СОДЕРЖАНИЕ

Введение ……..4
1 Основная часть ……6
1.1 Основные понятия баз данных ……6
1.1.1 Базы данных и системы управления базами данных ……6
1.1.2 Состав СУБД и свойства полей ……..7
1.1.3 Типы и безопасность данных ……9
1.2 Проектирование баз данных ……..12
1.2.1 Режимы работы с базами данных ……12
1.2.2 Объекты баз данных ……..12
1.2.3 Реляционная модель базы данных ……14
2 Практическая часть ……..16
2.1 Построение базы данных автосалона ……16
Заключение ……..28
Список использованной литературы ……29

ВВЕДЕНИЕ

Системы управления базами данных играют исключительную роль в организации современных промышленных, инструментальных и исследовательских информационных систем. В связи, с чем актуализируется проблема анализа особенностей использования систем управления базами данных и выбора конкретных инструментов определенной СУБД. Среди наиболее ярких представителей систем управления базами данных можно отметить: MS Access и MySQL.

Фактически каждому специалисту, не смотря на сферу его деятельности, приходится заниматься сбором, накоплением и обработкой данных. Microsoft Access является известной системой управления базами данных (СУБД). Как и остальные продукты данной категории, специализирована для хранения и поиска данных, представления информации в удобном виде и автоматизации нередко встречающихся операций (например, ведение счетов, учет, планирование и т.д.). С поддержкой MS Access возможно разрабатывать простые и удобные формы ввода данных, а еще делать обработку данных и выдачу сложных отчетов.

В последние годы создаются программы, способные обрабатывать большие потоки информации. С помощью таких программ и разрабатываются информационные системы. Целью любой информационной системы является обработка данных об объектах и явлениях реального мира и предоставление человеку нужной информации о них.

Актуальность исследования

В настоящее время базы данных повсеместно используются в современном обществе во всех отраслях производства.

Посредствам базы данных удобно составлять отчеты, находить быстро необходимые данные. Электронные адреса – это тоже своего рода база данных.

Большое количество информации требует создание новых

современных баз данных для дальнейшей работы с ней.

Цель выпускной квалификационной работы: ознакомление с основными средствами Microsoft Access для создания баз данных; создать собственную базу данных.

Задачи исследования:

- Изучить литературу по теме баз данных.
- Изучить и описать особенности работы программы Microsoft Access.
- Разработать собственную базу данных.
- Оптимизация автосалона с помощью внедрения базы данных

Объектом исследования являются современные базы данных.

Предметом исследования является создание базы данных с помощью Microsoft Access для обработки данных автосалона.

1 ОСНОВНАЯ ЧАСТЬ

1.1 ОСНОВНЫЕ ПОНЯТИЯ БАЗ ДАННЫХ

1.1.1 Базы данных и системы управления базами данных

База данных – это организованная структура, предназначенная для хранения информации. В современных базах данных хранятся не только данные, но и информация.

Это утверждение легко пояснить, если, например, рассмотреть базу данных крупного банка. В ней есть все необходимые сведения о клиентах, об их адресах, кредитной истории, состояние расчетных счетов, финансовых операциях и т.д. Доступ к этой базе данных имеется у достаточно большого количества сотрудников банка, но среди них вряд ли найдется такое лицо, которое имеет доступ ко всей базе полностью и при этом способно единолично вносить в нее произвольные изменения. Кроме данных, база содержит методы и средства, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

С понятием базы данных тесно связано понятие системы управления базой данных. Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнение ее содержимым, редактирование содержимого и визуализации информации. Под визуализацией информации базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройства вывода или передачи по каналам связи.

В мире существует множество систем управления базами данных. Несмотря на то, что они могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных понятий. Это дает возможность рассмотреть одну систему и обобщить ее понятия, приемы

и методы на весь класс СУБД. В качестве такого учебного объекта выберем СУБД Microsoft Access, входящую в пакет Microsoft Office [2,5].

1.1.2 Состав СУБД и свойства полей

Язык описания данных (ЯОД) – Средства описания данных в БД и связей между ними. Средствами этого языка описывается структура БД, форматы записей, пароли, защищающие данные.

Язык манипулирования данными (ЯМД) – язык для выполнения операций над данными, позволяющий менять их строение.

Для различных СУБД реализация этих уровней языков может быть различной. В одних случаях ЯОД и ЯМД требует составления пользователем программы полностью “вручную”, в других (что отражает современную тенденцию) в СУБД присутствует средства визуальной (зримой, наглядной) разработки программ. Для этого в современных СУБД имеются редакторы экранных форм, отчетов. “Кирпичиками” (инструментами) таких редакторов являются поля различных видов (поля ввода, поля вывода, вычисляемые поля), процедуры обработки различных типов (формы ввода, таблицы, отчеты, запросы). На основании созданных пользователем объектов программы – генераторы формируют программный код на языке конкретной машины или на промежуточном языке.

Поля базы данных не просто определяют структуру базы – они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей. Ниже перечислены основные свойства полей таблиц баз данных на примере СУБД Microsoft Access [6].

Имя поля – определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц).

Тип поля – определяет тип данных, которые могут содержаться в данном поле.

Размер поля – определяет предельную длину (в символах) данных, которые могут размещаться в данном поле.

Формат поля – определяет способ форматирования данных в ячейках, принадлежащих полю.

Маска ввода – определяет форму, в которой вводятся данные, а поле (средство автоматизации ввода данных).

Подпись – определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство Имя поля).

Значение по умолчанию – то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).

Условие на значение – ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).

Сообщение об ошибке – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных.

Обязательное поле – свойство, определяющее обязательность заполнения данного поля при наполнении базы.

Пустые строки – свойство, разрешающее ввод пустых строковых данных (от свойства Обязательное поле отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например текстовым).

Индексированное поле – если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значение в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Поскольку в разных полях могут содержаться данные разного типа, то и свойства у полей могут различаться в зависимости от типа данных. Так,

например, список вышеуказанных свойств полей относится в основном к полям текстового типа. Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним и свои. Например, для данных, представляющих действительные числа, важным свойством является количество знаков после десятичной запятой. С другой стороны, для полей, используемых для хранения рисунков, звукозаписей, видео клипов и других объектов OLE, большинство вышеуказанных свойств не имеют смысла [1].

1.1.3 Типы и безопасность данных

Таблицы баз данных, как правило, допускают работу с большим количеством разных типов данных. Так, например, базы данных Microsoft Access работают со следующими типами данных.

Текстовый – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).

Числовой – тип данных для хранения действительных чисел.

Поле Мемо – специальный тип данных для хранения больших объемов текста (до 65 535 символов). Физически текст не хранится в поле. Он храниться в другом месте базы данных, а в поле храниться указатель на него, но для пользователя такое разделение заметно не всегда.

Дата/время – тип данных для хранения календарных дат и текущего времени.

Денежный - тип данных для хранения денежных сумм. Теоретически, для их записи можно было бы пользоваться и полями числового типа, но для денежных сумм есть некоторые особенности (например, связанные с правилами округления), которые делают более удобным использование специального типа данных, а не настройку числового типа.

Счетчик – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование – для порядковой нумерации записей.

Логический - тип для хранения логических данных (могут принимать только два значения, например Да или Нет).

Гиперссылка – специальное поле для хранения адресов URL Web-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.

Мастер подстановок – это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать их из раскрывающегося списка [4,7].

Базы данных – это тоже файлы, но работа с ними отличается от работы с файлами других типов, создаваемых прочими приложениями. Выше увидели, что всю работу по обслуживанию файловой структуры берет на себя операционная система. Для базы данных предъявляются особые требования с точки зрения безопасности, поэтому в них реализован другой подход к сохранению данных.

Базы данных – это особые структуры. Информация, которая в них содержится, очень часто имеет общественную ценность. Нередко с одной и той же базой работают тысячи людей по всей стране. От информации, которая содержится в некоторых базах, может зависеть благополучие множества людей. Поэтому целостность содержимого базы не может и не должна зависеть ни от конкретных действий некоего пользователя, забывшего сохранить файлы перед выключением компьютера, ни от перебоев в электросети.

Проблема безопасности баз данных решается тем, что в СУБД для сохранения информации используется двойной подход. В части операций, как обычно, участвует операционная система компьютера, но некоторые операции сохранения происходят в обход операционной системы [3].

1.2 ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

1.2.1 Режимы работы с базами данных

Обычно с базами данных работают две категории пользователей.

Первая категория – проектировщики. Их задача состоит в разработке структуры таблиц базы данных и согласование ее с заказчиком.

Кроме таблиц проектировщики разрабатывают и другие объекты базы данных, предназначенные, с одной стороны, для автоматизации работы с базой, а с другой стороны – для ограничения функциональных возможностей работы с базой (если это необходимо из соображений безопасности).

Проектировщики не наполняют базу конкретными данными, (заказчик может считать их конфиденциальными и не предоставлять посторонним лицам). Исключение составляет экспериментальное наполнение модельными данными на этапе отладки объектов базы.

Вторая категория исполнителей, работающих с базами данных -пользователи. Они получают исходную базу данных от проектировщиков и занимаются ее наполнением и обслуживанием. В общем случае пользователи не имеют средств доступа к управлению структурой базы – только к данным, да и то не ко всем, а к тем, работа с которыми предусмотрена на конкретном рабочем месте.

Соответственно СУБД имеет два режима работы: проектировочный и пользовательский. Первый режим предназначен для создания или изменения структуры базы, а так же создание ее объектов. Во втором режиме происходит использование ранее подготовленных объектов для наполнения базы или получения данных из нее.

1.2.2 Объекты баз данных

Таблицы – это основные объекты любой базы данных. Во-первых, в таблицах хранятся все данные, имеющиеся в базе, а во-вторых, таблицы

хранят и структуру базы (поля, их типы и свойства).

Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразования данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнения таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое.

Если **запросы** – это специальные средства для отбора и анализа данных, то **формы** – это средства для ввода данных. Смысл их тот же – предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно с этим в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и прочее) для автоматизации ввода. Преимущества форм раскрываются особенно наглядно, когда происходит ввод данных с заполненных бланков. В этом случае форму делают графическими средствами так, чтобы она повторяла оформление бланка – это заметно упрощает работу наборщика, снижает его утомление и предотвращает появление печатных ошибок [10].

Отчеты. По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на принтер. В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов.

Страницы. Это специальные объекты баз данных, реализованных в последних версиях СУБД Microsoft Access (начиная с Access 2000). Правда, более корректно их называть страницами доступа к данным. Физически это особый объект, выполненный в коде HTML, размещаемый на Web-странице и передаваемый клиенту вместе с ней. Сам по себе этот объект не является

базой данной, но содержит компоненты, через которые осуществляется связь переданной Web-страницы с базой данных, остающейся на сервере. Пользуясь этими компонентами, посетитель Web-узла может просматривать записи базы в полях страницы доступа. Таким образом, страницы доступа к данным осуществляют интерфейс между клиентом, сервером и базой данных, размещенной на сервере. Эта база данных не обязательно должна быть базой данных Microsoft Access. Страницы доступа, созданные средствами Microsoft Access, позволяют работать также с базами данных Microsoft SQL Server.

Макросы и модули. Эти категории объектов предназначены как для автоматизации повторяющихся операций при работе с СУБД, так и для создания новых функций путем программирования. В СУБД Microsoft Access макросы состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с базой. Модули создаются средствами внешнего языка программирования, в данном случае языка Visual Basic for Applications. Это одно из средств, с помощью которых разработчик базы может заложить в нее нестандартные функциональные возможности, удовлетворить специфическое требование заказчика, повысить быстродействие системы управления, а также уровень ее защищенности [9].

1.2.3 Реляционная модель базы данных

Реляционная модель базы данных представляет собой множество взаимосвязанных двумерных таблиц – реляционных таблиц, в каждой из которых содержатся сведения об одной сущности автоматизированной предметной области.

Каждая строка в таблице содержит данные, относящиеся к некоторой вещи или какой-то её части и называется записью. Каждый столбец таблицы описывает какой-либо атрибут этой вещи. Например, атрибут сущности – город, экземпляр сущности – Ташкент.

Нормализация – это удаление из таблиц повторяющихся данных

путем их переноса в новые таблицы, экземпляры сущности которой не должны содержать повторяющихся значений. Минимальное дублирование данных в реляционной БД обеспечивают высокую эффективность поддержания БД в актуальном и непротиворечивом состоянии, однократный ввод и корректировку данных.

Ключ – это группа из одного или более атрибутов, которая уникальным образом идентифицирует строку. По значению ключа таблицы отыскивается единственная запись в таблице. Ключ может состоять из одного или нескольких атрибутов. Значение уникального ключа не может повторяться в нескольких записях.

Между полями, а так же между записями могут быть установлены различные логические связи. Логические связи дают возможность объединять данные из различных таблиц. Связь каждой пары таблиц обеспечивается одинаковыми полями в них – ключом связи.

Типы реляционных связей:

один к одному – это таблицы, содержащие какие-либо дополнительные сведения об отчётах первой таблицы;

один ко многим – каждая запись главной таблицы может быть поставлена в соответствии с одной или несколькими записей подчинённой таблицы [10].

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 ПОСТРОЕНИЕ БАЗЫ ДАННЫХ АВТОСАЛОНА

Для того чтобы лучше понять Access построим свою Базу данных. Сначала определимся, какую информацию будет содержать наша База данных. База данных может содержать информацию о чем угодно: о зарплате, о торговых поставках, о больных в поликлинике, о студентах ВУЗа, о друзьях, о книгах в вашей библиотеке. Построим простую - о продаже автомобилей в автосалоне. Эта База данных будет состоять из трёх взаимосвязанных таблиц:

1. Список автомобилей, который будет содержать такую информацию:

- Код машины.
- Марка авто.
- Цвет.
- Цена.

2. Список клиентов, который будет содержать такую информацию:

- Код клиента.
- Фамилия.
- Имя.
- Отчество.
- Адрес.
-Телефон.

3. Список продаж, который будет содержать такую информацию:

- Код продажи.
- Марка машины.
- Клиент.
- Дата продажи.

Графически это может быть представлено в виде трех таблиц:

Таблица 1. Автомобили

Код машины	Марка авто	Цвет	Цена

Таблица 2. Клиенты

Код клиента	Фамилия	Имя	Отчество	Адрес	Телефон

Таблица 3. Продажи

Код продаж	Марка машины	Клиент	Дата продажи

Запускаем Access и создаем новую базу данных: **Создать** – Новая база данных – вводим название новой базы «Автосалон.accdb» - и жмем кнопку **Создать** (Смотрим *Рис.1*). Сначала построим три выше описанных таблицы. Для этого перейдем в режим конструктор: это можно сделать нажав правую кнопку мыши по вкладке таблицы и в открывшемся списке выбрать **Конструктор** (*Рис.2*). Либо на верхней панели инструментов выбрать: вкладку Поля – Режимы – Конструктор (*Рис.3*). Появится окошко с именем таблицы, называем «Автомобили».

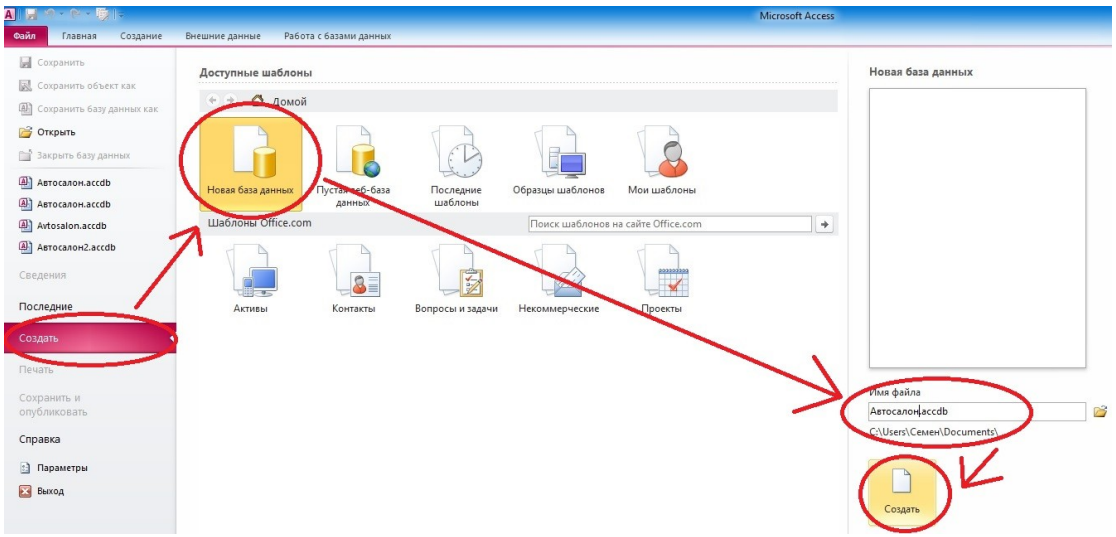

Рис.1 Создание новой базы данных

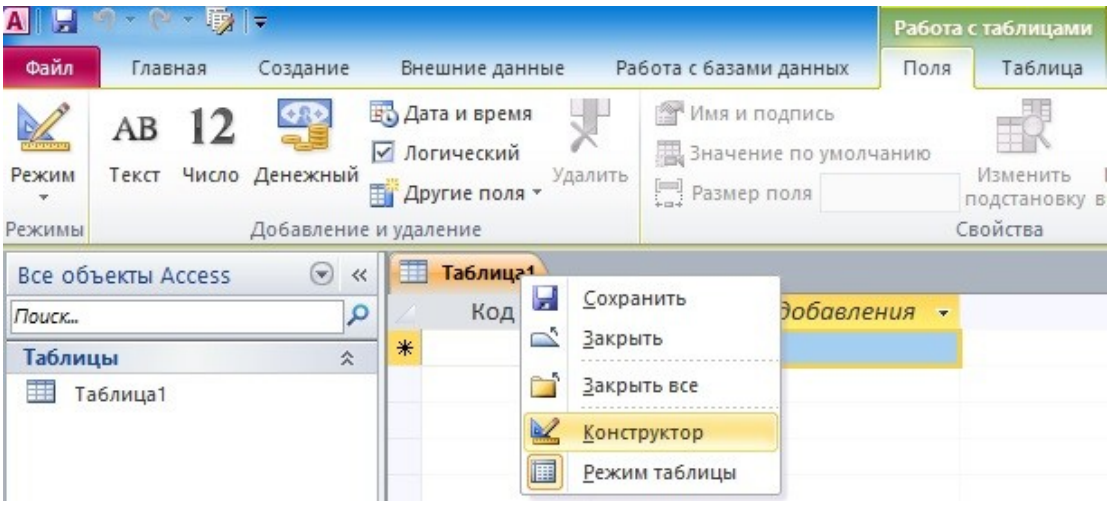

Рис.2 Переход в режим конструктора

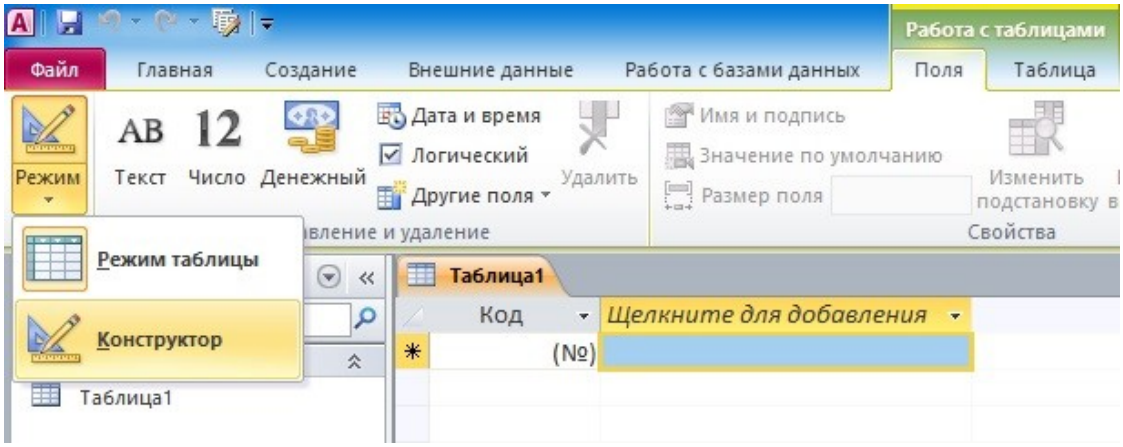

Рис.3 Другой вариант включения режима конструктора

Вводим имена полей и их типы:

Для таблицы «Автомобили»: 1 Имя поля «Код машины» – тип данных «Счетчик». 2 Имя поля «Марка авто» - тип данных «Текстовый» и размер поля «30». 3 Имя поля «Цвет» - тип данных «Текстовый», размер поля «30». 4 Имя поля «Цена» - тип данных «Денежный». Сохраняем (ctrl+S).

Для таблицы «Клиенты»:

Имя поля «Код клиента» – тип данных «Счетчик». И делаем его ключевым.

Имя поля «Фамилия» - тип данных «Текстовый» и размер поля «30».

Имя поля «Имя» - тип данных «Текстовый», размер поля «30».

Имя поля «Отчество» - тип данных «Текстовый», размер поля «30».

Имя поля «Адрес» - тип данных «Текстовый», размер поля «30».

Имя поля «Телефон» - тип данных «Текстовый», размер поля «11»,

Маска ввода «"+7-"999\-999\-99\-99». Сохраняем (ctrl+S).

Для таблицы «Продажи»:

Имя поля «Код продажи» – тип данных «Счетчик». Ключевое поле.

Имя поля «Марка машины» - тип данных «Мастер подстановок», выберем «**Объект после подстановки получит значения из другой таблицы**» жмем **Далее**, выбираем таблицу «Автомобили» жмем **Далее**, Выбираем «Марка авто» **Далее, Далее и Готово.**

Имя поля «Клиент» - тип данных «Мастер подстановок», выберем «**Объект поле подстановки получит значения из другой таблицы**» жмем **Далее**, выбираем таблицу «Клиенты» жмем **Далее**, Выбираем «Фамилия» **Далее, Далее и Готово.** 4 Имя поля «Дата продажи» - тип данных «Дата/время». Формат поля «**Краткий формат даты**». Сохраняем (ctrl+S). Закрываем все таблицы.

Теперь для ввода данных создадим формы. На панели инструментов, на вкладке **Создание** выбираем **Мастер форм**. В появившемся окне выбираем таблицу: Автомобили и выбираем все поля (*Рис.4*). **Далее.** Выбираем вид нашей формы «**в один столбец**». Имя пусть будет «Автомобили» и нажимаем **Готово.**

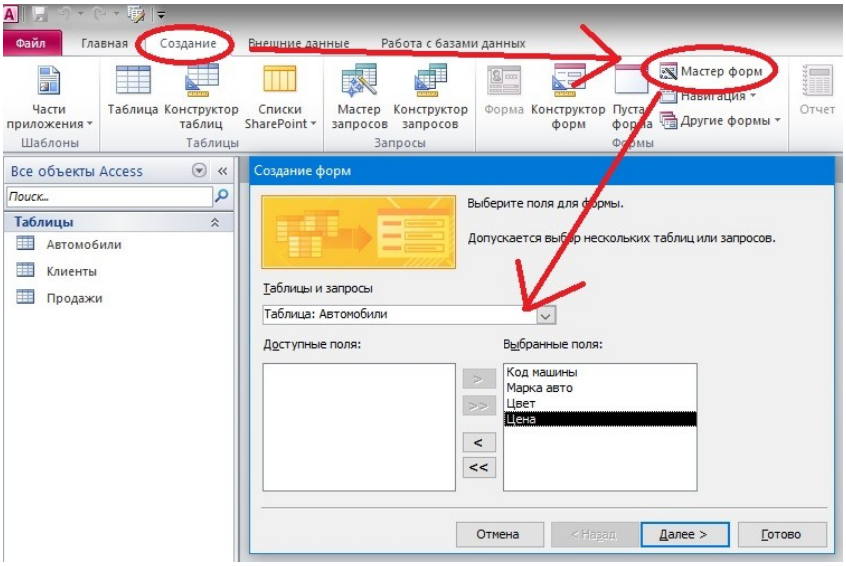

Рис.4 Создание форм

Создадим кнопки в этой форме: для этого переходим в режим **Конструктора.** На панели инструментов вкладка **Конструктор** - иконка

создания кнопки – и указываем куда установить кнопку – откроется новое окно в ней выбираем: **Переходы по записям** и **Предыдущая запись** (*Рис.5*), и жмем **Готово.** Создаем следующую кнопку: **Переходы по записям** и **Следующая запись**. Еще кнопка: **Обработка записей** и **Добавить запись**, **Далее** и выберем **Текст**, жмем **Готово**. И последнюю кнопку для закрытия формы: **Работа с формой** и **Закрыть форму**, **Готово**.

Переходим в режим формы и начинаем заполнять. После заполнения можно открыть таблицу «Автомобили» и проверить введенные данные.

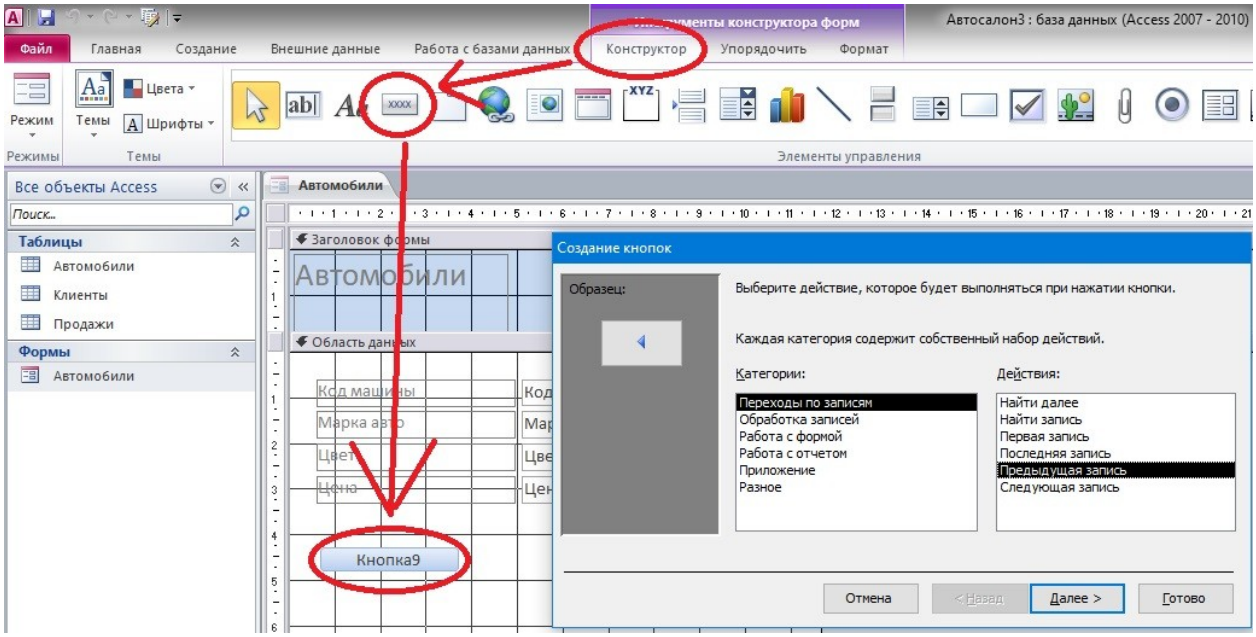

Рис.5 Создание кнопки в форме

Далее создадим вторую форму, но уже другим способом: выделяем таблицу «Клиенты» - и на панели инструментов заходим во вкладку **Создание - Другие формы - Несколько элементов** (*Рис.6*). В режиме конструктора добавим в эту форму только кнопку **Закрыть форму**. Переходим в режим формы и заполняем данными.

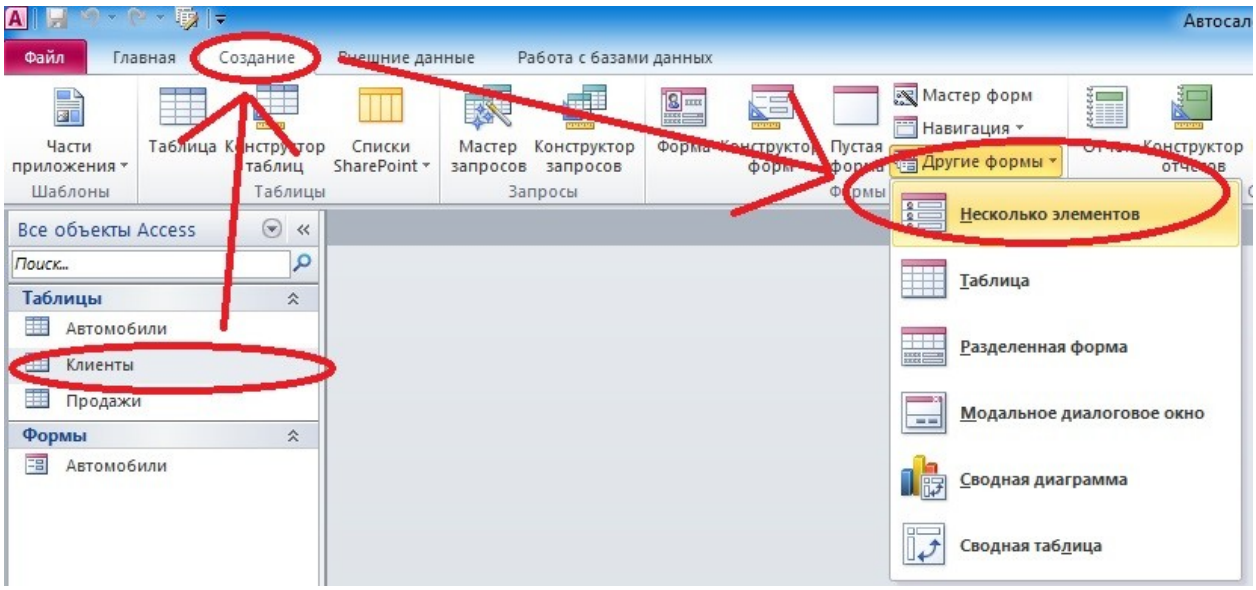

Рис.6 Другой способ создания форм

Идем дальше, создадим следующую форму третьим способом. Выделяем таблицу «Продажи» - на панели инструментов **Создание - Форма.** (*Рис.7*) Переходим в режим конструктора и создаем такие же кнопки как в форме «Автомобили» (можно их скопировать оттуда). Сохраняем с названием «Продажи» и закроем.

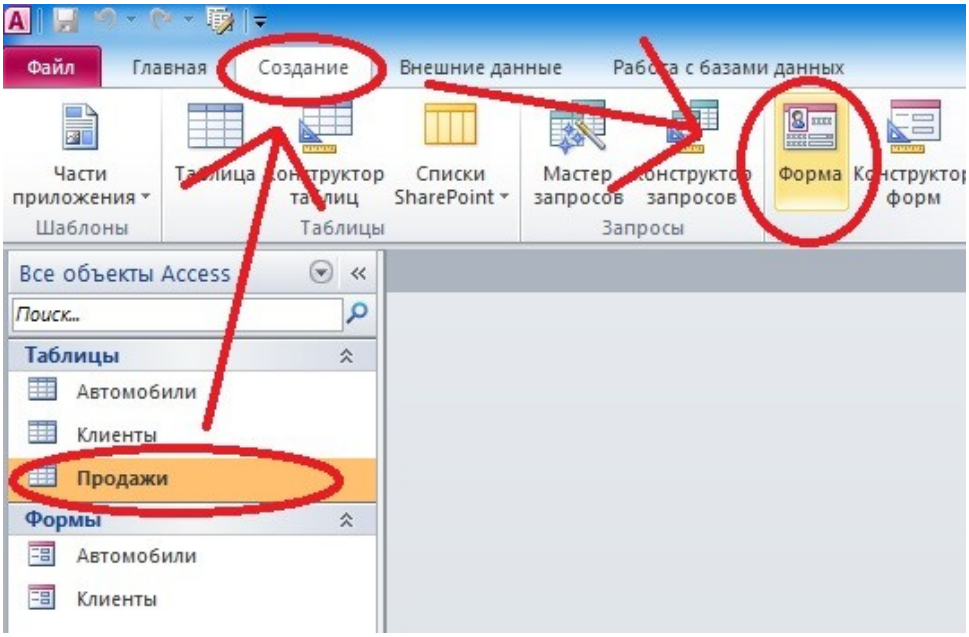

Рис.7 Третий способ создания форм

Перед тем как будем заполнять данные, создадим схемы данных, для корректного ввода данных в таблицу «Продажи»: на панели инструментов заходим в **Работа с базами данных** - и нажимаем **Схема данных** (*Рис.8*).

Дважды кликаем по связи между «Автомобили» и «Продажи» и в открывшемся окне ставим галочки: **Обеспечение целостности данных** и **каскадное обновление связанных полей** (*Рис.9*). Так же делаем со связью между «Продажи» и «Клиенты». Закрываем «Схему данных», открываем форму «Продажи» и заполняем.

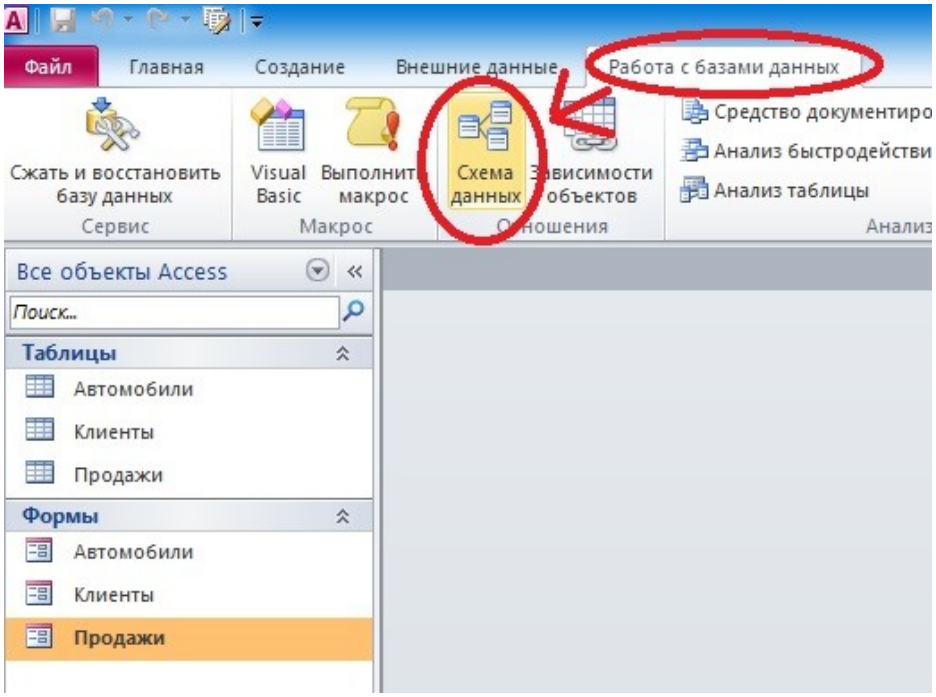

Рис.8 Создание схемы данных

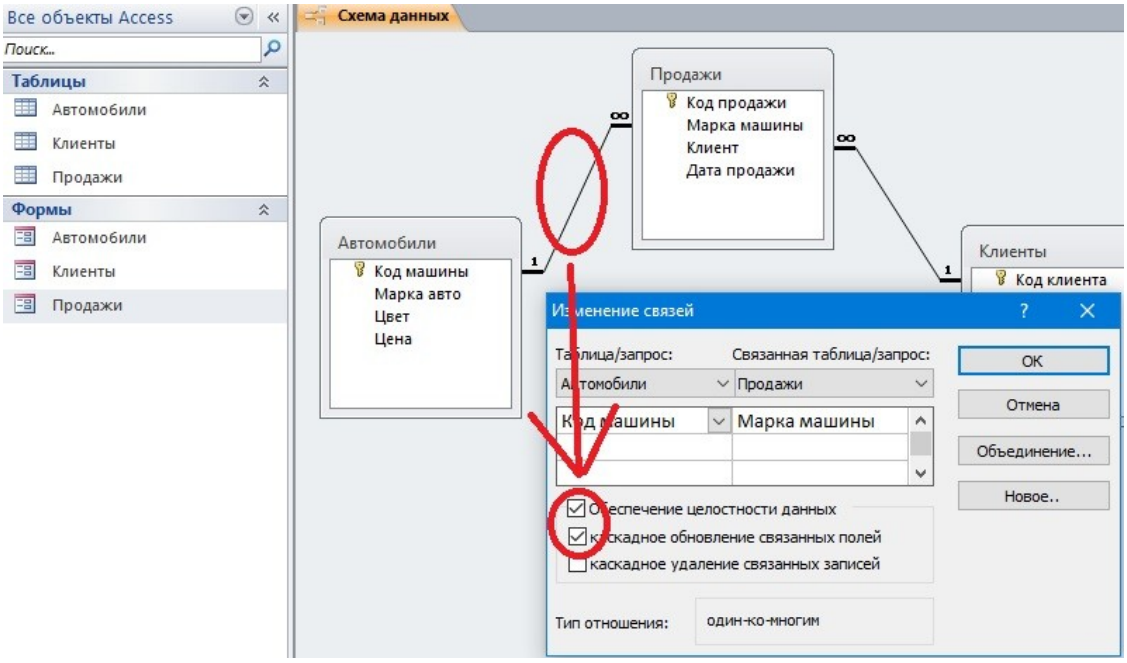

Рис.9 Настройка связей в схеме данных

Теперь создадим 6 запросов:

В первом запросе выведем на экран все данные об автомобиле определенного цвета. На панели инструментов вкладка **Создание** - **Конструктор запросов**. В открывшемся окне выберем таблицу «Автомобили», нажимаем кнопку **Добавить**. Добавляем все поля, кликая дважды на каждое поле (*Рис.10*). Внизу, в столбце «Цвет» - «Условие отбора» пишем: Белый. Нажав на кнопку **Выполнить** в панели инструментов, выведутся авто белого цвета. Сохраним данный запрос как «Белый авто».

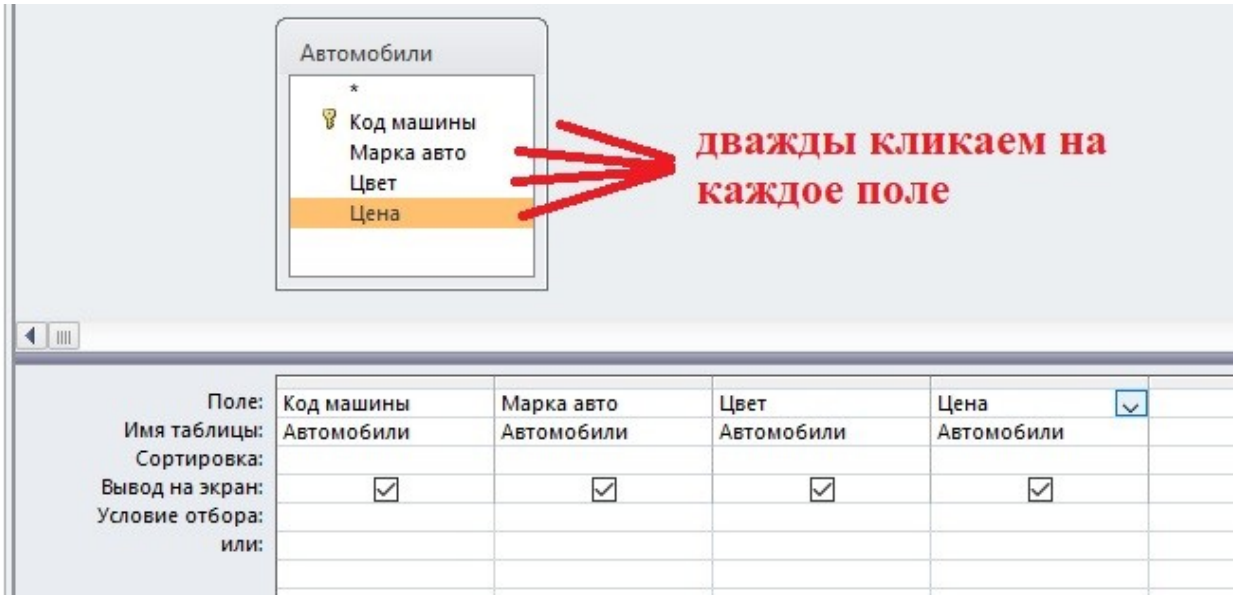

Рис.10 Добавление нужных полей в запрос

Во втором запросе выведем данные об авто цена которых превысил определенное число. Проделываем все тоже самое, что и в первом запросе, но «Условие отбора» пишем в «Цена»: >2600000.

В третьем запросе выведем на экран цвет и цену на авто, марку которого пользователь вводит с клавиатуры. Данный вид запроса называется параметрический. В «Конструкторе запросов» добавляем таблицу «автомобили». Выбираем поля «цвет», «розничная цена» и «марка авто». В «Условие отбора» под «марка авто» пишем: [Введите марку авто] (обезательно в квадратных скобках). Сохраняем данный запрос как: «Поиск по марке».

Четвертый запрос будет перекрестный. Выведем на экран доход от продажи каждой марки автомобиля. В «конструкторе запросов» выбираем

все три таблицы. В тело запроса добавляем: «Марка авто», «фамилия», «цена». Чтоб переключить вид запроса на перекрестный, нада на панели инструментов нажать кнопку «Перекрестный». Для перекрестной таблицы нужно указать: заголовки столбцов, заголовки строк и значение. А в групповой операции под Ценой, выбираем функцию Sum (*Рис.11*). Сохраним данный запрос как «перекрестный».

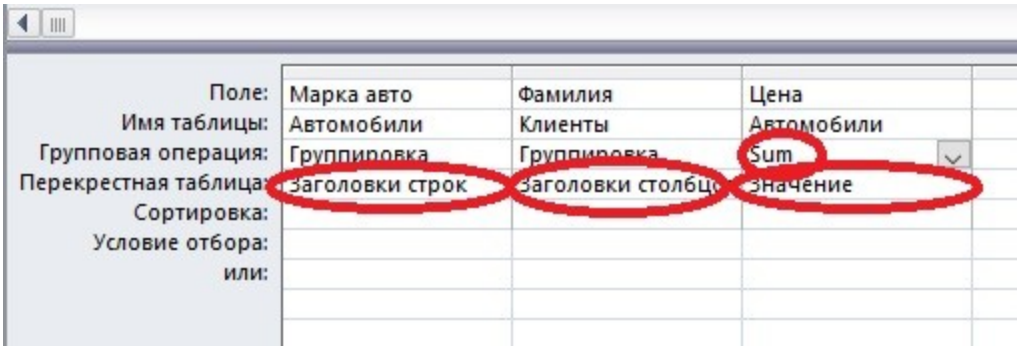

Рис.11 Настройка запроса «перекрестный»

Пятый запрос будет итоговый, в котором посчитаем стоимость проданных машин на каждую дату. В «конструкторе запросов» добавляем таблицы: «автомобили» и «продажи». Выбираем поля: «дата продажи» и «цена». На панели инструментов нажимаем кнопку **Итоги**. В групповой операции под Ценой, выбираем функцию **Sum**. Сохраняем данный запрос как «сумма продаж по дням».

В последнем запросе будем выводить на экран фамилии клиентов, которые начинаются на определённую букву. В «конструкторе запросов» добавляем таблицы: «клиенты». Выбираем поля: фамилия, имя, отчество и телефон. В условие отбора под фамилией пишем: **like [Введите 1 букву фамилии] & ***. Сохраняем как «Поиск по фамилии».

Далее переходим к отчетам. Выделяем таблицу «Автомобили», переходим по вкладку **Создание** и нажимаем кнопку **Отчет**. Сохраним данный отчет как «Автомобили».

Теперь создадим более сложный отчет, в котором будет применяться группировка и сортировка. Переходим на вкладку **Создание**, выберем **Мастер отчетов**. Из таблицы «автомобили» выбираем поля: «марка авто» и

«цена». Из таблицы «клиенты» выбираем поля: «фамилия» и «имя». Из таблицы «продажи» выбираем поля: «код продажи» и «дата продажи». Нажимаем «далее». В выборе вида представления данных выбираем «клиенты». Нажимаем **далее**. Потом опять **далее**. В окне выбора порядка сортировки и вычисления выбираем по дате продажи и чтобы была по возрастанию. Нажимаем **далее**. Макет – ступенчатый, ориентация – книжная. **Далее**. Назовем: «Отчет по продажам». **Готово**.

Перед созданием главной кнопочной формы, понадобится создать макросы для открытия таблиц и запросов. Заходим во вкладку **Создание** и кликаем по кнопке **Макрос**. Появляется выпадающий список. В нем выбираем пункт: «Открыть таблицу». В имени таблицы выбираем таблицу «автомобили». Сохраняем данный макрос. Создаем таким способом остальные макросы на открытие таблиц и запросов.

Для создания главной кнопочной формы, сперва нарисуем схему кнопок (*Рис.12*).

Начиная с Microsoft Access 2010 из меню инструментов разработчики убрали кнопку «диспетчер кнопочных форм». Чтоб добавить эту кнопку, переходим в **файл** - **параметры** - **панель быстрого доступа**. В списке выбираем все команды и ищем «диспетчер кнопочной формы». Выделяем его и нажимаем «добавить» и **ок.** Теперь эта кнопка появилась на панели инструментов (*Рис.13*). Нажимаем эту кнопку, откроется новое окно в которой будет только «главная кнопочная форма». Нажимая кнопку **создать**, создаем кнопки новых страниц: «таблицы», «запросы», «формы», «отчеты» и «выход». Теперь чтоб создать кнопки на этих новых страницах, надо два раза кликнуть на нужную страницу (например таблицы) и в открывшемся окне нажимаем кнопку **создать**. В открывшемся окне , в строке «Текст» вводим название кнопки. В строке «команда» выбираем «выполнить макрос».

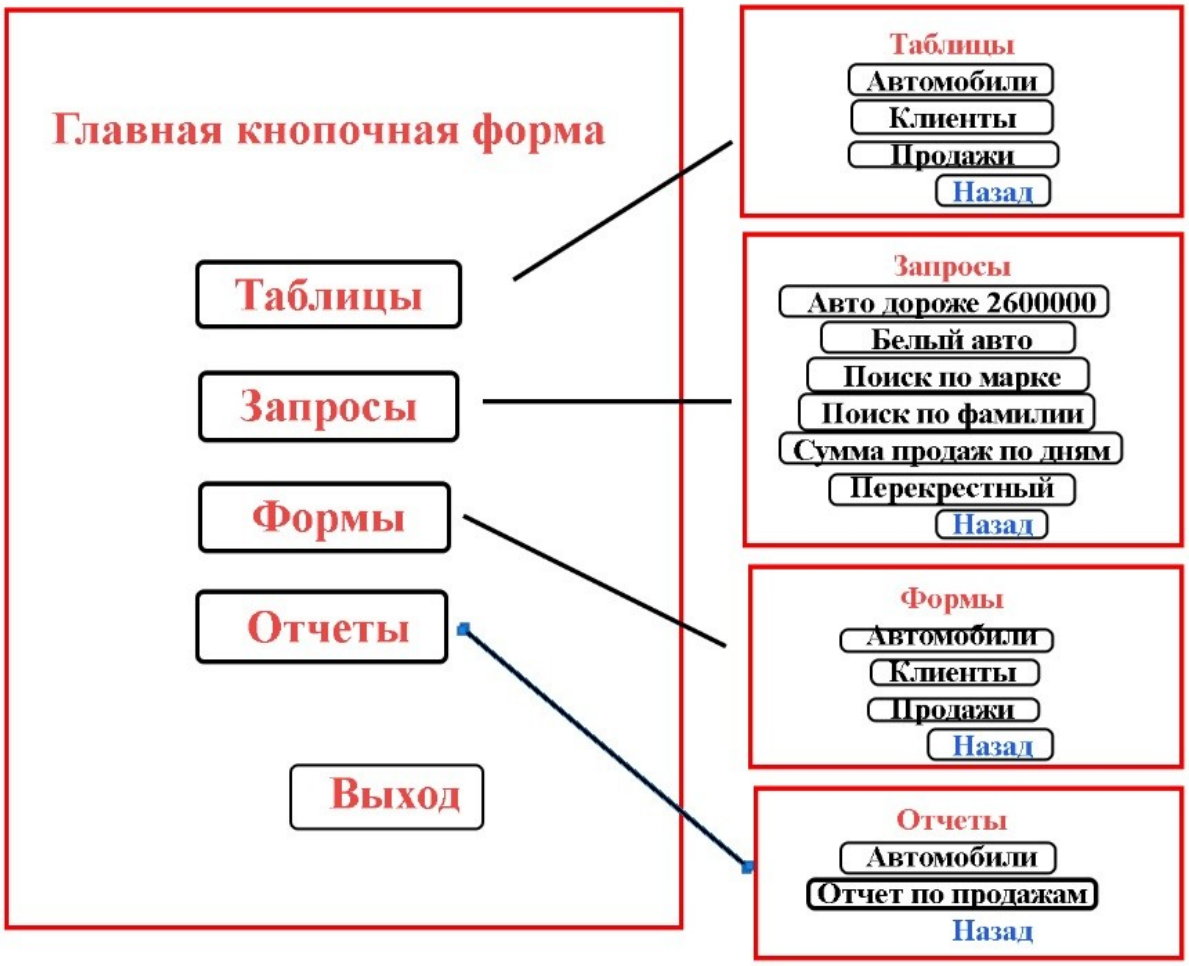

Рис.12 Схема кнопок главной кнопочной формы

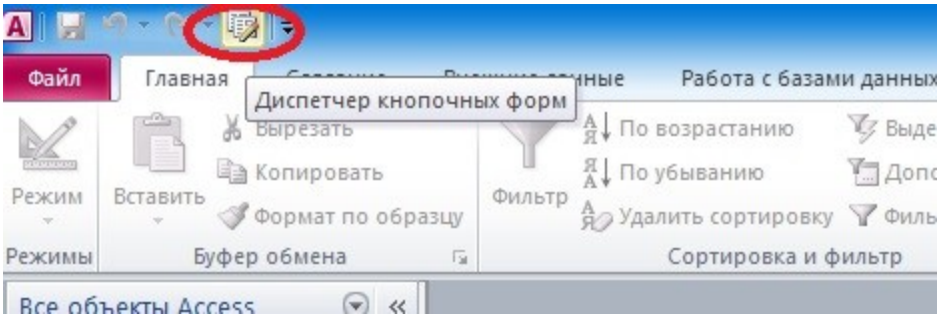

Рис.13 Добавленная кнопка «диспетчер кнопочных форм»

И в строке «макрос» выбираем нужный макрос, который создали ранее (это только для страниц «таблицы» и «запросы») (*Рис.14*). А для страниц «формы» и «отчеты» в строке «команда» выбираем: «открыть форму для добавления» и «открыть отчет» соответственно. А в следующей строке выбираем нужную форму или отчет. Для добавления кнопки «назад» в строке «команда» оставляем «перейти к кнопочной форме», а в строке «кнопочная форма» выбираем «главная кнопочная форма» (*Рис.15*) [8].

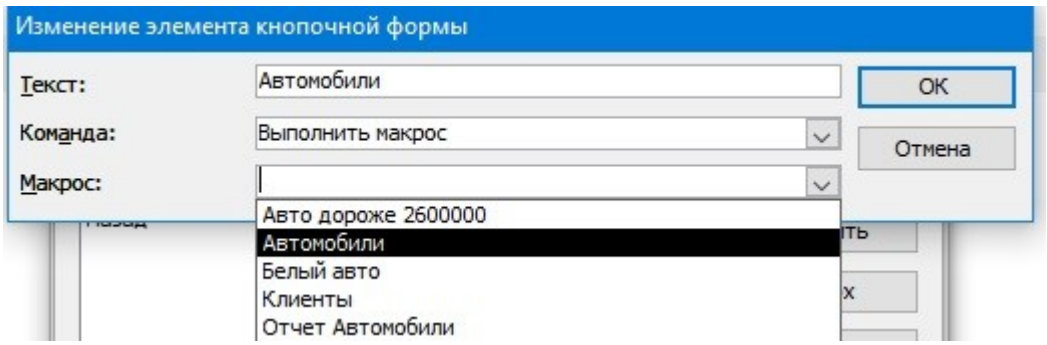

Рис.14 Создание кнопки открытия таблицы автомобили

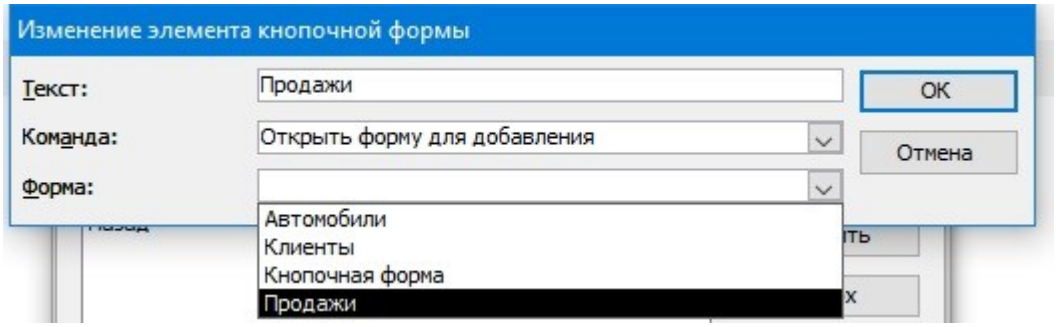

Рис.15 Создание кнопки открытия формы продажи

Для создания кнопок на главной странице нужно заново открыть диспетчер кнопочных форм и дважды кликнуть по **главная кнопочная форма**. Там создаем кнопки: «таблицы», «запросы», «формы», «отчеты» и «выход». В строке «текст» вводим название кнопки. В строке «команда» выбираем «перейти к кнопочной форме». В третей строке выбираем соответствующую страницу.

ЗАКЛЮЧЕНИЕ

Во введении обоснована актуальность выбора темы, определены предмет, объект, цель и соответствующие ей задачи, выявлена проблема и поставлена гипотеза.

В первой части рассмотрены общетеоретические вопросы Основные этапы создания базы данных в среде MS Access. Определяются основные понятия, обуславливается актуальность. Основные этапы создания базы данных в среде MS Access.

Во второй части, практической, рассмотрены те же понятия, на практической основе. Основные этапы создания базы данных в среде MS Access. Результаты исследования могут быть применены на практике.

Для создания базы данных использовали самый простой и достаточный набор операций, который позволил создать: три таблицы, шесть запроса, три формы, два отчёта, девять макросов.

Microsoft Access – это одна из понятных программ в кругу пользователей, позволяющая удобно работать в Базе Данных. Он имеет очень удобный и понятный интерфейс, очень многие операции у него автоматизированы с помощью различных Конструкторов.

Поставленные задачи выполнены, цель достигнута.