, колонтитулы — на каждой странице. Раздел примечаний отчета нередко используют для печати итоговых значений. Разделы отчета соответствуют разделам формы с теми же именами.

Разделы групп отчета. В целом, они аналогичны разделам данных формы. Группу часто называют группировкой, а процесс создания группы записей — группированием. Для печати промежуточных итогов в группировки отчетов можно включать заголовки и примечания. В разделы заголовков и примечаний групп вставляются статические (свободные) изображения, а в разделы групп — присоединенные изображения.

Подчиненные отчеты. Такие отчеты включают в отчеты точно так же, как в главные формы включают элементы управления Подчиненная форма/отчет.

*Типы отчетов Access*

Отчеты Access подразделяются на шесть категорий (макетов).

• Одностолбцовые отчеты. В них в одном длинном столбце полей перечисляются значения каждого столбца в каждой записи таблицы или запроса. Надписи указывают имя столбца, а в полях справа от надписей приведены значения. Средство Асеева Автоотчет: в столбец позволяет создать отчет, состоящий из одного столбца, с помощью кнопки Автоотчет панели инструментов. Одностолбцовые отчеты применяют редко: их распечатка требует большого расхода бумаги.

• Ленточные отчеты. В таких отчетах в озаглавленном столбце расположено каждое поле таблицы или запроса, а каждая запись помещается в отдельную строку. При печати каждое значение записи располагается в строке под заголовком соответствующего столбца. Если все столбцы не помещаются на одну страницу, печатаются дополнительные страницы, пока не отобразятся все столбцы. Затем печатается следующая группа записей

• Многостолбцовые отчеты. Они создаются из одностолбцовых отчетов. Информация, не поместившаяся в первый столбец, переходит в верхнюю часть второго столбца и т.д. В таком формате при печати отчета затрачивается меньше бумаги. Однако применение таких отчетов ограничено, поскольку результат выравнивания столбцов, как правило, не всегда отвечает требованиям пользователя.

• Отчеты с группированием данных. Самый популярный тип отчетов. Такие отчеты позволяют вычислять итоговые значения для групп и представляют информацию в удобном для использования виде. В конце отчета отображаются итоги.

• Почтовые наклейки. Особый тип многостолбцовых отчетов, создаваемых для печати имен и адресов (или других данных, расположенных в нескольких столбцах) в группах. Каждая группа полей заполняет ячейку сетки. Конструкция наклейки зависит от числа столбцов и строк на странице.

• Свободные отчеты. Такие отчеты состоят из подчиненных отчетов. При этом каждый подчиненный отчет создается на основе независимых источников данных, например, таблиц или запросов.

В первых четырех типах отчетов в качестве источника данных применяют таблицу или запрос формам Отчеты такого рода называют присоединенными к источнику данных. В свободном отчете главный отчет не связан с источником данных — таблицей или запросом. Однако подчиненные запросы, содержащиеся в свободных отчетах, должны быть связаны с источником данных.

*Использование мастера диаграмм для создания графика*

Использование мастера диаграмм значительно упрощает процесс создания графика. Этот инструмент можно применять для создания двух различных классов графиков и диаграмм.

• На несвязанном графике для каждой строки запроса строится отдельная линия. Типами несвязанных диаграмм являются гистограммы и диаграммы с областями.

• Связанные график или диаграмма строятся только для текущей записи формы, на которой они размещаются, и отображают значения всего одной строки запроса или таблицы.

Чтобы построить несвязанный график на основании данных, отобранных с помощью построенного ранее запроса Мастер Диаграмм, выполните следующие действия.

1.Создайте новую форму, выберите в верхнем списке пункт Диаграмма, a в раскрывающемся списке — запрос Мастер Диаграмм. Щелкните на кнопку ОК, чтобы открыть первое диалоговое окно для создания диаграммы.

2.Щелкните на кнопке » для добавления всех трех полей в диаграмму. Щелкните накнопку Далее, чтобы открыть следующее окно.

3.Щелкните на кнопку График, расположенной в третьем ряду набора кнопок с типам\* диаграмм. После этого щелкните на кнопку Далее для перехода к следующему окну.

4.Программа создания диаграмм попытается создать перекрестный запрос, основанный на типах данных из набора вашего запроса. В данном случае эта программа допустит ошибку, предположив, что вы собираетесь отобразить месяцы в окне легенды и типы продуктов вдоль горизонтальной оси диаграммы.

5. Нам нужно, чтобы в окне легенды были отражены категории, а месяцы отобразились вдоль горизонтальной оси. Перетащите кнопку Категория с правой стороны окна и опустите ее на место легенды. После этого точно таким же образом переместите кнопку Дата Исполнения под горизонтальную ось. На ней появится новый (немного урезанный) заголовок. Дважды щелкните на этой кнопке, откроется диалоговое окно Группировка, в котором вы можете выбрать любой интервал времени д.р. Выберите и щелкните на кнопке ОК. Для перехода к последнему окну мастера создания диаграмм щелкните на кнопке Далее.

6. В окне задания имени для диаграммы введите Сделки. Если имеется необходимость в отображении легенды, то установите опцию Да. Изменять установленные по умолчанию значения других опций не стоит.

7. Щелкните на кнопке Готово для отображения графика в форме. В появившейся диаграмме сложно что-то понять, да и наличие легенды создает видимость беспорядка. Ниже описывается, как устранить эти проблемы.

8. Щелкните на кнопке Вид, чтобы перейти в режим конструктора, и увеличьте размеры

9. Присвойте свойству Доступ свободной рамки объекта значение Да, а свойству Блокировка — Нет.

После завершения процесса создания структуры формы установите свойство Доступ в Нет, чтобы другие пользователи вашего приложения не смогли активизировать диаграмму и изменить ее структуру.

**4. Область применения ACCESS**

Разработанная мною база данных может использоваться маркетологами предприятий, обслуживающих и устанавливающих автоматы по оплате платежей через Интернет. Построив запрос можно проследить своевременность выплат и сроки продления договоров о взаимном сотрудничестве, а значит правильно проводить сбытовую политику. Построив зарос можно определить где чаще всего происходит поломка автомата и принять правильное решение по устранению неполадок, может полной замене автомата, что позволит грамотно вести пост-обслуживание клиентов. Запрос на «услуги автомата» позволит определить какой процент с каждой сделки имеет фирма в определенной Т.Т. и своевременно расширить возможные услуги в автомате, что обеспечит эффективное продвижение товара и стабильное ведение финансовой отчетности.

Разработанная база данных позволяет задавать различные запросы, помогающие маркетологам правильно планировать маркетинговую стратегию, товарную, ценовую, рекламную политику, а так же оценивать эффективность уже проведённых мероприятий и вносить коррективы.

Для того чтобы воспользоваться базой данных, нужно:

1.Войти в область Access.

2.Выбрать нужную база данных.

3.Перейти в область запроса и запросить данные, которые нам нужны для получения результата.

4.Перейти в область форм и создать форму отчёта.

5.Создать отчёт и получить результат, на основе которого делаются выводы (основываясь на данных, которые указываются в запросе).

**Выводы**

На основе полученных результатов, представленных в отчётах, маркетолог может сделать выводы, необходимые ему для принятия правильного решения относительно внедрения и правильного позиционирования товара на рынке, ведения товарной, сбытовой и рекламной политики, проведения маркетинговых мероприятий и анализа их эффективности и д.р. Так, например, из отчёта по объему заказа можно проследить какое количество товара было куплено различными предприятиями и какие тенденции можно из этого выявить. Отчёт по договорам позволяет сделать выводы о том, что, такие компании как: «Каре» и «Снегирь» являются постоянными задолжниками, при этом объем заказа у них достаточно велик и выплаты производятся с процентами по неустойке, поэтому отказываться от сотрудничества с ними пока не стоит. Отчет о клиентах позволяет менеджеру с наибольшей эффективностью работать с организациями-заказчиками, поскольку данная схема данных показывает все контакты с фирмой. Так как в Access можно задать любой отчёт, то с помощью получаемых результатов можно предугадать даже малейшие нюансы маркетинговой стратегии.

С помощью этой работы Вы познакомились с Access и теми её уникальными возможностями, благодаря которым она является самой лёгкой в использовании среди всех других систем управления базами данных.

Из этой работы Вы узнали: как с помощью мастера баз данных и стандартных шаблонов Access создать собственную базу данных, что такое стандартные объекты Access, такие как таблицы, формы, отчёты и модули; как с помощью панели инструментов и меню выполнять различные задачи; как выбрать оптимальный тип данных; как добавлять в таблицы новые записи, вводить и редактировать данные.

В работе рассказывается о мощном механизме отбора и обработки данных - запросах - и об организации межтабличных связей, являющихся основой реляционных СУБД. Мы узнали, как создаются самые простые запросы с помощью конструктора запросов; как отбирать нужные поля и описывать результирующие наборы записей; как объединять таблицы в мощные перекрестные запросы, суммирующие данные таблиц и представляющие результаты в очень наглядном и компактном виде.

Часть работы посвящена изучению объектов интерфейса Access. Благодаря формам с разнообразными элементами Ваши базы данных «оживут», а мощные генераторы позволят распечатывать профессионально оформленные отчёты. Их можно будет сохранять и в файлах, чтобы потом продолжить работу с ними в Excel или Word. Вы узнали о том, как с помощью мастеров избежать рутинной работы по созданию простейших основных и подчинённых форм; как работать с мастерами создания отчётов.

**Список используемой литературы**

1. Роджер Дженигс. Access 97. - М.: 1998

2. Диго СМ. Проектирование баз данных. - М.: «финансы и статистика», 1988

3. Морозов В.,Шафрин Ю. Курс компьютерной технологии. - М.: «АБФ», 1998

4. Информационные технологии в маркетинге. - М.: «Юнити», 2000

**Приложение**

Таблица №1

|  |  |  |
| --- | --- | --- |
| Договора | | |
| Название организации | Номер договора | Адрес |
| ООО "Амега" | 56041 | г. Исса, ул. Совхозная д 7б |
| ООО "Изумруд" | 56042 | г. Кузнецк, ул. Пролетарская д. 43а |
| ООО "Рафэл" | 56044 | г. Пенза, ул. Измайлова д.26 |
| ООО "Каре" | 56043 | г. Пенза, ул. Свердлова д. 1 |
| ООО "Снегирь" | 56045 | г. Сердобск, ул. Мира д. 2 |

Таблица №2

|  |  |
| --- | --- |
| Задолжность | |
| Название организации | Сумма задолжности |
| ООО "Амега" | 1000 |
| ООО "Изумруд" | 500 |
| ООО "Каре" | 2900 |
| ООО "Рафэл" | - |
| ООО "Снегирь" | 1400 |

Таблица №3

|  |  |  |  |
| --- | --- | --- | --- |
| Клиент | | | |
| Название организации | Адрес | ФИО директора | контакты |
| ООО "Амега" | г. Исса, ул. Совхозная д 7б | Мичурин Алексей Витальевич | (842)437095 |
| ООО "Изумруд" | г. Кузнецк, ул. Пролетарская д. 43а | Синицын Евгений Петрович | (845)345674 |
| ООО "Каре" | г. Пенза, ул. Свердлова д. 1 | Степанов Георгий Семенович | (8412)437613 |
| ООО "Рафэл" | г. Пенза, ул. Измайлова д.26 | Балашов Иван Васильевич | (8412)554378 |
| ООО "Снегирь" | г. Сердобск, ул. Мира д. 2 | Макарова Надежда Ивановна | (843)344906 |

Таблица №4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Объем заказа | | | | |
| Название организации | Срорк отгрузки | Срок приема | Выплата | Объем заказа |
| ООО "Амега" | 01.11.07 | 01.11. 07 | 5000 | 40 |
| ООО "Изумруд" | 02.11.07 | 03.11.07 | 4700 | 38 |
| ООО "Каре" | 03.11.07 | 03.11.07 | 1000 | 10 |
| ООО "Рафэл" | 04.11.07 | 05.11.07 | 6700 | 54 |
| ООО "Снегирь" | 05.11.07 | 06.11.07 | 4645 | 35 |

Запрос №1

|  |  |  |  |
| --- | --- | --- | --- |
| Задолжности | | | |
| Название организации | Сумма задолжности | Номер договора | Адрес |
| ООО "Рафэл" | - | 56044 | г. Пенза, ул. Измайлова д.26 |
| ООО "Изумруд" | 500 | 56042 | г. Кузнецк, ул. Пролетарская д. 43а |
| ООО "Каре" | 2900 | 56043 | г. Пенза, ул. Свердлова д. 1 |
| ООО "Снегирь" | 1400 | 56045 | г. Сердобск, ул. Мира д. 2 |
| ООО "Амега" | 1000 | 56041 | г. Исса, ул. Совхозная д 7б |

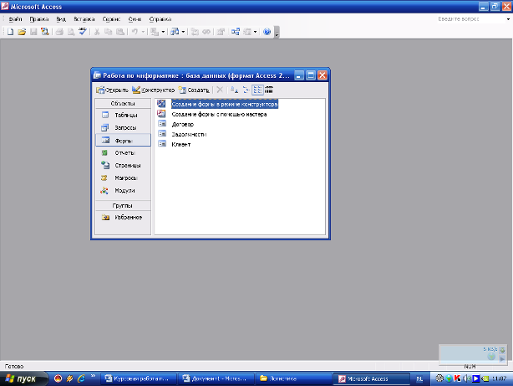
Запрос №2

|  |  |  |
| --- | --- | --- |
| Задолжность 1000руб | | |
| Название организации | Сумма задолжности | Номер договора |
| ООО "Амега" | 1000 | 56041 |

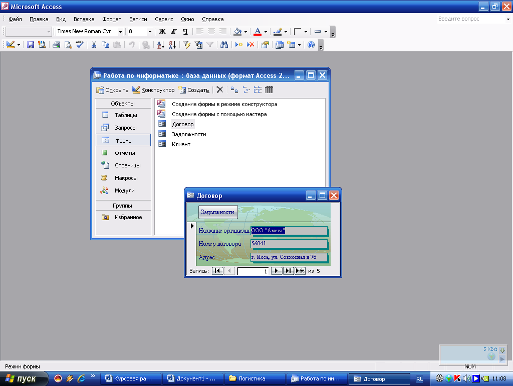
Запрос №3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Объем заказов | | | | |
| Название организации | Срорк отгрузки | Объем заказа | ФИО директора | контакты |
| ООО "Амега" | 01.11.07 | 40 | Мичурин Алексей Витальевич | (842)437095 |
| ООО "Изумруд" | 02.11.07 | 38 | Синицын Евгений Петрович | (845)345674 |
| ООО "Каре" | 03.11.07 | 10 | Степанов Георгий Семенович | (8412)437613 |
| ООО "Рафэл" | 04.11.07 | 54 | Балашов Иван Васильевич | (8412)554378 |
| ООО "Снегирь" | 05.11.07 | 35 | Макарова Надежда Ивановна | (843)344906 |

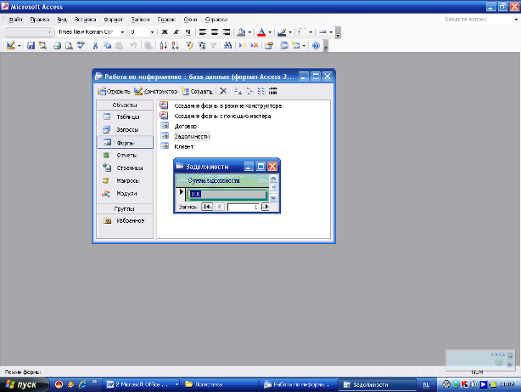
Форма



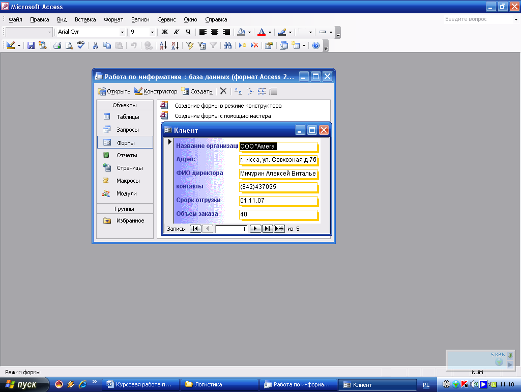
Форма №1



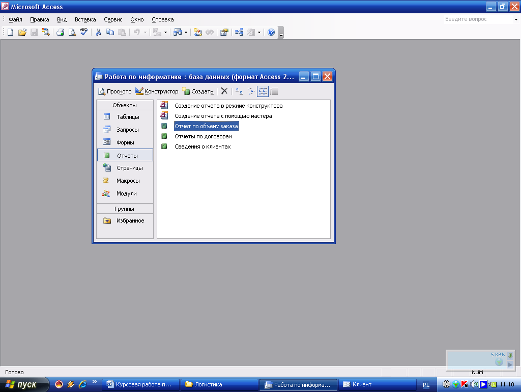
Форма №2



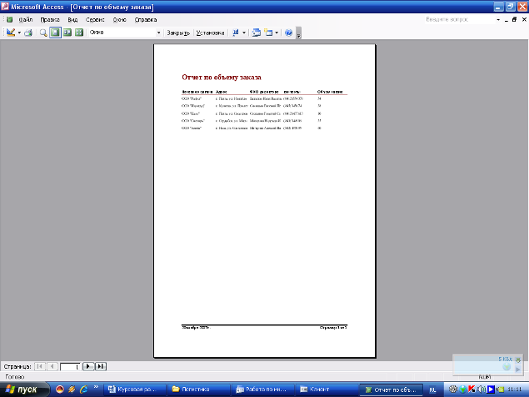
Форма №3



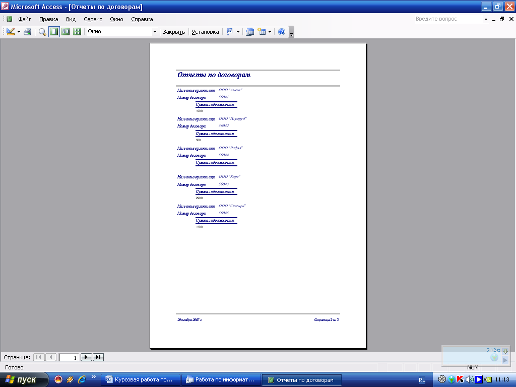
Отчеты



Отчет №1



Отчет №2



Отчет №3

