Курсовая работа по дисциплине «Информационное обеспечение систем управления» на тему: «Проектирование базы данных и систем управления базой данных в среде Microsoft Access»

**СОДЕРЖАНИЕ**

Введение

1. Предпосылки создания БД

1.1 Цель создания базы данных

1.2 Планирование процесса разработки

1.3 Требования, предъявляемые к проектируемой СУБД

2. Проектирование БД в среде Microsoft Access

* 1. Основные сведения

2.2 Системные требования

2.3 Основные функции

2.4 Элементы базы данных

2.5 Основные типы данных

2.6 Панели и меню

2.7 Visual Basic for Application

3. Применение MS Access в ООО”Форум Компьютерс”

# 3.1Сопровождение баз данных

3.1.1 Необходимость модернизации БД

3.1.2 Модификация таблиц

3.1.3 Создание таблицы с помощью мастера

3.1.4 Использование конструктора для модернизации

таблиц

3.1.5.Связывание таблиц с БД

3.2 Построение запросов

## 3.3 Применение языка SQL в MS Access

3.3.1 Использование выражения SELECT

3.3.2 Фильтрация данных с помощью предложения

WHERE

3.3.3 Сортировка данных конструкции выражения

SELECT

3.3.4 Группировка столбцов

3.3.5 Использование предложения HAVING

3.3.6 Объединение таблиц

3.4 Анализ и оптимизация разработанной (модернизированной)

БД и объектов управления данными

3.5 Программирование управление структур свойствами макросов

3.5.1 Макросы. Общие сведения

3.5.2 Создание макросов

3.5.3 Создание групп макросов

3.5.4 Запуск макроса

Заключение

Список использованной литературы

**ВВЕДЕНИЕ**

база данный мicrosoft аccess запрос макрос

Компьютерный центр Форум работает на рынке компьютерного оборудования уже более 10 лет, начав свою деятельность как самостоятельное юридическое лицо в рамках холдинга, сферой деятельности которого является комплексная автоматизация предприятий «под ключ» и системная интеграция, в тесном сотрудничестве с «Бизнес Консалтинг Групп».

**НАПРАВЛЕНИЯ ТОРГОВОЙ И СЕРВИСНОЙ ДЕЯТЕЛЬНОСТИ**

На сегодняшний день компьютерный центр Форум успешно работает в различных сегментах компьютерного бизнеса, четко разграничивая специфику розничных, дилерских и корпоративных продаж и соответствующего обслуживания.

**ПРОИЗВОДСТВО**

Особое значение для компании имеет производство и продажа персональных компьютеров и серверов под маркой «FORUM».

Компания имеет все необходимые сертификаты на производство компьютеров, на сервисное и гарантийное обслуживание.

**ДОСТАВКА**

С целью повышения качества обслуживания и создания максимально удобных условия работы для дилеров и розничных покупателей, к их услугам создан транспортный отдел, укомплектованный грузовыми и легковыми автомобилями (для доставки габаритного оборудования) а также мобильная курьерская служба. (подробнее о доставке)

**РЕГИОНАЛЬНЫЕ ОТДЕЛЕНИЯ**

Постоянно находясь в центре событий и анализируя ситуацию на рынке высоких технологий, руководство компьютерного центра Форум прекрасно понимает необходимость дальнейшего развития бизнеса и не успокаивается на достигнутом, постепенно расширяя круг своих интересов. С этой целью развернута программа организации филиальной сети в различных городах России.

За короткий срок в рамках этой программы уже открыты и успешно работают филиалы компании в нескольких городах Российской Федерации

**НАШИ ПАРТНЕРЫ И СТАТУСЫ**

|  |  |
| --- | --- |
|  | Авторизованный дилер |
|  | Valued partner |
|  | Авторизованный дилер |
|  | Авторизованный дилер |
|  | Авторизованный дилер |
|  | Авторизованный дилер |
|  | Официальный дилер |
|  | Авторизованный дилер |
|  | Официальный дилер |
|  | Авторизованный дилер |
|  | Авторизованный дилер |
|  | Авторизованный дилер |

**НАШИ КЛИЕНТЫ**

Мы гордимся, что являемся поставщиками компьютеров и оргтехники в такие организации учреждения как:

* Автобанк
* Арбор Мунди - сеть парфюмерных супермаркетов
* Банк Кредитования Малого Бизнеса
* Газпромбанк
* ГосАтомНадзор
* МАИ
* МИИГА
* Министерство Обороны РФ
* Министерство Образования РФ
* Министерство Труда и Социального Развития
* МИСИ
* МИФИ
* Московский Государственный Университет
* МФТИ
* Национальный Банк Развития
* Пробизнесбанк
* Роспромбанк
* Русский Международный Банк
* Ряд предприятий Министерства Обороны
* Сбербанк РФ (Лефортово)
* Cеть магазинов «Мегалюкс»

**1. ПРЕДПОСЫЛКИ СОЗДАНИЯ БД**

**1.1 Цель создания базы данных**

В деловой или личной сфере часто приходится работать с данными из разных источников, каждый из которых связан с определенным видом деятельности.

Целью создания БД является разработка автоматизированной системы, обрабатывающей информацию с использованием программного обеспечения персонального компьютера. Что значительно облегчает работу любого предприятия.

В качестве примера возьмем организацию ООО”Форум Компьютерс” и рассмотрим базу данных применяемую на ней.

Менеджеры фирмы желают постоянно иметь под рукой данные о товаре,имеющемся в данный момент в наличии . При этом мы располагаем сведениями о товаре (минимальном наборе его характеристик — марка, описание товара, тип товара, серийный номер, цена); о поставщиках, у которых мы покупаем товар, (название организации, к кому обращаться, его должность в организации, адрес и телефон); о собственно закупках товара (описание заказа, обещанная дата доставки, фактическая дата доставки, цена за доставку товара); о сотрудниках; о доставке товара; о клиентах или покупателях.

Проектируя базу данных, мы столкнёмся с рядом задач, которые нам предстоит решить. Вот, например, можно видеть, что хранение всей этой информации вкупе приведет к тому, что данные хоть и будут структурированы и содержаться в одной таблице, но кто захочет постоянно вводить данные о товаре, если мы его продаем 10 раз за один день? А о поставщике, если он поставляет несколько видов товара?

Таким образом, можно создать таблицы — «Товары», «Поставщики», «Заказы», «Доставка», «Сотрудники», «Клиенты» каждая из которых будет хранить строго определенный набор данных.

Но в этом случае появляется другой вопрос: а как можно узнать, какой поставщик доставил тот или иной товар? Сколько товара осталось на складе? Сколько продано товара? И какой именно товар продается согласно записи таблицы «Заказы»? Для решения этой задачи нужно определить между таблицами отношения — то есть в таблицу «Заказы» необходимо ввести поле, содержащее наименование товара, и поле, содержащее сведения о проведённых сделках: заказано, получено и т.д.. Однако если мы будем нумеровать (или кодировать) записи, то проблема лишних обращений к клавиатуре будет еще быстрее решена, за счет ввода только номера товара (код товара), номера заказа (код заказа), номера операции (код операции) — при условии, что соответствующие поля будут введены в состав таблиц «Товары», «Поставщики», «Заказы», «Клиенты», «Сотрудники».

**1.2 Планирование процесса разработки**

Процесс разработки проекта состоит из четырёх стадий:

1. сбора и анализа требований к проектируемой базе данных;
2. анализа архитектуры;
3. планирования процесса разработки;
4. реальное построение СУБД.

*Стадия сбора и анализа требований* заключается в общении с пользователями данной БД и фиксации их требований: постоянный, быстрый, а главное удобный доступ к интересующей информации; возможность легко вносить, изменять и удалять различные данные и т. д.

*Стадия* *анализа архитектуры СУБД* состоит в выборе дизайна, технологий обработки информации и инструментов разработки БД и СУБД. В данном примере инструментом будет являться СУБД Microsoft Access.

*Стадия планирования* – создание планов по: закупке продукции, её реализации, стоимости, ресурсам, издержкам, прибыли и т. д.

*Стадия* *построение СУБД* заключается в проведении работ, контроля выполнения планов и оценки полученных результатов.

**1.3 Требования, предъявляемые к проектируемой СУБД**

1. **Функциональность** – описание необходимых функций СУБД, возможность и правила работы с ними.

*Пример:* возможность не хранить некоторые данные, а рассчитывать их при запросах по исходным данным.

1. **Производительность –** время исполнения запроса, ёмкость БД, количество обслуживаемых клиентов.

*Пример:* возможность поддерживания большого количества пользователей корпоративной сети при среднем времени выполнения запроса.

1. **Безопасность –** ограничиваются возможностями использования данных для обработки.

*Пример:* предусмотреть необходимую регистрацию пользователей и предоставления им права постоянного доступа.

1. **Масштабируемость –** возможность увеличения количества полей, записей полей, пользователей, периодичность передачи данных.
2. **Возможность изменения конфигураций –** показывает, насколько заказчик может изменять программное обеспечение с помощью разработчика.

*Пример:* администратор должен иметь возможность изменять и формировать новые запросы, формы и т. д.

1. **Совместимость –** возможность совместимости работы с разными СУБД.
2. **Доступность –** определяет количество часов обслуживания пользователей, длительность проведения работ.

*Пример:* администратор СУБД регистрирует время для архивации данных и проведении профилактических работ.

1. **Простота эксплуатации –** определяет условия, которые создают удобства использования СУБД.
2. **Простота освоения СУБД –** определяет допустимую длительность освоения приёмов управления СУБД.

Итак, набросав примерный проект базы данных, приступим к его реализации.

**2. Проектирование бд в среде Microsoft Access**

**2.1 Основные сведения**

Приложение Microsoft Access 97/2000 (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, а так же рядом других Web браузеров.

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.

**2.2 Системные требования**

1. Процессор 80486DX33 или мощнее

2. Windows 95/98/2000/XP или Windows NT (версия не ниже 3.51)

3. Не мене 12 Мб оперативной памяти (для совместной работы с другими приложениями не менее 24 Мб)

4. Около 100 Мб дискового пространства (только для Access и новых баз данных).

**2.3 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.

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

8. 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.

**2.4 Элементы базы данных**

1. Таблицы. В базе данных информация хранится в виде двумерных таблиц. Можно так же импортировать и связывать таблицы из других СУБД или систем управления электронными таблицами. Одновременно могут быть открыты 1024 таблицы.

2. Запросы. При помощи запросов можно произвести выборку данных по какому-нибудь критерию из разных таблиц. В запрос можно включать до 255 полей.

3. Формы. Формы позволяют отображать данные из таблиц и запросов в более удобном для восприятия виде. С помощью форм можно добавлять и изменять данные, содержащиеся в таблицах. В формы позволяют включать модули.

4. Отчёты. Отчёты предназначены для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде. Отчёты так же позволяют включать модули.

5. Макросы (см. выше)

6. Модули. Модули содержат VBA-код, используемый для написания процедур обработки событий таких, как, например, нажатия кнопки в форме или отчёте, для создания функций настройки, для автоматического выполнения операций над объектами базы данных и программного управления операциями, т.е. добавление VBA-кода позволяет создать полную базу данных с настраиваемыми меню, панелями инструментов и другими возможностями.

Модули снимают с пользователя приложения необходимость помнить последовательность выбора объектов базы данных для выполнения того или иного действия и повышают эффективность работы.

База данных может содержать до 32768 объектов.

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

**2.5 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.

**2.6 Панели и меню**

До сих пор приходилось иметь дело с отдельными объектами Access. И с каждым из них нужно работать в интерактивном режиме, открывая в окне базы данных вкладку требуемого объекта.

Но можно связать объекты в единое приложение. При этом вам не потребуется умение создавать сложные программы. Благодаря использованию кнопочных форм, диалоговых окон и меню работу с приложением можно автоматизировать и упростить, скрывая от пользователя ненужные в текущий момент объекты Access. Кнопочная форма представляет собой созданную пользователем форму с кнопками. С их помощью можно, например, запускать макросы открытия форм и печати отчетов. С помощью кнопочной формы выполнить несколько действий можно, щелкнув на одной кнопке. Используя кнопочную форму и другие объекты можно объединять созданные ранее объекты базы данных в одном приложении (при этом приложение может иметь вид, отличный от внешнего вида Access). В основном кнопочная форма используется в качестве главного меню приложения.

Поскольку кнопочная форма выполняет только роль меню приложения, в ней следует использовать минимальное число элементов управления. Поэтому обычно в кнопочную форму включают только кнопки, надписи, рамки объектов, линии и прямоугольники. Другие типы элементов управления, такие как текстовые поля (связанные с полями базы данных), списки, поля со списком, диаграммы и подчиненные формы, обычно в этой форме не используются. Создавая кнопочную форму, в ней следует разместить надписи, а также линии, прямоугольники или рисунки, чтобы сделать ее эстетически более привлекательной. Кнопка является простейшим элементом управления формы и используется для запуска макроса. В Access благодаря макросам, запускаемым с помощью кнопок, можно выполнять самые различные задачи.

Помимо кнопочной формы Access позволяет создать специальное меню, с помощью которого можно расширить возможности разрабатываемой системы. В это меню можно добавить часто используемые команды, а также команды, которые уже заданы для кнопок формы. Созданное меню заменит стандартное меню Access. Специальное меню связано с формой. Можно создать одно меню и использовать его в нескольких формах.

В Access можно создать три типа меню:

1. Menu Bar. Обычное меню, которое может располагаться вверху формы и иметь выпадающие подменю.

2. Tool Bars. Группы пиктограмм, обычно располагающиеся под меню.

3. Shortcut Bar. Меню, всплывающие после щелчка правой кнопки мыши.

**2.7 Visual Basic for Application**

В Access пользователь может управлять выполнением приложении, например, изменить информацию в таблице или щелкнуть на командной кнопке.

Используя в приложении макросы и процедуры, можно реализовать реакцию пользователя на эти действия. Для этого в Access предусмотрены характеристики событий для каждого элемента управления, который помещается в форму. Подключая макрос или процедуру к определенному событию элемента управления, можно не заботиться о последовательности действий, выполняемых пользователем в форме. Обработка событий представляет собой программный код, выполняемый при возникновении этого события. Код непосредственно присоединяется к форме или отчету, содержащему обрабатываемое событие.

В предыдущих версиях Access имела собственный язык Basic, называемый Access Basic. В последних версиях Access он заменен языком Visual Basic for Applications (VBA) компании Microsoft. Несмотря на некоторую схожесть, между этими языками есть существенные различия. VBA становится общим языком для всех приложений Microsoft Office. VBA является современным языком структурного программирования. Находясь в окне модулей, можно создавать и редактировать код VBA и процедуры.Visual Basic для приложений играет важную роль при разработке баз данных Access. С помощью VBA можно настроить формы и отчеты, запустить макросы, а также отобразить объект Access в других приложениях или извлечь данные. Используя Visual Basic для приложений, можно вывести формы и отчеты, выполнить методы объектов, а также создать и изменить элементы. Кроме того, имеется возможность работать с информацией непосредственно: можно создать наборы данных, задать их параметры и изменить информацию в них.

**3. ПРИМЕНЕНИЕ MS ACCESS В ООО ”ФОРУМ КОМПЬЮТЕРС”**

В компании ООО”Форум Компьютерс” для обеспечения целостности данных о поставщиках, клиентах, товарах и т.д. используется программная среда MS Access. Выбор именно этой среды обусловлен рядом преимуществ в соотношении с электронными таблицами:

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.

# **3.1 Сопровождение баз данных**

**3.1.1 Необходимость модернизации БД**

Объем информации и ее структура со временем эксплуатации СУБД растет и становится неуправляемой, если не принять мер по упорядочиванию БД. Как правило изменения касаются не только данных и их структуры, но и объектов БД. Пример: Сведения о заказчиках могут стать неполными, если не включить в их адреса E-mail и Web-узлы.

**3.1.2 Модификация таблиц**

Чтобы включить в БД новые сведения, в одних случаях достаточно ввести в поле сущ. Таблицы, а в других необходимо создать новую таблицу.

Таблицы БД описывают определенные объекты или темы. Если включаемые данные представляющие собой доп. характеристику существующего объекта. Если нужно включить в новый объект тему, нужно создать новую таблицу.

Прежде чем приступить к созданию таблицы, нужно создать список полей, а также определить их форматы и другие свойства.

**3.1.3 Создание таблицы с помощью мастера**

В окне БД нажать на кнопку Создать и выбрать Мастер таблиц.

Ключ таблицы – одно или несколько полей, однозначно идентифицирующих запись в таблицы. Ключевые записи служат нескольким целям: Если сортировка таблиц не задана, записи упорядочиваются по значению ключа. Также ключи используются для взаимосвязи таблиц, с помощью которых устанавливаются взаимоотношения записей. Счетчик – присваивает уникальный номер. Поля типа Счетчик – используют в качестве ключей.

**3.1.4 Использование конструктора для модернизации таблиц**

Конструктор может быть использован для создания новой таблицы.

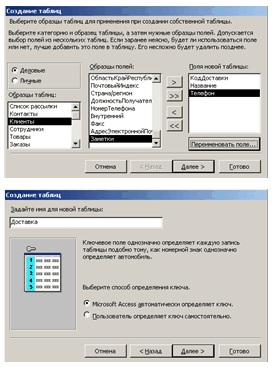
**3.1.5 Связывание таблиц с БД**

При дополнении новой таблицы, необходимо выяснить как она взаимодействует с другими таблицами и установить взаимосвязь между таблицами, записи которых логически связаны. Только в этом случай форму запроса показывают поля различных таблиц. Существует 2 способа. Можно установить 2 типа отношений: 1:1 и 1 ко многим – распространенный.

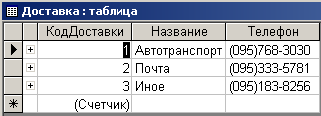
Пример: 1:1 – используется для обеспечения доступа ограниченным лицам.

Практическое применение:

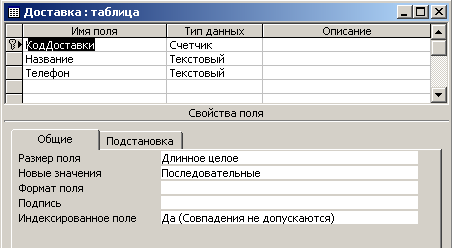
1) Создать с помощью мастера таблиц таблицу.



2) Заполняем таблицу в соответствии с требованиями предприятия. Получили:

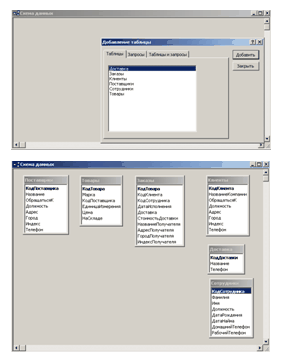


1. Откроем таблицу в конструкторе.

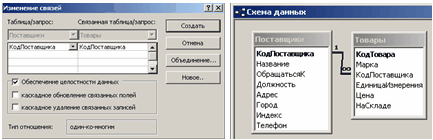


4) Повторяя шаги 1,2,3 создаем таблицы «Клиенты», «Поставщики», «Заказы», «Сотрудники» и «Товары».

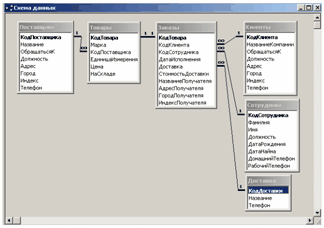
1. Созданные таблицы необходимо связать между собой для этого инициируем Сервис – Схема данных:



6) Удерживая левой кнопкой мыши перетаскиваем название поля «КодПоставщика» из таблицы «Поставщки» на одноименное поле в таблицу «Товары»:



1. Аналогичные операции произведем с остальными таблицами, получим:



**3.2 Построение запросов**

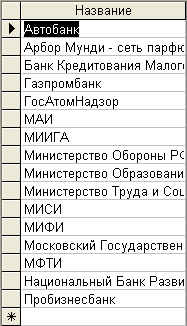
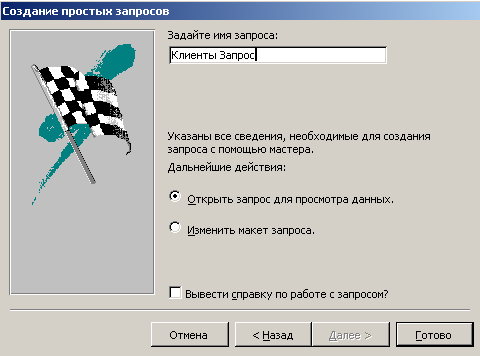
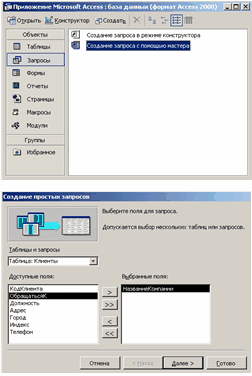
Microsoft Access предлагает ряд инструментов, которые позволяют получить ответы на поставленные перед базой данных вопросы. Access предлагает два способа задания запросов.

1. С помощью мастеров.
2. В режиме конструкторов.

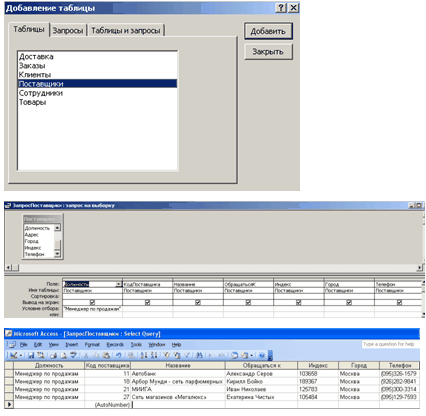
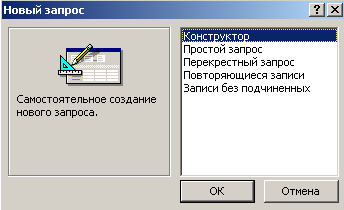
Мастер запросов выполняет пошаговую процедуру задания запроса. Режим конструктора позволяет не только задать запрос, но и модифицировать существующий. Запросы позволяют получить информацию, ограничивая диапазон просмотра данных, извлекая при этом поля из одной или нескольких таблиц. Запросы можно сохранять и использовать многократно.

Практическое применение:

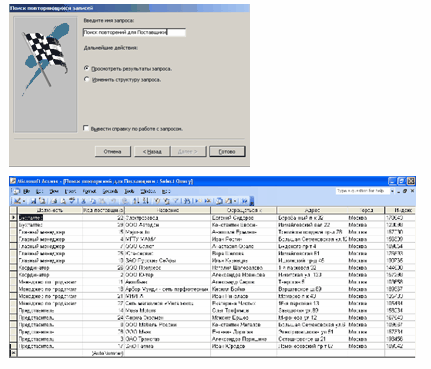
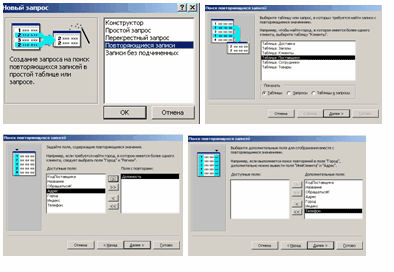
1. Открываем запросы.
2. Создаём новый запрос с помощью мастера.



4.Создаём новый запрос с помощью конструктора.



5. Создадим запрос на повторяющиеся записи.



## 3.3 Применение языка SQL в MS Access

Язык SQL – Structured Query Language используется в отличии от других языков программирования исключительно в задачах управления БД. С помощью SQL можно заставить программу выполнять на сетевом сервере.

**3.3.1 Использование выражения SELECT**

Команда SELECT состоит из служебного слова за которым следует список необходимых полей и выражение FROM, имя таблицы.

SELECT и FROM служебные слова SQL:

SELECT – название командные.

FROM – предложение, представляющее источник данных.

Список полей – параметры конструкций.

В качестве параметра списка полей допускает «\*».

Например, чтобы получить полное содержание таблицы :

SELECT \*FROM имя таблицы.

Порядок выполнение запроса на SQL:

1. Открыть БД.
2. В окне выбрать запросы.
3. Создать конструктор.
4. Добавить таблицу.
5. Вид/Режим SQL.

**3.3.2 Фильтрация данных**

Фильтрация данных с помощью предложения WHERE команды SELECT используется в тех случаях, когда необходимо ограничить множество записей возвращаемых запросом.

Конструкции SELECT предложение WHERE занимает место после FROM.

Синтаксис:

SELECT \_ список полей \_ FROM \_ список полей \_ WHERE \_ имя поля \_ оператор \_ значение (Or/And). За служебным словом WHERE следует набор предикатов в формате имя поля \_ оператор \_ значение. Количество предикатов неограниченно.

В качестве аргумента оператора используется любой из обычных операторов аргумента.

Значения – величина соответствующего типа (символьные литералы; слова, которые помечаются в кавычки).

Предикату может предшествовать служебное слово NOT, обозначающее унарный оператор отношения. Предикаты склеиваются с помощью операций конъюнкций (+), дизъюнкции Or.

Операторы применяющиеся в предложении WHERE:

1) Оператор BEETWEN сопровождается сл. сл. And использующий в контексте предложения WHERE для задания границ интервала, величины аргумента.

SELECT\*FROM \_ Music \_ WHERE \_ Artist BETWEEN ‘Elvis’ AND ‘Rolling Stones’.

В этом примере команда SELECT возвратит набор данных, содержащих сведения об альбомах исполнителей, имена которых содержат сведения об альбомах исполнителей и попадают в интервал от P до R.

2) Оператор IN. Он позволяет задавать искомые значения в полях таблиц. Синтаксис конструкций IN в контексте команды SELECT имеет вид: SELECT \_ список полей \_ FROM \_ имя таблицы. WHERE \_ имя поля \_ IN \_(значение 1, значение 2, … )

Например:

SELECT\*FROM \_ Music \_ WHERE \_ Publisher ‘Polygram’ ’Elektra’ ‘Capitol’

Указанная команда SELECT вернёт все строки данных, которые имеют отношения к звукозаписям, выпущенным компанией «Polygram», «Elektra», «Capitol».

Логические операторы конъюнкции (AND) и дизъюнкции (OR) требует наличие двух операторов. Оператор NOT – унарный оператор. Предложение WHERE может содержать любое число предикатов, соединяющих вместе логическими операторами сгруппированных с помощью круглых скобок.

SELECT\*FROM Music WHERE \_ Publisher =‘Capitol’ AND LastName = \_ ‘Cocker’ OR LastName = ‘Merchat’.

**3.3.3 Сортировка данных**

Сортировка данных конструкции выражения SELECT позволяет упорядочить возвращённые наборы данных по возрастанию или убыванию значений полей.

Этой цели служит приложение ORDER BY. После которого указывается одно или несколько выражений вида: имя поля \_ [Порядок сортировки, раздел. запятой]. Аргументы имя поля указывают на поля таблиц перечисленные в предложении FROM. В качестве необез. параметра – порядок – порядок сортировки допустимо использовать сл. сл. ASC и DESC возвращает по возрастанию и убыванию соответственно.

SELECT\*FROM \_ Music \_ ORDER BY Artist DESC , Title Asc

**3.3.4 Группировка столбцов**

Предложение GROUP BY применяется для группировки данных в столбцах. К нему необходимо обращаться при использовании так называемых агрегатирующих функций языка SQL. Например SUM. Группируя данные по определённым столбцам возвращающего набора следует включать в группу, либо все столбцы набора данных, либо те из них, которые не использованы в качестве аргументов агрегатирующих функций.

Предложение GROUP BY применяется в тех случаях, когда необходимо получить только одну строку из группы строк, в определённых структурах, которых характеризуется практичные значения.

**3.3.5 Использование предложения HAVING**

Предложение HAVING подобно WHERE помогает ограничить объем множества данных, получаемых в результате выполнения SELECT.

HAVING позволяет включать любое число предикатов, объединенных посредством булевых логических операторов.

Листинг 1:

Демонстрирует пример использования HAVING и применение вложенного запроса.

1: SELECT \* FROM Music WHERE ID =

2: SELECT Music\_ID FROM TRACKS

3: GROUP BY Music\_ID

4: Having CDATE (SUM (Track\_Length))>CDATE (“0:6:0”)

Строка 1 содержит заголовок внешнего запроса.

В строках 2,3,4 расположен текст подчиненного запроса. Подзапрос группирует записи таблицы TRACKS в соответствии с полями MUSIC\_ID.

Предложение HAVING осуществляет сравнение суммы, продолжительность звучания всех композиций данного альбома. С константой равной 6 мин.

В результате выполнения всего запроса будут возвращен набор записей таблицы. MUSIC для каждой из которых существует внешний ключ из таблицы Tracks и удовлетворяется условие подчиненного запроса.

В строке 4 демонстрируется пример употребления встроенной SQL функции CDATE, выполняющей преобразование числа значения DATE TIME.

В нашем случае с помощью CDATE осуществляется сопоставление функции внешних интервалов длительности звучания, выраженных в секундах.

**3.3.6 Объединение таблиц**

Реляционная БД предполагает наличие механизма объединения хранящихся в нескольких таблицах. Процесс и результат сбора данных об определенном объекте, которые хранятся в нескольких таблицах – объединение таблиц.

INNER JOIN – позволяет возвратить все записи для которых выполняется условие равенство, содержимого столбцов 2-х объединяемых таблиц.

Пример:

SELECT \*

FROM MUSIC INNER JOIN TRACKS ON\_MUSIC. ID, TRACKS. MUSIC\_ID

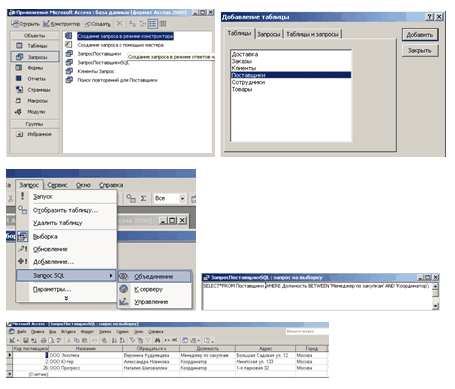
Эта конструкция возвратит все записи таблиц MUSIC и TRACKS для которых MUSIC и TRACKS равны.

LEFT JOIN – Применяется в случаях, когда следует вернуть все записи левой таблицы, и только те строки правой, значения полей которой соотв. данным левой таблицы.

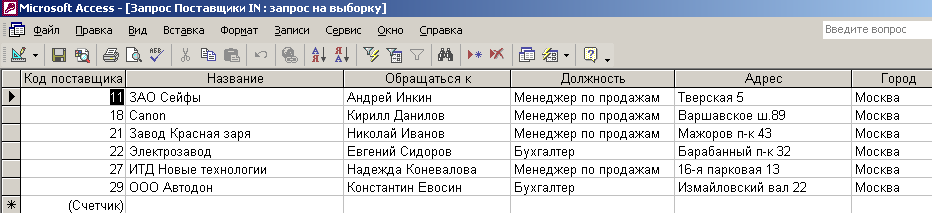
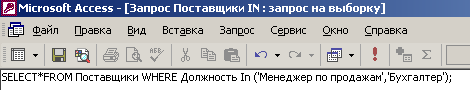
RIGHT JOIN – противоположно по значению оператора LEFT JOIN. При использовании этого оператора, возвращ. набор данных будет содержать

Практическое применение:

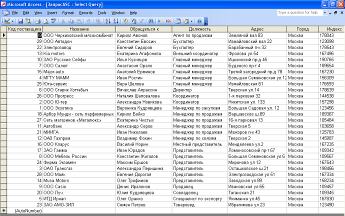
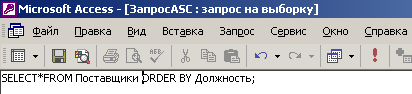
1. Оператор BEETWEN сопровождается сл. сл. And использующий в контексте предложения WHERE для задания границ интервала, величины аргумента.



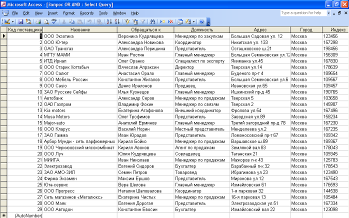
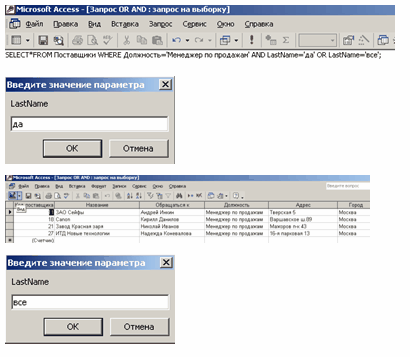
2. Оператор IN. Он позволяет задавать искомые значения в полях таблиц.



3. Сортировка данных конструкции выражения SELECT позволяет упорядочить возвращённые наборы данных по возрастанию или убыванию значений полей.



4. Логические операторы конъюнкции (AND) и дизъюнкции (OR).



**3.4 Анализ и оптимизация разработанной (модернизированной) БД и объектов управления данными**

Для быстродействия БД в среде Access можно пользоваться анализатором быстродействия. Анализатор быстродействия недоступен для вызова проектов MS Access. Проектом БД называется программное средство, обеспечивающее подключение к БД MS SQL Server, который используется для создания приложений в архитектуре «Клиент- Сервер». Файл проекта не содержит данных, как объектов так и таблиц.

Последовательность проведения работ по оптимизации БД.

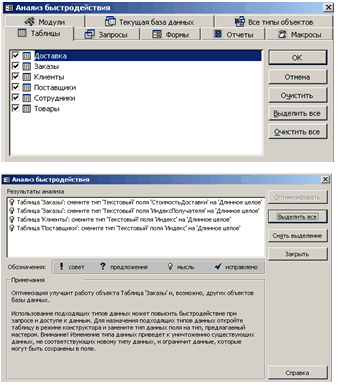
1. Открыть БД Access быстродействие которой необходимо повысить.
2. Выбрать Сервис – Анализ – Быстродействие.
3. Выбрать вкладку соответствующую типу объекта БД, который нужно оптимизировать, выбрать вкладку все типы объектов для просмотра всех объектов.
4. Выбрать имена оптимизированных объектов и нажать ОК.
5. Повторить пункты 3,4.

Далее БД предложит:

1. Совет
2. Предложение
3. Мысль

Предложение – подразумевает определенные компромиссы применяемые к оптимизации.

Для получения дополнительных сведений, целесообразно в мысли выбрать опцию таблицы и руководствоваться ею для совершенствования структуры БД.



**3.5 Программирование управление структур свойствами макросов**

**3.5.1 Макросы. Общие сведения**

Макрос представляет собой набор микрокоманд, которые создаются для автоматизации часто выполняемых задач. Группа макросов позволяет выполнять несколько задач последовательно. Основной компонент макроса – замкнутая инструкция, самостоятельная или в комбинации с другими макрокомандами, определяющая выполняемые в макросе действия. Если макросов много объединение родственных макросов в группы используются под одним общим именем.

Например: следующая группа макросов связанная с именем кнопки состоит из трёх родственных макросов: «Сотрудники», «Товары», «Отчёты». В каждом макросе выполняется макрокоманда «открыть форму».

В некоторых случаях требуется выполнять макрокоманду только при выполнении нескольких условий. Условие включения поля «Автор» и отношение «Равно» должно так же соединять и значение поле.

Если в макросе проверяется условие назначения, то для одних значений одно сообщение для других другое.

В подобных случаях условия позволяют определить порядок передачи управления между макросом в макросе. Условие задаётся любым логическим выражением, которое возвращает true или false. В зависимости от значений true или false управление передаётся разным макрокомандам.

После задания макрокоманд следует задать в нижней части макроса аргументы макрокоманды.

Основные правила задания аргументов макрокоманд.

1. В общем случае следует задавать аргументы макрокоманд в том порядке в котором они перечислены.
2. Если макрокоманда добавляется в макрос путём перетаскивания объекта БД из окна БД Access. Автоматически задаются подходящие аргументы для этой макрокоманды.
3. Если макрокоманда имеет аргумент, вызывающий имя объекта БД значение этого аргумента и соответствующий тип объекта можно задавать автоматически перетащив объект из окна БД в поле аргумента.
4. Для задания многих аргументов макрокоманд используется выражение перед которым используется «=».

**3.5.2 Создание макросов**

1) В окне БД выделить макросы

2) Создать

3) Добавить инструкции

* Нажать иконку ”раскрытие” и выбрать имя макрокоманды.
* В нижней части окна при необходимости указать аргументы макрокоманды, 2 значения которой являются объектами БД. Эти значения можно задавать путем перетаскивания.

**3.5.3 Создание групп макросов**

Чтобы объединить несколько макросов, необходимо создать группу макросов.

В условии макроса допускается использование любых выражений, которые возвращают значение true и false. Макрос будет выполняться, когда выражение true.

Таблица 1.

|  |  |
| --- | --- |
| Город = ”Москва” | Поле город из которого запускается макрос, содержит значения ”Москва” |
| Dcount (”[Код Заказы]”,”Заказы ”)>35 | Имеется более 35 значений |
| Dcount(”\*”,”Заказано”,[Код Заказа]=Forms! ”[Пример формы]! [Код заказа]”)>3 | Имеется больше 3 значений в заказано, у которых значение поля Код Заказа совпадает с Заказано. |
| Forms! [Товары]! [На складе]<5 | Значение поля На складе в форме товары <5 |
| Is Null ([Имя]) | Поле имя содержит пустое значение |
| [Страна]=”Литва” And Forms! [Сумма продаж]![Объем заказов]>100 | Поле Страна содержит значение Литва в котором Сумма продаж >100 |
| Msg Box (”Изменить?”)=1 | Нажать ОК. ОК – открывается ф-я Msg Box. |

**3.5.4 Запуск макроса**

Можно запускать непосредственно в группе макросов из другого макроса, или процедуры обработки события, а также в ответ на событие в форме, отчете, элементе управления.

Запуск из окна БД

Сервис – макрос – включить макрос процедуру. VBA Do cmd. Run Macro ”Выход”.

Access реагирует на событие различных типов, возникающих в формах, отчетах и элементах управления. В том числе на нажатие кнопок.

Порядок действия:

1. Открыть форму или отчет в конструкторе и вывести элементы управления.
2. Перев. события и свойства в ответ на который должна выполняться процедура.
3. Нажать кнопку Построить.

**ЗАКЛЮЧЕНИЕ**

Современную жизнь представить без современной техники просто невозможно. Ни одна фирма не обходится без помощи компьютеров, том числе и Форум Компьютерс.

В данной курсовой работе была разработана база данных для конкретного предприятия. Для этого использовалась система управления базами данных (СУБД) – Microsoft Access**.** Проведены исследования и разработка основные объекты СУБД Microsoft Access: таблицы, запросы, формы, отчёты, страницы, макросы и модули. При проектировании были выполнены цели, поставленные во время планирования БД и СУБД. Разработанная база данных полностью отвечает предъявленным к ней требованиям. В работе применялись знания, полученные за время изучения курса "Информационное обеспечение систем управления"и из дополнительных источников.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Г.М. Холодов. Лекции по предмету «Информационное обеспечение».

2. Г.М. Холодов, Т.И. Крыжановская. Лабораторный практикум по предмету «Информационное обеспечение».

3. Access – для самостоятельного освоения. А.И.Бородина, Л.И.Крошинская, Е.Н.Лядинская. Мн.:НО ООО «БИП-С»,2002. с.136.

4.Создание баз данных в СУБД Access. Анисимова Ж.Н. Мн.:БГУ,1998.

5.Access в примерах. Гончаров А. СПБ.:Питер,2000.

6.Практическое пособие по информатике. Microsoft Access. Шилина А.Л. Минск 2005.

7. www.forum3.ru