Министерство Образования и Науки РФ

Федеральное агентство по образованию

ГОУ ВПО Череповецкий Государственный Университет

Инженерно-экономический институт

Кафедра: ПО ЭВМ

Курсовая работа

Прикладные аспекты информационных технологий

Дисциплина: Информатика

Выполнил: студент гр. 3СМ-11

Фомичев Д.И.

Проверила: преподаватель

Майтама Е.В.

Череповец

2009 г.

Задание на курсовую работу

Разработать реляционную базу данных продаж и товаров компьютерного магазина средствами СУБД Access. База данных должна состоять не менее, чем из четырех таблиц данных, форм для ввода и просмотра данных, не менее шести запросов на выборку данных, отчетов о проделанной работе.

Оглавление

Введение

1. Теоретическая часть СУБД Access

1.1 Базы данных

1.1.1 История развития

1.1.2 Понятия о базах данных и системах управления ими

1.1.3 Классификация баз данных

1.1.4 Инфологическая модель базы данных, ее виды, проектирование

1.2 СУБД Access 2007

1.2.1 Основные сведения

1.2.2 Основные функции

1.2.3 Элементы базы данных

1.2.4 Таблицы

1.2.5 Запросы

1.2.6 Формы

1.2.7 Отчёты

1.2.8 Макросы

2. Практическая часть

2.1 Состав базы данных

2.2 Описание выполнения работы

2.2.1 Таблицы

2.2.2 Схема данных

2.2.3 Запросы

2.2.4 Формы

2.2.5 Отчёты

Заключение

Список использованной литературы

Введение

база данный реляционный запрос макрос

Основой для учета, контроля и планирования служат всевозможные картотеки, регистрационные журналы, списки и т.д. Они постепенно накапливаются и обновляются. При большом объеме информации поиск и обобщение необходимых сведений, осуществляемых вручную, представляют собой довольно трудоемкий процесс.

С появлением ЭВМ и использованием их для обработки информации появилась возможность автоматизировать решение многих информационно - справочных и расчетных задач.

Первоначально для накопления и хранения информации на ЭВМ применялись локальные массивы (или файлы), при этом для каждой из решаемых функциональных задач создавались собственные файлы исходной и результатной информации. Это приводило к значительному дублированию данных, усложняло их обновление, затрудняло решение взаимосвязанных проблемных задач.

Постепенно с развитием программного обеспечения ЭВМ появились идеи создания управляющих систем, которые позволяли бы накапливать, хранить и обновлять взаимосвязанные данные по целому комплексу решаемых задач, например при автоматизации бухгалтерского учета на предприятии. Эти идеи нашли свое воплощение в системах управления базами данных (СУБД). СУБД взаимодействуют не с локальными, а взаимосвязанными по информации массивами, называемыми базами данных. С появлением персональных компьютеров СУБД становятся наиболее популярным средством обработки табличной информации. Они являются инструментальным средством проектирования банков данных при обработке больших объемов информации.

Программное обеспечение для работы с базами данных используется на персональных компьютерах уже довольно давно. К сожалению, эти программы либо были элементарными диспетчерами хранения данных и не имели средств разработки приложений, либо были настолько сложны и трудны, что даже хорошо разбирающиеся в компьютерах люди избегали работать с ними до тех пор, пока не получали полных, ориентированных на пользователя приложений.

Microsoft Access - это функционально полная реляционная СУБД. В ней предусмотрены все необходимые вам средства для определения и обработки данных, а также для управления ими при работе с большими объемами информации. Что касается легкости использования, то Microsoft Access совершил здесь настоящий переворот, и многие для создания своих собственных баз данных и приложений обращаются именно к нему.

Система управления базами данных предоставляет вам возможность контролировать задание структуры и описание своих данных, работу с ними и организацию коллективного пользования этой информацией. СУБД также существенно увеличивает возможности и облегчает каталогизацию и ведение больших объемов хранящейся в многочисленных таблицах информации. СУБД включает в себя три основных типа функций: определение (задание структуры и описание) данных, обработка данных и управление данными. Все эти функциональные возможности в полной мере реализованы в Microsoft Access. В практике, как правило, необходимо решать и задачи с использованием электронных таблиц и текстовых процессоров. Например, после подсчета или анализа данных необходимо их представить в виде определенной формы или шаблоны. В итоге пользователю приходится комбинировать программные продукты для получения необходимого результата. В этом смысле все существенно упростят возможности, предоставляемые Microsoft Access. Предлагается рассмотреть эти возможности.

# **1. Теоретическая часть** СУБД Access

## **Базы данных**

### История развития

В истории создания автоматизированных информационных систем относительно независимо развивались два направления:

1. разработка автоматизированных информационных систем (АИС) как автоматизированных систем управления (АСУ);

2. разработка автоматизированных систем научно-технической информации (АСНТИ).

Работы по их созданию начались практически одновременно в 60-е гг.

Первое направление - разработка АИС и АСУ - было инициировано научно - техническим прогрессом и возникшими в связи с этим проблемами организационного управления (рост количества информации, трудности с её обработкой "вручную").

Зарубежная практика шла по пути разработки отдельных программных процедур, например, для бухгалтерии, учета материальных ценностей, и основные работы проводились в направлении исследования и совершенствования возможностей вычислительной техники, разработки средств, обеспечивающих наиболее рациональную организацию информационных массивов, удобный для пользователя интерфейс, наращивание памяти ЭВМ.

В нашей стране проблема обеспечения информацией управленческих работников была поставлена сразу системно. Была разработана классификация АСУ, в которой прежде всего выделялись АСУ разных уровней системы управления - для уровня предприятий и организаций, отраслевые, республиканские и региональные и общегосударственная автоматизированная система Аналогично на уровне предприятий, и особенно создаваемых в 70-е гг. научно-производственных объединений (НПО), в структуре АСУП (или интегрированных АСУ объединений) выделялись уровни (страты) - АСУ объединения, АСУ предприятий и организаций (научно-исследовательских институтов, конструкторских бюро), входящих в НПО, АСУ производств, комплексов цехов, АСУ цехов и участков. На уровне цехов и участков АСУ вначале разделялись на АСУ технологическими процессами, АСУ технической и технологической подготовки производства, АСУ организацией производства.

Работы по созданию централизованных общегосударственных АСУ и АСНТИ были приостановлены в связи с преобразованиями 1991 г. Однако, при переходе к рыночной экономике, к правовому государству возрастает роль еще одного важного вида информации - нормативно-правовой и нормативно-методической, регламентирующей деятельность предприятий при предоставлении им большей самостоятельности и сокращении организационно-распорядительной документации (текущих приказов и распоряжений, ревизующих командно-административные методы управления).

В дальнейшем, по мере развития предприятий и их АСУ, особенно в условиях предоставления большей самостоятельности производствам и цехам и перераспределению управленческих функций между администрацией предприятия и руководителями производств и цехов, также стало более удобным представлять структуру АСУ в виде многоуровневой, стратифицированной. Разделение АСУ на функциональную и обеспечивающую части, а последней - на информационное обеспечение, техническое, организационное, программное и другие виды обеспечения позволило привлечь для уточнения соответствующих видов обеспечения специалистов в этих областях. Такой подход к организации разработок АСУ помог справиться со сложностью системы и ускорить разработку АСУ путем параллельного проведения работ по анализу и выбору структуры отдельных видов обеспечения. Однако если разрабатывать отдельные проекты, то после разработки возникает достаточно сложная задача их согласования, взаимоувязки принятых структур этих видов обеспечения, критериев, учитываемых при их разработке и. Поэтому на определенном этапе развития работ по созданию АСУ был даже сформулирован специальный принцип единства информационного обеспечения, технического и программного, как основных видов обеспечения.

В настоящее время существует огромное количество готовых программных продуктов. Поэтому, нет необходимости при создании на предприятии автоматизированной системы заниматься самостоятельной разработкой программного обеспечения.

### **Понятия о базах данных и системах управления ими**

Базы данных являются одним из основных компонентов современных информационных систем. Информационная система — это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации.

Цель любой информационной системы — обработка информации конкретной предметной области.

Под предметной областью понимается совокупность связанных между собой функций, задач управления в некоторой области деятельности предприятия, с помощью которых достигается выполнение поставленной цели.

База данных - это приложение, обеспечивающее создание, хранение, обновление и поиск информации в базе данных, а также управление безопасностью и целостностью данных. В простейшем случае - это программа, которая обеспечивает работу с информацией. Информация в базе может быть любая, при этом, как правило, работа с информацией подразумевает следующие действия:

* добавление новой информации;
* поиск информации;
* изменение информации;
* удаление информации из базы данных;

Особенностями такой совокупности данных являются:

1. достаточно большие объемы информации;
2. максимально возможная компактность хранения данных;
3. возможность извлечения из базы данных разнообразной информации в определенной предметной области;
4. удобные для пользователя вид и форма извлекаемой информации;
5. высокая скорость доступа к данным;
6. надежность хранения информации и возможность Предоставления санкционированного доступа к данным, для отдельных пользователей;
7. удобство и простота конструирования пользователем запросов, форм и отчетов для выборки данных.

Мир программных систем, позволяющих использовать базы данных, довольно многообразен. В настоящее время существует достаточно большое количество программных систем, позволяющих создавать и использовать локальные и удаленные базы данных. Среди наиболее известных можно отметить Paradox, dВase, FoxPro, MS Access, InterBase, Oracle, Infomix, MS SQL Server и другие.

В состав среды Borland Delphi входят компоненты, позволяющие создавать программы работы с файлами данных, созданных различными системами: от dBase до Infomix и Oracle. Среда разработки позволяет программисту с помощью утилиты Database Explorer создавать файлы баз данных в различных форматах.

### Классификация баз данных

Базы данных могут быть локальные и удаленные. Отличия в типах систем состоит в расположении программы, использующей данные, самих данных, а также способа разделения данных между несколькими пользователями.

Локальная база данных. Данные локальной базы данных (файлы данных) находятся на одном (локальном) устройстве, в качестве которого может выступать диск компьютера или диск другого компьютера, работающего в сети.

Для обеспечения разграниченного доступа к данным между несколькими пользователями в локальных базах данных применяется метод, получивший название блокировка файлов. Метод блокировки заключается в том, что пока данные используются одним пользователем, другой пользователь не может работать с этими данными. В качестве примера локальных баз данных можно привести программные системы Paradox, dBase, FoxPro и MS Access.

Удаленная база данных. Данные (файлы) удаленной базы данных находятся на удаленном компьютере. При этом не следует путать сетевые диски компьютеров, работающих в сети, и каталоги удаленного компьютера.

Программа работы с удаленной базой данных состоит из двух частей: клиентской и серверной (такая технология получила название "клиентсервер"). Клиентская часть программы, работающая на компьютере пользователя, обеспечивает взаимодействие с серверной программой с помощью запросов, передаваемых на удалённый компьютер.

Серверная часть программы, работающая на удалённом компьютере, принимает запросы, выполняет их и пересылает данные клиентской программе. Запросы представляют собой команды на специальном языке структурированных запросов (Structured Query Language, SQL).

Архитектура клиент-сервер обладает рядом преимуществ:

* обеспечивается более широкий доступ к существующим базам данных;
* повышается общая производительность системы: поскольку клиенты и серверы находятся на разных компьютерах, их процессоры способны выполнять приложения параллельно. Настройка производительности компьютера с сервером упрощается, если на нем выполняется только работа с базой данных;
* снижается стоимость аппаратного обеспечения; достаточно мощный компьютер с большим устройством хранения нужен только серверу – для хранения и управления базой данных;
* сокращаются коммуникационные расходы. Приложения выполняют часть операций на клиентских компьютерах и посылают через сеть только запросы к базам данных, что позволяет значительно сократить объем пересылаемых по сети данных;
* повышается уровень непротиворечивости данных. Сервер может самостоятельно управлять проверкой целостности данных, поскольку лишь на нем определяются и проверяются все ограничения. При этом каждому приложению не придется выполнять собственную проверку;
* архитектура клиент-сервер естественно отображается на архитектуру открытых систем.

### **Инфологическая модель базы данных, ее виды, проектирование**

Основой базы данных является модель данных. Информационно-логическая (мифологическая) модель предметной области отражает предметную область в виде совокупности информационных объектов и их структурных связей.

**Информационный объект** — это описание некоторой сущности (явления, реального объекта, процесса) в виде совокупности логически связанных реквизитов. Информационный объект имеет множество реализаций — экземпляров.

Базы данных также могут быть иерархическими, сетевыми и реляционными.

**Иерархическая (древовидная) модель данных** представляет собой иерархию элементов, называемых узлами. Узел — это совокупность атрибутов данных, описывающих информационный объект. На самом верхнем уровне имеется только один узел — корень. Каждый узел кроме корня связан только с одним узлом на более высоком уровне, называемом исходным узлом для данного узла. Каждый узел может быть связан с одним или несколькими узлами более низкого уровня, называемыми порожденными (подчиненными). Узлы, не имеющие порожденных узлов, называются листьями. Количество деревьев в базе данных определяется числом корневых записей. К каждой записи базы данных существует только один путь.

**Сетевая модель** также основывается на понятиях узел, уровень, связь. Сетевая модель данных — это модель, в которой порожденный узел может иметь более одного исходного узла. В сетевой структуре любой элемент любого уровня может быть связан с любым другим элементом.

**Реляционная модель базы данных** состоит из одного или нескольких файлов, каждый из которых соответствует одной таблице.

Основная цель проектирования базы данных — это сокращение избыточности хранимых данных, а следовательно, экономия объема используемой памяти, уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте. При создании баз данных следует придерживаться методологии нормализации отношений.

Процесс проектирования информационных систем является достаточно сложной задачей. Он начинается с построения инфологической модели данных, т. е. идентификации сущностей. Затем необходимо выполнить следующие шаги процедуры проектирования даталогической модели, т. е. мифологическая модель должна быть: отображена в компьютероориентированную даталогическую модель, "понятную" СУБД.

* Представить предметную область в виде совокупности отдельных независимых друг от друга объектов, каждый из которых будет описываться своей таблицей.
* Для каждой таблицы определить ключевые поля; установить связи между таблицами; для каждой связи определить тип.
* Разработать структуру каждой таблицы: перечень полей, их типы и свойства.
* Заполнить таблицы данными.
* Разработать необходимые запросы к БД, входные и выходные формы и отчеты.
* Предусмотреть возможность автоматизации часто выполняемых действий путем создания макросов и программных модулей.

## **СУБД** Access 2007

### Основные сведения

Приложение Microsoft Access 2007 (далее Access) является мощной и высокопроизводительной 32-разрядной системой управления реляционной базой данных (далее СУБД).

База данных – это совокупность структурированных и взаимосвязанных данных и методов, обеспечивающих добавление выборку и отображение данных.

Реляционная база данных. Практически все СУБД позволяют добавлять новые данные в таблицы. С этой точки зрения СУБД не отличаются от программ электронных таблиц (Excel), которые могут эмулировать некоторые функции баз данных.

Существует три принципиальных отличия между СУБД и программами электронных таблиц:

1. СУБД разрабатываются с целью обеспечения эффективной обработки больших объёмов информации, намного больших, чем те, с которыми справляются электронные таблицы.
2. СУБД может легко связывать две таблицы так, что для пользователя они будут представляться одной таблицей. Реализовать такую возможность в электронных таблицах практически невозможно.
3. СУБД минимизируют общий объём базы данных. Для этого таблицы, содержащие повторяющиеся данные, разбиваются на несколько связанных таблиц.

Access – мощное приложение Windows. При этом производительность СУБД органично сочетаются со всеми удобствами и преимуществами Windows.

Как реляционная СУБД Access обеспечивает доступ ко всем типам данных и позволяет одновременно использовать несколько таблиц базы данных. Можно использовать таблицы, созданные в среде Paradox или dBase. Работая в среде Microsoft Office, пользователь получает в своё распоряжение полностью совместимые с Access текстовые документы (Word) , электронные таблицы (Excel), презентации (PowerPoint). С помощью новых расширений для Internet можно напрямую взаимодействовать с данными из World Wide Web и транслировать представление данных на языке HTML, обеспечивая работу с такими приложениями как Internet Explorer и Netscape Navigator.

Access специально спроектирован для создания многопользовательских приложений, где файлы базы данных являются разделяемыми ресурсами в сети. В Access реализована надёжная система защиты от несанкционированного доступа к файлам.

База данных храниться в одном файле, но профессиональные пользователи предпочитают разделять базу данных на два файла: в одном хранятся объекты данных (таблицы, запросы), в другом объекты приложения (формы, отчёты, макросы, модули).

В последних версиях Access представлен новый формат файла (MDE) –библиотеки, с помощью которого можно создавать приложения, не включая VBA- код.

Несмотря на то, что Access является мощной и сложной системой, его использование не сложно для непрофессиональных пользователей.

### Основные функции

1. Организация данных. Создание таблиц и управление ими.
2. Связывание таблиц и обеспечение доступа к данным. Access позволяет связывать таблицы по совпадающим значениям полей, с целью последующего соединения нескольких таблиц в одну.
3. Добавление и изменение данных. Эта функция требует разработки и реализации представленных данных, отличных от табличных (формы).
4. Представление данных. Access позволяет создавать различные отчёты на основе данных таблиц и других объектов базы данных.
5. Макросы. Использование макросов позволяет автоматизировать повторяющиеся операции. В последних версиях Access макросы используют для совместимости.
6. Модули. Модули представляют собой процедуру или функцию, написанные на

Access VBA (диалект Visual Basic Application). Эти процедуры можно использовать для сложных вычислений. Процедуры на Access VBA превышают возможности стандартных макросов.

Защита базы данных. Эти средства позволяют организовать работу приложения в многопользовательской среде и предотвратить несанкционированный доступ к базам данных.

Средства печати. С помощью этой функции Access позволяет распечатать практически всё, что можно увидеть в базе данных.

Access так же позволяет создавать дистрибутивные диски для распространения готового приложения (с помощью Office Developer Edition Tools). Распространение подразумевает поставку всех необходимых файлов на каком-либо носителе.

Пакет ODE Tools включает мастер установки, автоматизирующий создание средств распространения и программы установки. Он так же позволяет выполнение приложения на компьютерах, на которых не установлен Access.

### Элементы базы данных

1. Таблицы.
2. Запросы.
3. Формы
4. Отчёты.
5. Макросы
6. Модули.

База данных может содержать до 32768 объектов.

В состав Access входит множество мастеров, построителей и надстроек, которые позволяют упростить процесс создания объектов базы данных.

### Таблицы

1. Создание таблицы в режиме Мастера таблиц

Процесс создания объекта базы данных при помощи мастера делится на несколько шагов, на каждом из которых можно установить требуемые характеристики создаваемого объекта. Мастер таблиц позволяет создать новую таблицу на основе 33 образцов таблиц делового применения и 20 образцов таблиц личного применения (для английской версии Access соответственно 77 и 44).

1. Создание таблицы в режиме Конструктора таблиц

Если импортировать или связать данные нельзя, то чаще всего таблицы создаются в режиме конструктора, который позволяет определить требуемую структуру таблицы. В режиме конструктора таблиц можно увидеть названия, типы данных, описания назначения, а также некоторые дополнительные свойства полей таблицы.

1. Создание таблицы непосредственно в режиме таблицы

Access дает возможность создания таблиц непосредственно в режиме таблицы. При этом Access создает пустую таблицу на основе таблицы, используемой по умолчанию, со структурой, имеющей 20 полей и 30 пустых записей. После этого можно вводить данные прямо в таблицу. При ее сохранении Access анализирует данные и выбирает тот тип для каждого поля, который больше всего соответствует введенным данным. Создание таблиц в режиме таблицы имеет ограниченное применение. Поля таблицы не имеют содержательных имен, так что почти всегда нужно будет редактировать структуру, чтобы переименовать поля.

Кроме того, из примера Access не всегда может правильно определить тип данных. Таблицы, которые созданы в режиме таблицы, не могут включать объекты OLE и мемо-поля. Если есть желание иметь такие поля, вам нужно изменить структуру таблицы. При создании таблицы в режиме таблицы не происходит экономии времени по сравнению с традиционным способом определения полей и установки их свойств в структуре пустой таблицы.

Перед началом заполнения таблицы нужно определить связи этой таблицы с другими таблицами базы данных, определить ключевое поле и создать индексы.

Связь между таблицами устанавливает отношения между совпадающими значениями в ключевых полях - обычно между полями разных таблиц. В большинстве случаев связывают ключевое поле одной таблицы с соответствующим ему полем (часто имеющим то же имя), которое называют полем внешнего ключа во второй таблице. Таблица, содержащая ключевое поле, называется главной, а таблица, содержащая внешний ключ — связанной. Существует четыре вида отношений:

Один к одному. При использовании отношения "один к одному" запись в таблице "А" (главной таблице) может иметь не более одной связанной записи в таблице "В" (связанной таблице) и наоборот. Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу.

Связь с отношением "один к одному" используют для разделения очень широких таблиц или для отделения части таблицы по соображениям защиты. В таблицах связанных отношением "один к одному" ключевые поля должны быть уникальными.

Один ко многим. Связь с отношением "один ко многим" является наиболее часто используемым типом связи между таблицами. В такой связи каждой записи в таблице "А" могут соответствовать несколько записей в таблице "В", а запись в таблице "В" не может иметь более одной соответствующей ей записи в таблице "А". В таблице "А" ключевое поле должно быть уникальным. Связь с отношением "один ко многим" таблиц "А" и "В" можно рассматривать как связь с отношением "многие к одному" таблиц "В" и "А".

Многие к одному. При использовании отношения "многие к одному" одной записи в таблице "А" может соответствовать одна запись в таблице "B", а запись в таблице "В" может иметь более одной соответствующей ей записи в таблице "А". В таблице "А" ключевое поле должно быть уникальным. Связь с отношением "один ко многим" таблиц "А" и "В" можно рассматривать как связь с отношением "многие к одному" таблиц "В" и "А".

Многие ко многим. При использовании отношения "многие ко многим" одной записи в таблице "А" могут соответствовать несколько записей в таблице "В", а одной записи в таблице "В" несколько записей в таблице "А". В этом случае не требуется уникальности ключевых полей таблиц "А" и "В". Этот вид отношения используется для связывания таблиц, имеющих только внешние ключи.

Связанные поля не обязательно должны иметь одинаковые имена, однако, они должны иметь одинаковые типы данных. Кроме того, связываемые поля типа "Числовой" должны иметь одинаковые значения свойства "Размер поля". Исключениями из этого правила являются поля счетчика с последовательной нумерацией, которые могут связываться с числовыми полями размера "Длинное целое". Существует возможность создать связь между текстовыми полями разной длины, но это может вызвать сложности при создании запросов.

Одним из основных принципов разработки реляционных баз данных является то, что все данные, содержащиеся в поле таблицы, должны иметь один и тот же тип. Для каждого поля таблицы необходимо задать тип данных. По умолчанию используется тип данных "Текстовый". При задании типа данных поля можно также указать размер, формат и другие параметры, влияющие на отображение значения поля и точность числовых данных.

Основные типы данных:

1. Текстовый. Текст или числа, не требующие проведения расчётов.
2. МЕМО. Поле этого типа предназначено для хранения небольших текстовых данных (до 64000 символов). Поле этого типа не может быть ключевым или проиндексированным.
3. Числовой. Этот тип данных содержит множество подтипов. От выбора подтипа (размера) зависит точность вычислений.
4. Счётчик. Уникальные, последовательно возрастающие числа, автоматически вводящиеся при добавлении новой записи в таблицу.
5. Логический. Логические значения, а так же поля, которые могут содержать одно из двух возможных значений.
6. Денежный. Денежные значения и числовые данные, используемые в математических вычислениях.
7. Дата/Время. Дата и время хранятся в специальном фиксированном формате.
8. Поле объекта OLE. Включает звукозапись, рисунок и прочие типы данных. Поле этого типа не может быть ключевым или проиндексированным.
9. Гиперсвязь. Содержит адреса Web-страниц.

1.2.5 Запросы

Запросы являются важным инструментом в любых системах управления базами данных. Они используются для выделения, обновления и добавления новых записей в таблицы. Чаще всего запросы используются для выделения специфических групп записей, чтобы удовлетворить определенному критерию. Кроме того, их можно использовать для получения данных из различных таблиц, обеспечивая единое представление связанных элементов данных.

В Access существует четыре типа запросов для различных целей

Запросы на выборку отображают данные из одной или нескольких таблиц в виде таблицы.

Перекрестные запросы собирают данные из одной или нескольких таблиц в формате, похожем на формат электронной таблицы. Эти запросы используются для анализа данных и создания диаграмм, основанных на суммарных значениях числовых величин из некоторого множества записей.

Запросы на изменение используются для создания новых таблиц из результатов запроса и для внесения изменений в данные существующих таблиц. С их помощью можно добавлять или удалять записи из таблицы и изменять записи согласно выражениям, задаваемым в режиме конструктора запроса.

Запросы с параметрами - это такие запросы, свойства которых изменяются пользователем при каждом запуске. При запуске запроса с параметром появляется диалоговое окно, в котором нужно ввести условие отбора. Этот тип запроса не является обособленным, т. е. параметр можно добавить к запросу любого типа.

### 1.2.6 Формы

Формы Access позволяют создавать пользовательский интерфейс для таблиц базы данных. Хотя для выполнения тех же самых функций можно использовать режим таблицы, формы предоставляют преимущества для представления данных в упорядоченном и привлекательном виде. Формы позволяют также создавать списки значений для полей, в которых для представления множества допустимых значений используются коды. Правильно разработанная форма ускоряет процесс ввода данных и минимизирует ошибки.

Формы создаются из набора отдельных элементов управления: текстовые поля для ввода и редактирования данных, кнопки, флажки, переключатели, списки, метки полей, а также рамки объектов для отображения графики и объектов OLE. Форма состоит из окна, в котором размещаются два типа элементов управления: динамические (отображающие данные из таблиц), и статические (отображающие статические данные, такие, как метки и логотипы).

Формы Access являются многофункциональными; они позволяют выполнять задания, которые нельзя выполнить в режиме таблицы. Формы позволяют производить проверку корректности данных, содержащихся в таблице. Access позволяет создавать формы, включающие другие формы (форма внутри формы называется подчиненной). Формы позволяют вычислять значения и выводить на экран результат.

Приложения для работы с базами данных можно разделить на три основных категории:

* Управление транзакциями. Эти формы, работают непосредственно с таблицами, при условии, что активна только одна таблица
* Доступ к данным. Такие формы предназначены только для просмотра информации.
* Администрирование баз данных. Приложения этой категории выполняют административные функции такие, как создание баз данных или таблиц, разграничение прав доступа пользователей к объектам баз данных, обеспечение безопасности с помощью шифрования, периодическое уплотнение базы данных, а также операции резервного копирования.

Простейший путь создания основной и подчиненной форм – использование Мастера форм, который позволяет создавать формы (с подчиненными или без), содержащие поля из одной или более таблиц или запросов. Мастер форм создает базовый внешний вид формы и добавляет текстовые поля для отображения и редактирования значений полей таблиц. Независимо от того, являетесь ли вы профессиональным разработчиком приложений или начинающим пользователем, использование Мастера форм заметно упрощает и ускоряет процесс создания простых форм, которые затем можно усовершенствовать в режиме конструктора.

### 1.2.7 Отчёты

Конечным продуктом большинства приложений баз данных является отчет. Создаваемые Access отчеты делятся на шесть основных типов:

1. Отчеты в одну колонку представляют собой один длинный столбец текста, содержащий значения всех полей каждой записи таблицы или запроса. Надпись указывает имя, а справа от нее указывается значение поля. Новое средство
2. В ленточных отчетах для каждого поля таблицы или запроса выделяется столбец, а значения всех полей каждой записи выводятся по строчкам, каждое в своем столбце
3. Многоколоночные отчеты создаются из отчетов в одну колонку при использовании колонок "газетного" типа или колонок "змейкой", как это делается в настольных издательских системах и текстовых процессорах.

В основном отчёты проще всего построить при помощи Мастера отчётов. Мастер отчетов старается создать оптимальный вариант окончательного отчета с первой попытки. Обычно мастер в достаточной степени приближается к законченному варианту, так что тратиться намного меньше времени на редактирование базового отчета мастера, чем ушло бы на создание отчета из незаполненного шаблона.

1.2.8 Макросы

При работе с базой данных часто приходится снова и снова выполнять одни и те же задачи. Чтобы не повторять одни и те же действия, можно использовать макросы. База данных неуклонно разрастается по мере того, как туда вносятся новые записи в формы, создаются запросы и распечатываются новые отчеты. При этом разрабатываются и сохраняются для последующего использования новые объекты. Приходится регулярно выполнять некоторые задачи. Для выполнения таких регулярных задач удобно использовать макросы Access. После создания этих небольших программ можно запускать их автоматически при определенных действиях пользователя (например, при щелчке на кнопке или при открытии определенной формы). Для автоматического запуска макросов в Access используются так называемые события. Макросы Access позволяют автоматически повторять различные действия и тем самым избавляют от написания сложных программ и подпрограмм. Выполнение прикладной задачи можно представить как результат последовательного выполнения действий, каждое из которых может быть записано соответствующей командой на языках Visual Basic и VBA. Кроме того, с помощью макроса можно имитировать выбор команды меню или перемещение мыши. Макросы Access обычно не дублируют отдельных нажатий клавиш или перемещений мыши, выполняя особые, заданные пользователем задачи, например открытие формы или запуск отчета. Используя макросы для выполнения рутинных операций, в существенно экономит время и силы. Кроме того, поскольку всякий раз при запуске макроса будет осуществляться одна и та же последовательность действий, макрос сделает работу с базой данных более эффективной и аккуратной.

2. Практическая часть

##

## 2.1 Состав базы данных

В моей базе данных присутствуют:

Таблицы:

1. Номенклатура товара;
2. Описание товаров;
3. Поставщик;
4. Продажи;
5. Производитель;
6. Цена;

Запросы:

1. Запрос клиента;
2. Запрос на остаток в магазине
3. Запрос на остаток на складе
4. Запрос на продажу по дате
5. Запрос по продаже
6. Суммарный запрос по продажам

Формы:

1. Номенклатура
2. Поставщик
3. Продажи
4. Производитель
5. Товары

Отчёты:

1. Отчет по продажам
2. Товары
3. Поставщики

## 2.2 Описание выполнения работы

###

### 2.2.1 Таблицы

1. Номенклатура товара

Сначала в режиме конструктора я задал имена полей и тип данных в них:

Рис.1

Затем в режиме таблицы ввел записи:

Рис.2

1. Описание товаров

Таблица имеет следующие поля:

Рис.3

Тип данных столбцов "Общее количество" и "Остаток в магазине" я поменял на Числовой.

1. Поставщик

Создал в режиме конструктора таблицу и заполнил её данными.

Таблица имеет следующие поля:

Рис.4

1. Продажи

Я создал таблицу с полями:

Рис.5

И в полях "Кол-во проданных единиц товара" и "Дата продажи тип полей поменял соответственно на Числовой и Дата/Время.

1. Производитель

Создал таблицу:

Рис.6

Цена

В режиме конструктор создал таблицу с полями:

Рис.7

И тип поля "Цена товара" поменял на денежный.

###

### 2.2.2 Схема данных

Рис.8

Для создания выполним команду Работа с базами данных/Схема данных. В появившемся диалоговом окне мы добавим шесть таблиц "Номенклатура товара", "Описание товаров", "Поставщик", "Продажи", "Производитель", "Цена". Установим между ними связь "один-ко-многим".

2.2.3 Запросы

1. Запрос клиента

Рис.9

Я создавал в режиме конструктора, для этого выполнил команду

Создание/Конструктор запросов:

Рис.10

Добавил необходимые таблицы, выбрал нужные поля в них, в столбце Наименование товара ввел условие отбора [Введите наименование товара].

1. Запрос на остаток в магазине

Рис.11

В режиме конструктора добавил необходимые таблицы, выбрал нужные поля в них, в столбце Код товара ввел условие отбора [Введите код товара].

Рис.12

1. Запрос на остаток на складе

В режиме конструктора добавил необходимые таблицы, выбрал нужные поля в них.

Рис.13

Результат отбора - общее кол-во всех товаров на складе.

1. Запрос на продажу по дате

Рис.14

В режиме конструктора добавил необходимые таблицы, выбрал нужные поля в них, в столбце Дата продажи ввел условие отбора [Введите дату], в столбце Кол-во проданных единиц товара в строке Групповая операция выбрал Sum, а в условии отбора >0

Рис.15

1. Запрос по продаже

Рис.16

В режиме конструктора добавил необходимые таблицы, выбрал нужные поля в них, в столбце Наименование товара ввел условие отбора [Введите наименование товара], в столбце Кол-во проданных единиц товара в строке Групповая операция выбрал Sum, а в условии отбора >0 и в столбце Цена в строке Групповая операция выбрал тоже Sum.

Рис.17

1. Суммарный запрос по продажам

Рис.18

В режиме конструктора добавил необходимые таблицы, выбрал нужные поля в них, в столбце Кол-во проданных единиц товара в строке Групповая операция выбрал Sum, а в условии отбора >0 и в столбце Цена в строке Групповая операция выбрал тоже Sum.

###

### 2.2.4 Формы

Все формы создал, используя команду Создание/Форма. В появившемся диалоговом окне я выбирал пункты необходимые для данной формы, затем каждую корректировал в режиме макета или конструктора. Изменял стиль текста, фон, границы, Нажав на кнопку на панели инструментов Кнопка добавлял в формы кнопки для выхода, поиска записи, перехода к другой форме или запросу.

1. Номенклатура

Рис.20

Создана в режиме Форма,добавлена кнопка перехода к Запросу по продажам.

1. Поставщик

Рис.21

Создана в режиме Разделенная форма.

1. Продажи

Рис.22

Создана в режиме Несколько элементов, добавлена кнопка для перехода к форме Товары.

1. Производитель

Рис.23

Создана в режиме Форма,добавлена кнопка для выхода из формы.

1. Товары

Создана в режиме Разделенная форма, добавлена кнопка для поиска нужной записи.

Рис.24

### 2.2.5 Отчёты

Отчеты я создавал, выбрав необходимую таблицу и выполнив команду Создание/Отчет, затем дорабатывал в режиме Конструктора и Макета.

# Заключение

В рамках курсового проекта по дисциплине "Информатика" был разработан программный продукт прикладного уровня "Комплектующие компьютера и поставщики" при использовании программы Access.

В современных условиях руководителям предприятий, организаций приходиться иметь дело с таким большим количеством информации, она так быстро меняется, что её часто становится просто невозможно обрабатывать "вручную". Кроме того, на больших предприятиях с большими оборотами продукции существует необходимость учёта и контроля большого объёма финансовой, производственной, закупочно-сбытовой, маркетинговой информации.

И для этого и создаются автоматизированные системы для сбора, обработки и хранения информации. Такие информационные системы должны облегчить процесс работы с информацией, циркулирующей на предприятии.

В результате выполненной разработки можно сделать следующие выводы: при разработке программного обеспечения был пройден полный цикл проектирования программы от постановки задачи до введения выходного результата на исполнение и эксплуатацию.

Разработанная программная система позволяет упрощение процедуры поиска необходимой продукции для потенциальных клиентов, сокращение времени, которое они затрачивают при выборе подходящего во всех отношениях варианта комплектующих компьютера.

И поэтому суть разработанного данного проекта заключается в том, чтобы предоставить максимальный объем информации о предлагаемых к реализации комплектующих для покупателей и заказчиков. Также моя программа дает информацию о конкретно интересующем товаре, предоставляет его характеристики и возможности.

# Список использованной литературы

1. Кренке Д. Теория и практика построения баз данных. – 8-е изд. – СПб.: Питер, 2003. 800с.
2. Бакаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2002. – СПб.: БХВ-Петербург, 2002. – 720 с.
3. Microsoft Access 2000. Шаг за шагом: Практическое пособие / Пер. с англ. – М.: ЭКОМ, 2000. – 352 с.
4. Информатика. Базовый курс / Под ред. С.В. Симоновича.– СПб.: Питер, 2001. – 640 с.
5. Бойко В.В., Савинков В.М. "Проектирование информационной базы автоматизированной системы на основе СУБД" М.: Финансы и статистика, 1982.