

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ  
«МИСиС»  
НОВОТРОИЦКИЙ ФИЛИАЛ

Кафедра электроэнергетики и электротехники

**Л.Г. Казаккулова**

## **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И МЕНЕДЖМЕНТЕ**

Лабораторный практикум для студентов направлений  
38.03.01 "Экономика", 38.03.02 "Менеджмент"

Новотроицк, 2015

УДК 004.658.2  
ББК 32.97  
К 14

**Рецензенты:**

*Ведущий инженер ОИТ, ОО «Орский» филиала «Нижегородский»  
ОА «АЛЬФА-БАНК»  
Д.Ю. Плясунов*

*Зав. кафедрой электроэнергетики и электротехники Новотроицкого филиала ФГАОУ ВПО «Национальный исследовательский технологический университет «МИСиС», доцент, к.т.н.*

***С.Н. Басков***

Казаккулова Л.Г. Информационные технологии в экономике и менеджменте: Лабораторный практикум. — Новотроицк: НФ НИТУ «МИСиС», 2015. — 49с.

Лабораторный практикум содержит теоретические основы, упражнения и задания, предназначенные для изучения возможностей СУБД Microsoft Access. Пособие предназначено для использования на лабораторных и практических занятиях при изучении дисциплин «Информационные технологии в экономике» и «Информационные технологии в менеджменте» для студентов, обучающихся по направлениям 38.03.01 "Экономика" и 38.03.02 "Менеджмент" всех форм обучения.

*Рекомендовано Методическим советом НФ НИТУ «МИСиС»*

© ФГАОУ ВПО  
«Национальный исследовательский  
технологический университет  
«МИСиС»,  
2015

## Содержание

Предисловие .....	4
Лабораторная работа №1. Система управления базами данных. Основные понятия .....	5
Лабораторная работа №2. Создание базы данных в Microsoft Access .....	17
Лабораторная работа №3. Конструирование запросов .....	22
Лабораторная работа №4. Создание форм. Редактирование элементов форм ...	32
Лабораторная работа №5. Подготовка отчетов. Редактирование структуры отчета .....	43
Список литературы .....	48

## Предисловие

Лабораторный практикум является составной частью курса «Информационные технологии в экономике» и может быть использован при проведении практических занятий по дисциплине «Информационные технологии в менеджменте». В связи с разницей в часовой нагрузке у студентов очной и заочной форм обучения рекомендуется для студентов, обучающихся по заочной форме обучения выполнение первых трех лабораторных работ. Лабораторные работы позволяют студентам изучить возможности системы управления базами данных Microsoft Access. Полученные знания позволят лучше понимать работу современных информационных систем поддержки управления предприятием.

Выполнение упражнений и заданий из данного практикума позволит сформировать у студентов соответствующие компетенции.

Для направления 38.03.01 "Экономика":

– ОК-13 – владеет основными методами, способами и средствами получения, хранения, переработки информации, имеет навыки работы с компьютером как средством управления информацией, способен работать с информацией в глобальных компьютерных сетях;

– ПК-4 – способен осуществлять сбор, анализ и обработку данных, необходимых для решения поставленных экономических задач.

Для направления 38.03.02 "Менеджмент ":

– ОК-16 – понимание роли и значения информации и информационных технологий в развитии современного общества и экономических знаний;

– ОК-17 – владеть основными методами, способами и средствами получения, хранения, переработки информации, навыками работы с компьютером как средством управления информацией;

– ОК-18 – способностью работать с информацией в глобальных компьютерных сетях и корпоративных информационных системах.

На лабораторных работах студент должен иметь: данный лабораторный практикум и устройство для хранения результатов работы (флешка).

При выполнении лабораторной работы соблюдайте следующий порядок:

1 изучите теоретический материал и выполните упражнения расположенные в разделе «Теоретическое введение»;

2 выполните задания для самостоятельной работы;

3 ответьте на контрольные вопросы, приведенные в конце каждой лабораторной работы.

Защита лабораторной работы выполняется на текущем или следующем занятии.

Порядок защиты лабораторной работы:

1 предъявите преподавателю выполненные задания по текущей работе;

2 ответьте на контрольные вопросы и вопросы преподавателя.

## Лабораторная работа №1

Тема: Система управления базами данных. Основные понятия.  
Создание базы данных в Microsoft Access

**Цель работы:** приобрести навыки работы в Microsoft Access. Научиться проектировать и создавать базу данных на примере базы данных «Автосалон».

### 1 Теоретическое введение

База данных – это организованная структура, предназначенная для хранения информации.

С помощью специальных программ обслуживания можно сохранить данные в специальной форме, выбрать из них требуемые сведения, обработать, передать коллегам, распечатать и т.д.

#### 1.1 Базы данных. Основные понятия

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

Ядром любой базы данных является модель данных. **Модель данных** — совокупность структур данных и операций их обработки.

СУБД основывается на использовании **иерархической, сетевой или реляционной модели**, на комбинации этих моделей или на некотором их подмножестве .

**Иерархическая структура** представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево). Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей.

**В сетевой структуре** при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

**Реляционные модели** характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных. Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представ-

ляет собой двумерный массив и обладает следующими свойствами:

- каждый элемент таблицы—один элемент данных;
- все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Под проектированием реляционной базы данных понимается принятие решения о количестве таблиц, входящих в эту базу данных, количестве полей входящих в каждую таблицу, типах этих полей, отношениях между таблицами.

Реляционной таблицей можно представить информацию о студентах, обучающихся в вузе. Отношения представлены в виде **таблиц**, строки которых соответствуют **записям**, а столбцы — **полям**. Поле, каждое значение которого однозначно определяет соответствующую запись, **называется простым ключом (ключевым полем)**. Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет **составной ключ**. В примере, таблица 1.1 - **Студент**, ключевым полем таблицы является "**№ личного дела**". Чтобы связать две реляционные таблицы 1.2 и 1.3, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы **внешний ключ** — ключ второй таблицы.

Таблица 1.1 - Студент

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	111
16593	Петрова	Анна	Владимировна	15.03.75	112
16693	Анохин	Андрей	Борисович	14.04.76	111

Таблица 1.2 - Сессия

№ личного дела	Оценка 1	Оценка 2	Оценка 3	Оценка 4	Средний балл
16493	3	4	5	4	4,00
16593	4	5	4	4	4,25
16693	3	4	3	3	3,25

Таблица 1.3 - Стипендия

Средний балл	Процент
4,00	80
4,25	85
3,25	65

Рассмотрим пример реляционной модели, построенной на основе отношений: **Студент**, **Сессия**, **Стипендия**.

Таблицы **Студент** и **Сессия** имеют совпадающие ключи (**№ личного дела**), что дает возможность легко организовать связь между ними. Таблица **Сессия** имеет первичный ключ **№ личного дела** и содержит внешний ключ **Средний балл**, который обеспечивает ее связь с таблицей **Стипендия**, рисунок 1.1.

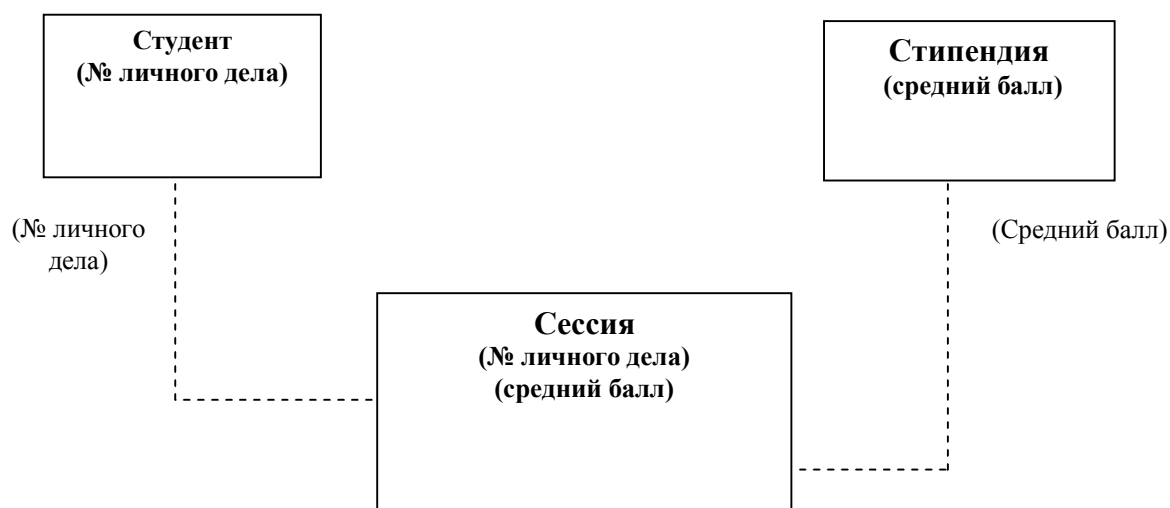


Рисунок 1.1 – Связь таблиц через первичный ключ

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 нам возможность рассмотреть одну систему и обобщить ее понятия, приемы и методы на весь класс СУБД. В качестве такого учебного объекта рассмотрим СУБД **Microsoft Access**, входящую в пакет **Microsoft Office**. **Microsoft Access** поддерживает **реляционную модель** данных.

## 1.2 Создание базы данных

Создание базы данных проводится в несколько этапов.

Этап 1. Определение структуры базы данных. На этом этапе устанавливается, по какому принципу будет располагаться информация (в виде таблиц, файлов и т.п.), какой тип отношений будет между данными данными и т.д. Система управления базами данных поддерживает один из возможных типов моделей данных – сетевую, иерархическую или реляционную.

Этап 2. Разработка средств, для заполнения базы данных справочной и рабочей информацией, заполнение базы данных, определение областей доступа к информации для разных пользователей.

Этап 3. Организация средств поиска и обработки информации базы данных.

Этап 4. Оформление результатов поиска и обработки информации в виде отчетов, писем, диаграмм и т.д.

### 1.3 Структура базы данных

Таблица является основным объектом структуры базы данных. Структура таблицы определяется совокупностью столбцов. В каждой строке таблицы содержится по одному значению в соответствующем столбце. В таблице не может быть двух одинаковых строк. Общее число строк не ограничено.

Столбец – соответствует некоторому элементу данных – атрибуту, который является простейшей структурой данных. Каждый столбец таблицы должен иметь имя соответствующего элемента данных (атрибута). Один или несколько атрибутов, значения которых однозначно идентифицируют строку таблицы, являются ключом таблицы.

Если в базе нет никаких данных (пустая база), то это все равно полноценная база данных. Хотя данных в базе и нет, но информация в ней все-таки есть — это **структура базы**. Она определяет **методы занесения данных и хранения их** в базе.

Базы данных могут содержать различные объекты, но основными объектами любой базы данных являются ее таблицы. Простейшая база данных имеет хотя бы одну таблицу. Соответственно, структура простейшей базы данных тождественно равна структуре ее таблицы.

Структуру двумерной таблицы образуют столбцы и строки. Их аналогами в структуре простейшей базы данных являются **поля и записи**. Если записей в таблице пока нет, значит, ее структура образована только **набором полей**. С изменением состава полей базовой таблицы (или их свойств), изменяется структура базы данных и, соответственно, получается новая база данных.

### 1.4 Свойства полей базы данных

Поля базы данных не просто определяют структуру базы — они еще определяют групповые свойства данных, записываемых в ячейки ячейки, принадлежащие каждому из полей. Основные свойства полей таблиц баз данных:

- 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 в 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 (проверка ошибочности выполняется автоматически, если задано свойство Условие на значение);

– 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 к 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 значения 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. 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. Для данных, представляющих действительные числа, важным свойством является количество знаков после десятичной запятой.

## 1.5 Типы данных

Базы данных Microsoft Access работают с типами данных, представленными в таблице 1.4.

Таблица 1.4 - Описание типов данных

Название типа	Описание типа
1	2
Текстовое	Текст или числа, не требующие проведения расчетов. Размер поля до 255 символов. Устанавливается по умолчанию.
Поле МЕМО	Длинный текст. Размер текста до 65535 символов.
Числовой	Числовые данные, используемые для расчетов. Размер поля 1, 2, 4 или 8 байт в зависимости от свойств поля
Дата время	Дата и время, относящиеся к годам с 100 по 9999. Размер поля 8 байт.

Продолжение таблицы 1.4

1	2
Денежный	Денежные значения и числовые данные, используемые в математических расчетах и содержащие до 15 знаков в целой части и до 4 знаков в дробной. Размер поля 8 байт.
Счетчик	Уникальное, последовательно возрастающее или случайное число, автоматически выводящееся на экран при добавлении записи в базу данных. Размер 4 байта.
Логический	Логические значения, так же как поля, содержащие одно из двух возможных значений, например Да или Нет. Размер поля 1 байт.
Поле объекта OLE	Объект, связанный или внедренный в поле таблицы. Например: рисунок из Paint или текстовый файл Word и т.д. Размер поля до 1 Гбайта.
Гиперссылка	Строка состоящая из букв и цифр, представляющая собой адрес ссылки: текст, путь к файлу, смещение внутри файла и т.д.
Мастер подстановок	Создает поле, в котором предлагается выбор значений из списка или поля со списком, содержащих значения столбца другой таблицы. Выбор этого типа запускает мастер подстановок, который определяет размер и тип ключевого поля

## 2 Упражнения и задания к лабораторной работе

**Упражнение 1.** Спроектировать базу данных «Avto», которая моделирует работу автосалона и содержит сведения о марках легковых автомобилей, продавцах, покупателях и сделках по продаже автомобилей.

Всю информацию, входящую в базу данных, можно разделить на две части: нормативно-справочную (сведения о марках автомобилей, покупателях и продавцах) и рабочую (оперативную) (сведения о сделках) и разместить в четыре таблицы: Автомобили, Продавцы, Покупатели и Заказы.

Таблица «**Автомобили**» должна содержать следующие поля (атрибуты): Код модели, Название модели, Мощность двигателя, Цвет, Количество дверей, Коробка передач, Обивка, Другое оснащение, Заводская цена, Транспортные издержки, Предпродажная подготовка, Специальная модель.

Таблица «**Продавцы**» должна содержать следующие поля (атрибуты): Код продавца, Фамилия, Имя, Отчество, Дата рождения, Дата поступления на работу, Домашний адрес, Телефон.

Таблица «**Покупатели**» должна содержать следующие поля (атрибуты): Код покупателя, Фамилия, Имя, Отчество, Дата рождения, Дата поступления на работу, Домашний адрес, Телефон, Постоянный клиент.

Таблица «**Заказы**» должна содержать следующие поля (атрибуты): Номер заказа, Код модели, Код продавца, Код покупателя, Дата.

При этом таблица Заказы содержит ряд полей (Код модели, Код продавца,



➤ Для **Таблицы 1** определите поля и их свойства в соответствии с таблицей 1.5.

Таблица 1.5 - Структура таблицы «Автомобили» базы данных Avto

Имя поля	Тип данных	Описание	Свойства поля
1	2	3	4
Код автомобиля	Счетчик	Номер автомобиля в базе данных автосалона	Поле определяется как ключевое. Все свойства поля по умолчанию.
Код модели	Числовой	Код модели по каталогу автомобилей	Размер поля: Целое Кол-во десятичных знаков: 0 Обязательное поле: Да Индексированное поле: Да (совпадения не допускаются). Остальные свойства по умолчанию
Название модели	Текстовое	Полное название модели по каталогу автомобилей	Размер поля: 20 Обязательное поле: Да Остальные свойства по умолчанию.
Цвет	Текстовое	Цвет модели по каталогу автомобилей	Размер поля: 20 Обязательное поле: Да Остальные свойства по умолчанию.
Мощность двигателя	Числовой	Мощность двигателя данной модели, лошадиная сила	Размер поля: Целое Кол-во десятичных знаков: 0 Условие на значение: >0 Сообщение об ошибке: Введено отрицательное значение для мощности двигателя! Значение по умолчанию: 40 Обязательное поле: Да Остальные свойства по умолчанию.
Количество дверей	Числовой	Количество дверей данной модели	Размер поля: Целое Кол-во десятичных знаков: 0 Условие на значение: >0 and <6 Сообщение об ошибке: Введено неверное число для количества дверей автомобиля! Значение по умолчанию: 4 Обязательное поле: Да Остальные свойства по умолчанию.

Продолжение таблицы 1.5

1	2	3	4
Коробка передач	Числовой	Количество возможных передач для переключения скорости для данной модели	Размер поля: Целое Кол-во десятичных знаков: 0 Условие на значение: >0 and <7 Сообщение об ошибке: Введено не верное число для количества передач автомобиля! Значение по умолчанию: 6 Обязательное поле: Да Остальные свойства по умолчанию.
Обивка	Текстовое	Обивка салона модели по каталогу автомобилей	Размер поля: 20 Обязательное поле: Да Остальные свойства по умолчанию.
Другое оснащение	Поле МЕМО	Перечисление дополнительного оснащения	Все свойства по умолчанию.
Заводская цена	Денежный	Стоимость автомобиля на заводе изготовителе	Формат поля: Денежный Количество десятичных знаков: 2 Условие на значение: >0 Сообщение об ошибке: Введено отрицательное число для стоимости автомобиля! Обязательное поле: Да Остальные свойства по умолчанию.
Транспортные издержки	Денежный	Затраты на доставку автомобиля в пункт продажи	Формат поля: Денежный Количество десятичных знаков: 2 Условие на значение: >0 Сообщение об ошибке: Введено отрицательное число транспортных издержек! Обязательное поле: Да
Предпродажная подготовка	Денежный	Стоимость предпродажной подготовки автомобиля в пункте продажи	Формат поля: Денежный Количество десятичных знаков: 2 Условие на значение: >0 Сообщение об ошибке: Введено отрицательное число предпродажной подготовки автомобиля! Обязательное поле: Да Остальные свойства по умолчанию.

Продолжение таблицы 1.5

1	2	3	4
Специальная модель	Логический	Указание является ли модель специальной.	Формат поля: Да \ Нет Остальные свойства по умолчанию.

➤ Проектирование таблицы заканчивается установкой ключевого поля. Для этого надо установить курсор в поле **Код автомобиля**, которое предполагается быть ключевым и, выполнить команду **Конструктор / Ключевое поле**.

➤ Сохраните построенную таблицу с именем «Автомобили», выполнив команду **Сохранить**. Перейдите в **Режим таблицы** и закройте таблицу «Автомобили».

➤ Аналогичным образом выполните проектирование таблиц «**Продавцы**» и «**Покупатели**» (создание новой таблицы выполните используя **Создание/ Таблица**). Для таблицы таблиц «**Продавцы**» поля: **Код продавца** (тип поля - счетчик, ключевое поле), **Фамилия, Имя, Отчество, Дата рождения, Телефон, Адрес проживания, Дата поступления на работу**. А для таблицы «**Покупатели**» определите поля: **Код покупателя** (тип поля - счетчик, ключевое поле), **Фамилия, Имя, Отчество, Телефон, Постоянный клиент**.

➤ Выполните проектирование таблицы «**Заказы**». Создайте поля: **Код заказа**(тип поля – счетчик, ключевое поле), **Модель автомобиля, Фамилия продавца, Фамилия клиента**. Таблица «Заказы» содержит поля, которые используют данные из других таблиц Для организации этого процесса используется **мастер подстановок**. Рассмотрим создание поле подстановки для выбора модели автомобиля.

➤ Перейдите в режим Конструктор таблицы «**Заказы**». Установите курсор в столбец **Тип данных** (без указания имени поля) и выберите в раскрывающемся списке **Мастер подстановок**, рисунок 1.3.

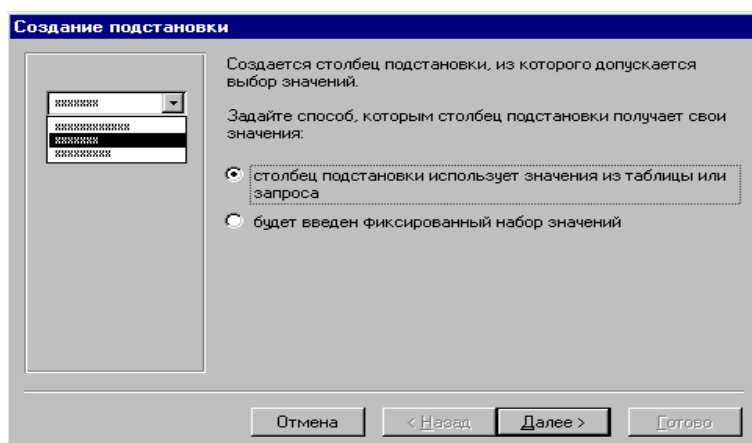


Рисунок 1.3 – Окно мастера подстановок

➤ В данном окне нажмите кнопку "**Далее>**". Установите переключатель в области **Показать** в позицию **Таблицы** и в списке таблиц выбрать таблицу «**Автомобили**», рисунок 1.4.

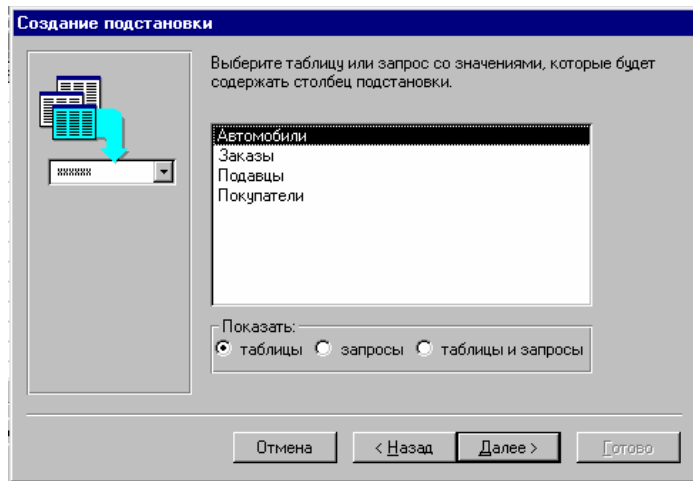


Рисунок 1.4 – Окно мастера подстановок при выборе таблиц

➤ Выбрать поля, которые будут доступны для просмотра в мастере подстановок. Для этого выбрать поле из списка **Доступные поля** и с помощью кнопки со знаком «>» перенести в список **Выбранные поля**. Из таблицы **Автомобили** выбрать поля: **Код модели, Название модели, Мощность двигателя, Цвет, Количество дверей и Заводская цена**. Подтвердить действие кнопкой **Далее**, рисунок 1.5.

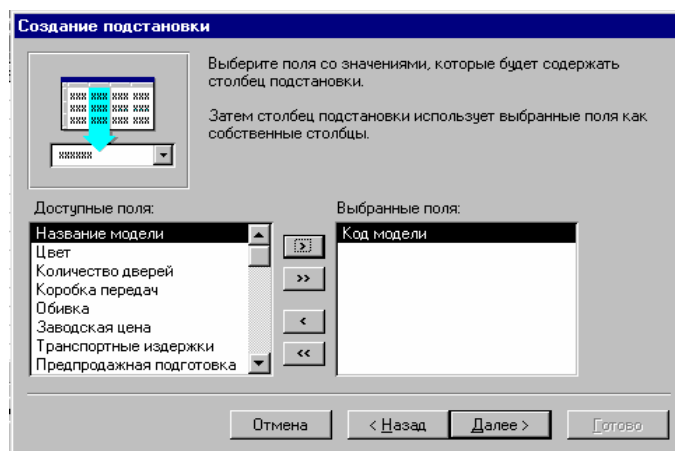


Рисунок 1.5 - Окно мастера подстановок при выборе полей

➤ Установите флажок в поле **Скрыть ключевой столбец** и определите ширину каждого столбца. Для этого установите курсор мыши на границу раздела столбцов и перетащите ее, удерживая левую клавишу мыши. Подтвердите действие кнопкой **Далее**.

➤ Задать подпись для столбца подстановки: **Название модели**. Подтвердить действие кнопкой **Готово**. Обратите внимание на то, что **Имя поля** принимает название ключевого столбца выбранной таблицы и тип этого же поля.

➤ Ключевым полем для таблицы «**Заказы**» должно являться поле **Код заказа**.

**Задание 1.** Для таблицы **Заказы** создать поля для использования данных из таблиц «**Продавцы**» и «**Покупатели**» для полей **Фамилия продавца** и **Фамилия клиента** соответственно. Добавьте в режиме **Конструктор** поле **Дата формирования заказа**.

### **3 Контрольные вопросы**

- 1 Что называется базой данных?
- 2 Какие функции выполняет система управления базами данных?
- 3 Какие основные этапы выполняются при создании базы данных?
- 4 Назовите основные объекты реляционной базы данных.
- 5 Назовите и охарактеризуйте типы данных в СУБД Access.
- 6 В чем заключается проектирование базы данных?
- 7 Какие свойства определяются для различных типов данных?
- 8 Опишите работу мастера подстановок и его назначение.



## Лабораторная работа №2

### Тема: Схема данных

**Цель работы:** научиться создавать связи между таблицами, используя схему данных и редактировать структуру базы данных.

#### 1 Теоретическое введение

##### 1.1 Схема данных. Создание связей

**Схема данных** является графическим отображением базы данных и используется Access в процессе работы с базой данной. Создание схемы данных позволяет упростить конструирование многотабличных форм, запросов и отчетов, а также обеспечить поддержание целостности взаимосвязанных данных при корректировке таблиц.

Схема данных задает структуру базы данных. В ней определяются и запоминаются **связи между таблицами**. Схема данных графически отображается в своем окне, где таблицы представлены списками полей, а связи – линиями между полями разных таблиц. Схема данных ориентирована на работу с таблицами, отвечающих требованию нормализации, между которыми установлены связи: один – ко – многим ( $1: \infty$ ) и один – к – одному ( $1:1$ ). При построении схемы данных Access автоматически определяет по выбранному полю связи между таблицами. Если поле, по которому нужно установить связь, является уникальным ключом как в главной таблице, так и в подчиненной, Access устанавливает связь один – к – одному. Если поле связи является уникальным ключом в главной таблице, а в подчиненной таблице является не ключевым, Access устанавливает связь один – ко многим от главной таблицы к подчиненной.

При выборе в качестве поля связи в главной таблице не ключевого поля Access сообщает, что тип отношения не может быть определен. В этом случае между таблицами можно установить только связь – объединения. Связь объединения обеспечивает объединение записей двух таблиц, имеющих одинаковые значения в поле связи. Причем производится объединение записи из одной таблицы, с каждой записью из другой таблицы при условии равенства значений в поле связи. Результатом объединения записей, который определяется выбором одного из трех способов, может быть:

– объединение только тех записей, в которых связанные поля обеих таблиц совпадают;

– объединение тех записей, в которых связанные поля обеих таблиц совпадают, а также объединение всех записей из первой таблицы, для которых нет связанных полей во второй с пустой записью второй таблицы;

– объединение тех записей, в которых связанные поля обеих таблиц совпадают, а также объединение всех записей из второй таблицы, для которых нет связанных полей в первой с пустой записью первой таблицы.

Для связей типа 1:1 и 1:∞ можно задать параметр обеспечения целостности данных. Обеспечение целостности данных означает выполнение для взаимосвязанных таблиц перечисленных ниже условий корректировки баз данных:

- в подчиненную таблицу не может быть добавлена запись с несуществующим в главной таблице значением ключа связи;
- в главной таблице нельзя удалить запись, если не удалены связанные с ней записи в подчиненной таблице;
- изменение значения ключа связи главной таблицы должно приводить к изменению соответствующих значений в записях подчиненной таблице.

При попытке пользователя нарушить эти условия в операциях добавления или удаления данных в связанных таблицах Access не допускает выполнение операции.

Установление между двумя таблицами связи типа 1:∞ и 1:1 и задание для нее параметров целостности данных возможно только при следующих условиях:

- связываемые поля имеют одинаковый тип, причем имена полей могут быть различными;
- обе таблицы сохраняются в одной базе данных;
- главная таблица связывается с подчиненной по первичному простому или составному ключу (или уникальному индексу) главной таблицы.

Если для выбранной связи обеспечивается целостность данных, можно задать режим каскадного обновления и удаления связанных записей. В режиме каскадного обновления связанных записей при изменении значения в поле связи главной таблицы Access автоматически изменит значения в соответствующем поле в подчиненных записях. В режиме каскадного удаления связанных записей при удалении записи из главной таблицы Access выполняет каскадное удаление подчиненных записей на всех уровнях.

## 1.2 Редактирование структуры базы данных

В процессе разработки структуры базы данных может возникнуть необходимость в ее редактировании. К **редактированию структуры** таблицы можно отнести: **добавление новых полей таблицы, удаление полей таблицы, изменение порядка полей, установку и удаление ключевых полей.**

Для внесения изменений в структуру таблицы надо активизировать имя таблицы в окне **База данных** и выбрать кнопку **Конструктор**.

Для **выделения поля** в режиме конструктора надо установить курсор мыши в крайнюю правую область строки и выполнить одинарный щелчок. Вся строка будет выделена для редактирования. Команда **Правка / Удалить** выполнит удаление строки после соответствующего подтверждения.

Для **выделения последовательного интервала из нескольких строк** надо выделить **первое поле** диапазона и, **удерживая** клавишу **Shift** последнее.

Для **выделения нескольких не смежных строк** надо выделить **первую строку**, а затем, **удерживая** клавишу **Ctrl**, все остальные поля диапазона.

Для **вставки строки**, необходимо поставить курсор на строку, **перед** которой будет вставлена новая строка и выполнить команду **Вставка/Строки**, затем ввести имя поля и определить тип данных этого поля.

### 1.3 Заполнение таблиц базы данных

После создания и проверки структуры базы данных можно приступить к заполнению таблиц данными. Для заполнения таблицы данными надо выделить имя таблицы в окне базы данных и нажать кнопку **Открыть**. При заполнении каждой таблицы следует строго соблюдать тип данных для каждого поля.

**Редактирование записи** в поле может производиться после нажатия клавиши **F2**.

При выборе поля, которое строилось на основе **мастера подстановок**, появляется знак раскрывающегося списка. Значения должны выбираться из этого списка.

## 2 Упражнение к лабораторной работе

**Упражнение 1.** Создать связи между таблицами Автомобиля, Продавцы, Покупатели и Заказы.

➤ При всех закрытых окнах таблиц открыть окно схемы данных с помощью команды **Работа с базами данных / Схема данных**, рисунок 2.1.

Если открытое окно пусто, то выполнить добавление всех таблиц в схему данных. При активной закладке **Таблицы** в списке указаны имена всех таблиц базы данных. Из списка необходимо выбрать таблицу, для которой определяются связи и после выбора кнопки **Добавить** таблица помещается в окно **Схемы данных**. Если для выбранных таблиц определены связи, то они автоматически отображаются. После выбора всех таблиц закрыть окно **Добавить**.

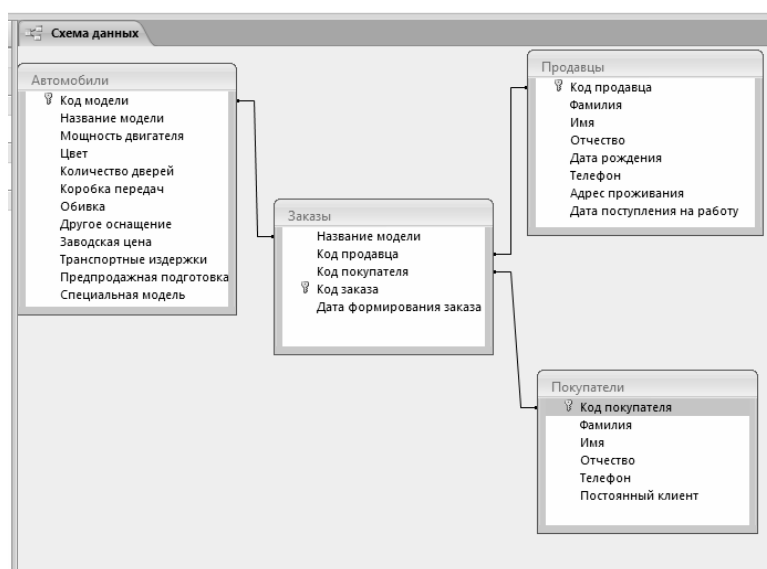


Рисунок 2.1 – Схема данных базы данных Avto

- Выполните изменение существующих связей с 1:1 на 1:∞. Для этого:
  - **выделите** связи для изменения. Выделение связи производится одинарным щелчком левой клавиши мыши на ее изображении;
  - вызовите окно **Связи**. Вызов этого окна выполняется с помощью вкладки **Конструктор/ Изменить связь...** Двойной щелчок клавишей мыши также приводит к открытию этого окна;
  - откройте окно **Связи** для поля **Код покупателя**. Установите флажки в поля **Обеспечение целостности данных**, **Каскадного обновления** и **Удаления данных**. Подтвердите преобразования кнопкой **Ок**, рисунок 2.2.

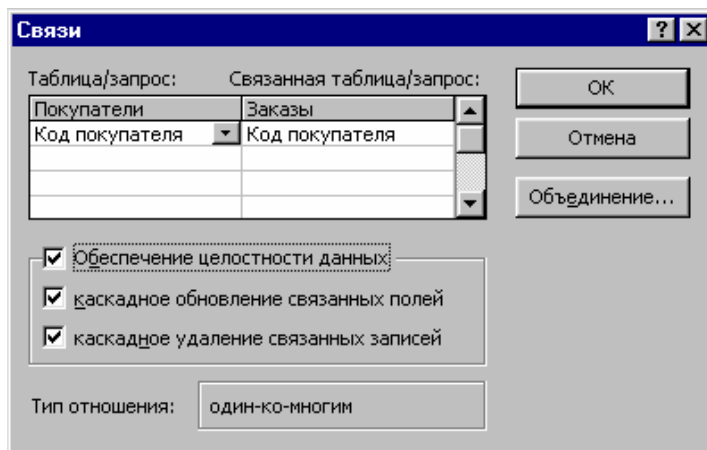


Рисунок 2.2 – Окно изменения связей в базе данных

Обратите внимание на то, что изменилось изображение для измененной связи 1:∞. **1** – обозначает поле в главной таблице, **∞** – в подчиненной таблице, рисунок 2.3.

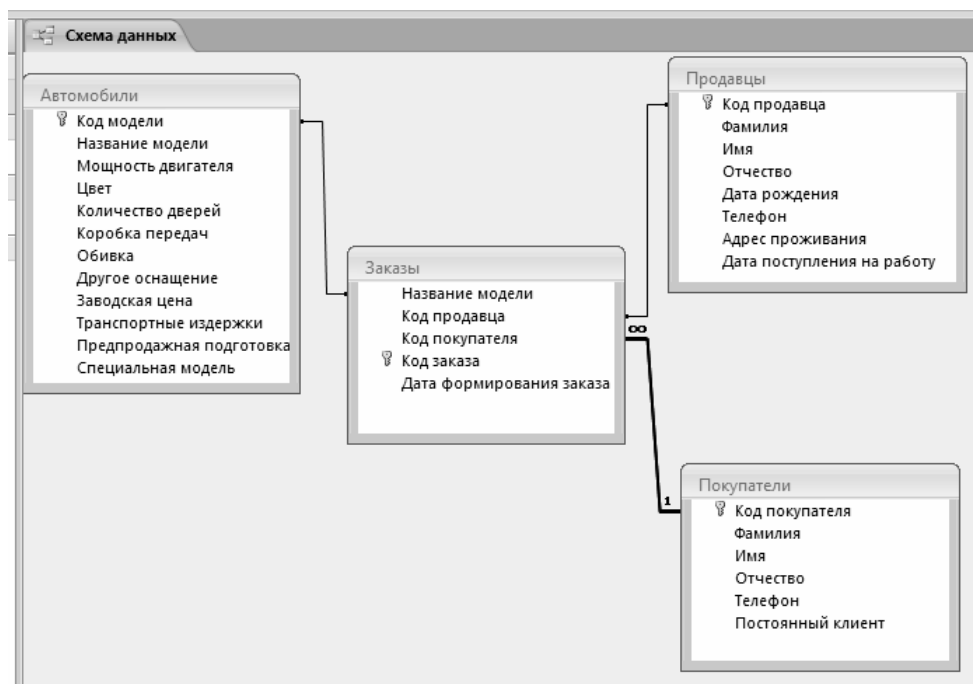


Рисунок 2.3 – Результат изменения связей 1:∞

➤ Выполните удаление связи для поля **Код покупателя**. Выделив связь поля **Код покупателя**, выполнить вызов контекстного меню и команду **Удалить**. Удаление связи будет выполнено после подтверждения действия.

➤ Выполните восстановление удаленной связи.

Для создания связи зафиксировать курсор мыши на поле **Код покупателя** в главной таблице, т.е. на той откуда выбираются данные, и удерживая левую кнопку мыши перетащить указатель на поле **Код покупателя** в подчиненной таблице, т.е. в ту куда помещаются данные. В окне связи установить параметры целостности данных и подтвердить создание связи кнопкой **Создать**.

➤ Закройте окно схемы данных с сохранением всех внесенных изменений.

➤ Войдите в режим конструктора для таблицы **Продавцы** и удалите из таблицы поле **Дата поступления на работу**.

➤ Выполните вставку поля **Количество детей** в таблице **Продавцы** после поля **Отчество**. Выделите поле, следующее за полем **Отчество** и выполните команду **Вставка / Строки**. Введите имя поля и определить тип данных этого поля. Закройте конструктор таблицы с сохранением изменений.

➤ Выполнить заполнение полей таблиц **Автомобили**, **Продавцы**, **Покупатели** с клавиатуры. Каждая таблица должна содержать не менее 10 записей.

➤ Выполнить заполнение таблицы **Заказы**, используя значения поля подстановок.

При выборе поля, которое строилось на основе мастера подстановок, появляется знак раскрывающегося списка. Значения должны выбираться из этого списка. В таблицу **Заказы** занесите не менее 20 записей.

➤ Сохранить данные каждой таблицы.

### 3 Контрольные вопросы

- 1 Опишите назначение схемы данных.
- 2 Как создаются связи между записями?
- 3 Чем отличаются связи типа 1:1 и 1:∞?
- 4 В чем состоит редактирование структуры базы данных?
- 5 В чем заключается процедура заполнения таблиц базы данных?

## Лабораторная работа №3

### Тема: Конструирование запросов

**Цель работы:** приобрести навыки построения запросов средствами СУБД Access.

#### 1 Теоретическое введение

Запрос позволяет выбрать необходимые данные из одной или нескольких взаимосвязанных таблиц, произвести вычисления и получить результат в виде таблицы. Через запрос можно произвести обновление данных в таблицах, добавление или удаление данных. Запрос строится на основе одной или нескольких таблиц базы данных. Кроме того, могут использоваться сохраненные таблицы, полученные в результате выполнения других запросов. Запрос может строиться, непосредственно на другом запросе с использованием его временной таблицы с результатами.

В Microsoft Access имеется удобное для пользователя графическое средство формирования запроса по образцу – QBE (Query By Example), с помощью которого легко может быть построен сложный запрос. Запрос QBE включает подсхему данных с используемыми таблицами и бланк запроса. При конструировании запроса достаточно, работая мышью, выделить и перетащить необходимое поле из таблицы, представленной в схеме данных запроса, в бланк запроса и ввести условия отбора записей.

Средствами запроса можно:

- выбрать записи, удовлетворяющие условиям отбора;
- включить в результирующую таблицу запроса нужные поля;
- произвести вычисления в каждой из полученных записей;
- сгруппировать записи с одинаковыми значениями в одном или нескольких полях и выполнить над ними групповые операции;
- произвести обновление полей в выбранном подмножестве записей;
- создать новую базу данных, используя данные из существующих таблиц;
- удалить выбранное подмножество записей из таблицы базы данных;
- добавить выбранное подмножество записей в другую таблицу.

Последовательное выполнение ряда запросов позволяет решать достаточно сложные задачи, не прибегая к программированию. В Access может создаваться несколько видов запросов: **запрос на выборку** – выбирает данные из таблиц и других запросов, и его результатом является таблица, которая существует до закрытия запроса; **запрос на создание таблицы** – основан на запросе выборки, но, в отличие от него, результат запроса сохраняется в новой таблице; **запросы на обновление, удаление, добавление** – это запросы-действия, в результате выполнения которых изменяется таблицы исходных данных.

**Запрос на выборку** играет особую роль, так как на его основе строятся запросы другого вида. Разработка запроса производится в режиме **Конструктора запросов**.

## 1.1 Операции и функции, используемые при построении запросов

В запросе над полями могут производиться вычисления. Результат вычисления образует вычисляемое поле в таблице, образуемой по запросу. При каждом выполнении запроса производится вычисление с использованием текущих значений полей. При вычислениях могут использоваться арифметические выражения и встроенные функции Access.

При построении запроса часто при записи условий отбора используют встроенные функции Access. Все эти функции представлены в построителе выражений, который может быть запущен с помощью команды контекстного меню **Построить** для строки **Условие отбора**. Данные функции для групповых операции представлены в таблице 3.1.

Таблица 3.1 - Групповые операции запросов

Функция	Результат	Тип поля
<i>Sum</i>	Сумма значений поля	Числовой, Дата/время, Денежный и Счетчик
<i>Avg</i>	Среднее от значений поля	Числовой, Дата/время, Денежный и Счетчик
<i>Min</i>	Наименьшее значение поля	Текстовый, Числовой, Дата/время, Денежный и Счетчик
<i>Max</i>	Наибольшее значение поля	Текстовый, Числовой, Дата/время, Денежный и Счетчик
<i>Count</i>	Число значений поля без учета пустых значений	Текстовый, Числовой, Дата/время, Денежный, Счетчик, Логический, Поле объекта OLE
<i>StDev</i>	Среднеквадратичное отклонение от среднего значения поля	Числовой, Дата/время, Денежный и Счетчик
<i>Var</i>	Дисперсия значений поля	Числовой, Дата/время, Денежный и Счетчик

Примеры использования различных функций приведены в таблицах 3.2, 3.3, 3.4.

Таблица 3.2 - Примеры выражений, использующих в качестве условий отбора текстовые значения

Поле	Выражение	Описание
1	2	3
Пункт Назначения	"Киев"	Отображает заказы на доставку товаров в Киев.
Пункт Назначения	"Киев" Or "Минск"	Использует оператор Or для отображения заказов на доставку товаров в Киев или Минск.

Продолжение таблицы 3.2

1	2	3
Дата Отгрузки	Between #05.01.95# And #10.01.95#	Использует оператор Between...And для отображения заказов на отгрузку товаров не ранее 5-января-95 и не позднее 10-января-95.
Дата Отгрузки	#02.02.95#	Отображает заказы на отгрузку товаров 2-февраля-95.
Страна Доставки	In("Канада", "Великобритания")	Использует оператор In для отображения заказов на доставку товаров в Канаду или Великобританию.
Страна Доставки	Not "США"	Использует оператор Not для отображения заказов на доставку товаров во все страны, за исключением США.
Имя Получателя	Like "С*"	Заказы на доставку товаров клиентам, имена которых начинаются с буквы С.
Название	>="Н"	Отображает заказы на доставку товаров в фирмы, названия которых начинаются с букв, находящихся в диапазоне от Н до Я.
Код Заказа	Right([Код Заказа], 2) ="99"	Использует функцию Right для отображения заказов, код которых заканчивается на 99.
Название	Len([Название]) >Val(30)	Использует функции Len и Val для отображения заказов для фирм, названия которых состоят из более чем 30 символов

Таблица 3.3 - Примеры выражений, которые в качестве условия отбора используют часть значения поля

Поле	Выражение	Отображает
Имя Клиента	Like "С*"	Выполненные заказы для клиентов, имена которых начинаются с буквы С.
Имя Клиента	Like "*Импорт"	Выполненные заказы для клиентов, имена которых заканчиваются на слово "Импорт".
Имя Клиента	Like "[А-Д]*"	Выполненные заказы для клиентов, имена которых начинаются с букв от А до Д.
Имя Клиента	Like "*тр*"	Выполненные заказы для клиентов, имена которых содержат сочетание букв «тр».



Продолжение таблицы 3.3

1	2	3
Имя Клиента	Like "Дом мод?"	Выполненные заказы для клиентов, у которых имя начинается со слова «Дом», а второе слово в имени содержит четыре буквы, причем первые три из них «мод», а четвертая неизвестна.

Таблица 3.4 - Примеры выражений, которые управляют датами

Поле	Выражение	Описание
Дата Исполнения	Between Date( ) And DateAdd ("m", 3, Date( ))	Использует оператор Between...And и функции DateAdd и Date для отображения заказов, требующих исполнения в течение трех месяцев, начиная с сегодняшней даты включительно.
Дата Заказа	< Date( )- 30	Использует функцию Date для отображения заказов, сделанных ранее, чем 30 дней назад.
Дата Заказа	Year([ДатаЗаказа])=1996	Использует функцию Year для отображения заказов, сделанных в 1996 году.
Дата Заказа	DatePart("q", [ДатаЗаказа])=4	Использует функцию DatePart для отображения заказов на четвертый квартал.
Дата Заказа	DateSerial(Year([ДатаЗаказа]), Month([ДатаЗаказа])+1, 1)-1	Использует функции DateSerial, Year и Month для отображения заказов на последний день каждого месяца.
Дата Заказа	Year([ДатаЗаказа])= Year (Now()) And Month ([ДатаЗаказа]) = Month(Now())	Использует функции Year и Month, а также оператор And для отображения заказов текущего года и месяца.

Конкретное значение поля в условии отбора может вводиться непосредственно в бланк запроса или задаваться пользователем в диалоговом окне при его выполнении. Чтобы выводилось диалоговое окно для ввода конкретного значения поля в условия отбора, нужно определить параметры запроса. Порядок определения параметра запроса следующий:

- выполнить построение запроса выбора без указания условий отбора;
- в строке **Условия отбора** для выбранного поля в квадратных скобках надо ввести имя параметра. Например, **[Введите марку машины]**;
- выполнить команду **Запрос / Параметры** и в диалоговом окне **Параметры запроса** в колонке **Параметры** указать то же название, соблюдая регистр букв и знаки препинания. Тип данных определяется автоматически по рассматриваемому полю;

- запустить запрос на выполнение. Сразу после запуска появится диалоговое окно, в котором следует ввести искомое значение поля и подтвердить выполнение запроса кнопкой **Ok**.

## 2 Упражнения и задания к лабораторной работе

**Упражнение 1.** Создать запрос выбора списка постоянных клиентов. Запрос строится на основе таблицы **Покупатели**.

- Открыть базу данных **Avto**.
- Выбрать вкладку **Создание/Другие/ Конструктор запросов**. В открывшемся окне **Добавление таблицы**, рисунок 3.1, выбрать таблицу **Покупатели** и нажать кнопку **Добавить**. Затем кнопкой **Заккрыть** выйти из окна **Добавление таблицы**.

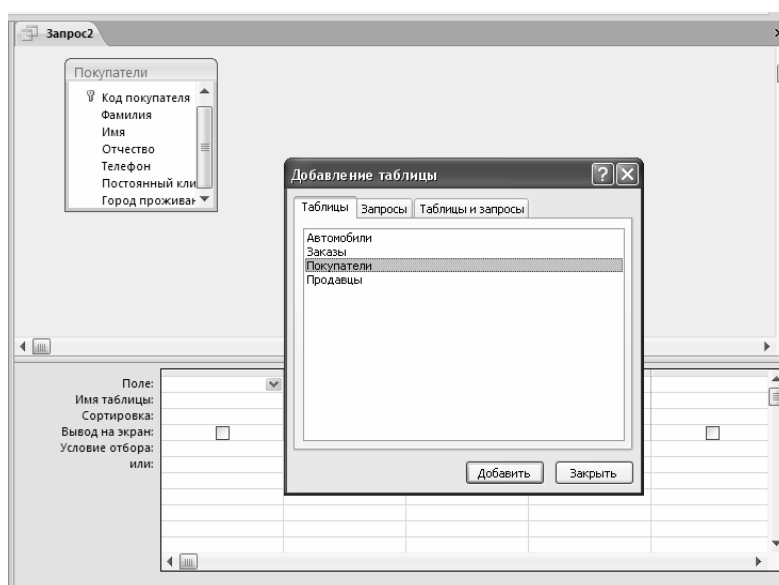


Рисунок 3.1 – Выбор таблиц для построения запроса

➤ В заголовке окна запроса появится строка: **Запрос1**. Окно запроса разделено на две панели. Верхняя панель содержит схему данных запроса, которая включает выбранные для данного запроса таблицы. Таблицы представлены списками полей. Схема данных отображает связи между таблицами, имеющимися в схеме данных базы, и связи объединения, которые устанавливает Access, если таблицы имеют поля с одинаковыми именами и типами данных. Нижняя панель является бланком запроса по образцу, который нужно заполнить. Каждый столбец относится к одному полю, с которым нужно работать в запросе. Поля могут использоваться для включения их в результат выполнения запроса, для задания сортировки по ним и для задания условий отбора записей, без включения в результирующую таблицу.

➤ Для того чтобы поместить поле **Фамилия** из таблицы в нижнюю панель надо зафиксировать курсор мыши на этом поле и перетащить его в первый

столбец конструктора запроса, при этом автоматически указывается имя поля и имя таблицы.

➤ Прodelать аналогичную операцию для полей **Имя**, **Отчество**, **Телефон** и **Постоянный клиент**, рисунок 3.2.

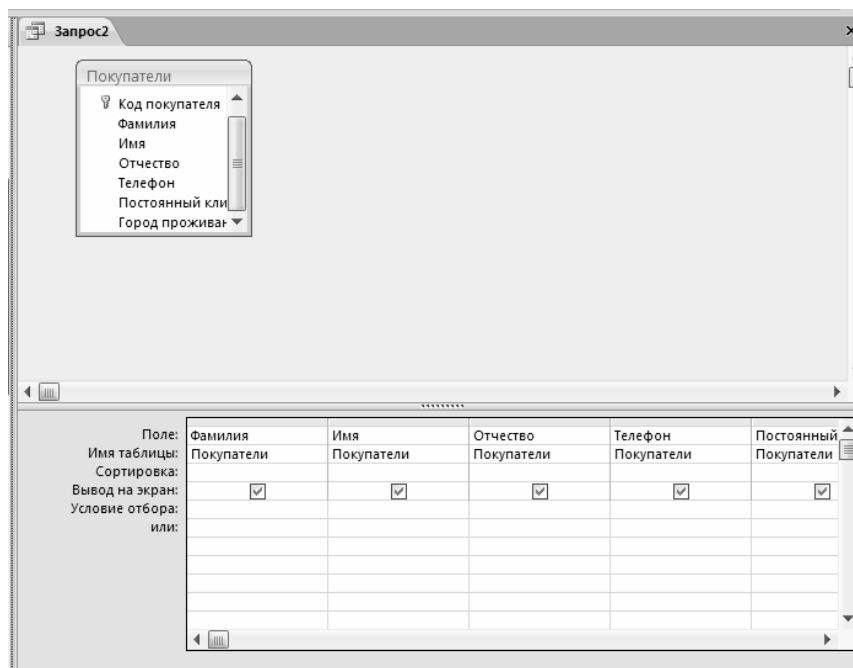


Рисунок 3.2 – Результат перемещения полей

➤ В строке **Сортировка** в поле **Фамилия** установите параметр **По возрастанию**, выбрав его из списка, рисунок 3.3.

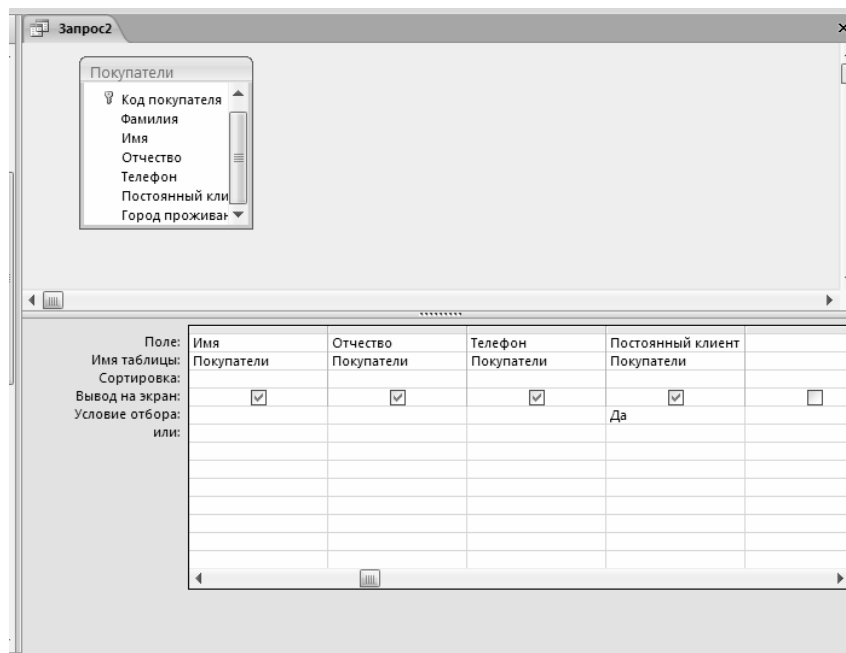


Рисунок 3.3 – Определение условия для поля Постоянный клиент

➤ В строке **Вывод на экран** снят флажок из поля **Постоянный клиент**.

➤ В строке **Условие отбора** в поле **Постоянный клиент** указать в поле **Условие отбора** слово «Да», рисунок 3.

**Упражнение 2.** Изменить запрос **Постоянные клиенты** таким образом, чтобы выбирались только те, которые подавали заказ в 2015 году.

➤ Открыть запрос **Постоянные клиенты** в режиме **Конструктора**.

➤ Добавить в схему данных таблицу **Заказы**. С помощью контекстного меню, вызываемое правой кнопкой мыши, выполните **Добавить таблицу...**, рисунок 3.4, в появившемся окне выберите таблицу **Заказы**. Или из окна с перечнем всех таблиц перетащите с помощью мыши таблицу **Заказы** в поле формирования запроса.

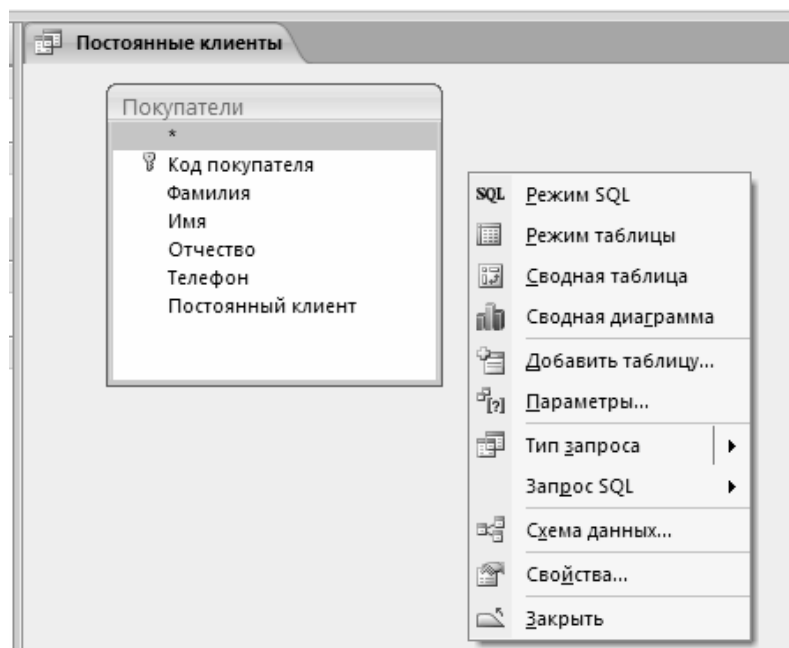


Рисунок 3.4 – Добавление таблицы к схеме данных

➤ Из таблицы **Заказы** перетащить поле **Дата формирования заказа** в последний столбец бланка запроса.

➤ В строке **Условие отбора** с помощью **Построителя выражений (Конструктор/ Построитель)** сформировать формулу: **Year([дата формирования заказа])=2015**

➤ Просмотреть результаты работы такого запроса выполнив команду **Конструктор (работа с запросами)/ Выполнить**.

➤ Сохраните и закройте созданный запрос.

**Упражнение 3.** Создать запрос выбора, который позволяет получить сведения о постоянных клиентах и количестве заказов, сделанных каждым постоянным клиентом.

➤ Создать запрос выбора аналогично заданию 2, не указывая в строке **Условия отбора** никаких данных.

➤ Подсчет количества операций выполняется с помощью групповой операции. Если строка **Групповая операция** отсутствует в бланке запроса, то

необходимо ее включить в режиме **Конструктор (работа с запросами)/ Показать или скрыть/ Итоги**  $\Sigma$ . Затем активизируется строка **Групповая операция**.

➤ В поле **Постоянный клиент** в строке **Групповые операции** из списка выбрать функцию **Count** – подсчет количества записей, удовлетворяющих условию отбора при одинаковых остальных полях. При правильном выполнении задания получится бланк запроса, представленный на рисунок 3.5.

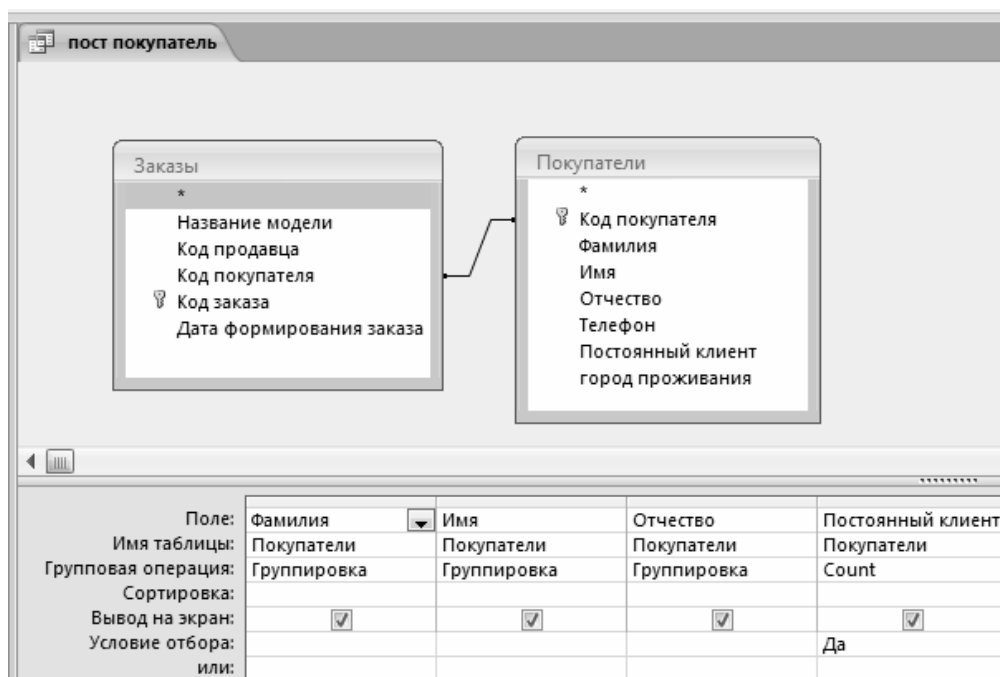


Рисунок 3.5 – Создание запроса по количеству заказов

➤ Для поля **Постоянный клиент** необходимо определить свойства. Определение свойств выполняется с помощью команды **Свойства** из контекстного меню этого поля, вызываемое правой кнопкой мыши. В окне свойств определить: в закладке **Общие – Описание: Количество заказов покупателя; Подпись: Количество заказов**. Закройте окно свойств.

➤ Сохранить запрос с именем **Количество заказов у постоянных клиентов**.

➤ Проследить работу запроса. Закройте запрос.

**Упражнение 4.** Создать запрос, с помощью которого определяется общая стоимость автомобиля в салоне.

Для решения задачи необходимо выполнить следующие действия.

➤ Открыть конструктор запроса и добавить таблицу **Автомобили**.

➤ Из таблицы **Автомобили** поместить в бланк запроса поля: **Название модели, Мощность двигателя, Цвет, Количество дверей, Коробка передач**.

➤ В последнем столбце в строке **Поле** записать формулу для вычисления общей цены автомобиля в виде:

**Цена:[Заводская цена]+[Транспортные издержки] +[Предпродажная подготовка]**

При вводе формулы используйте **Конструктор (работа с запросами)/ Настройка запроса/ Построитель**. Следует внимательно следить за написанием названий полей из таблицы **Автомобили**, вплоть до соблюдения регистров.

➤ Сохранить запрос с именем **Стоимость Автомобилей** и проследить его работу. Закройте запрос.

**Запросы выбора** являются основой для построения запросов различного вида. Для определения запроса необходимо первоначально **создать** запрос выбора с указанными критериями и, находясь в конструкторе запроса, **определить** его вид с помощью команд **Конструктор (работа с запросами)/ Тип запроса**. Тип запроса – это одна из команд: **Выборка, Создание таблицы, Обновление, Удаление**.

Для того чтобы можно было использовать **запросы выбора** необходимо:

– вывести Панель сообщений. Для ее отображения надо перейти на вкладку **Работа с базами данных** и в группе **Отображение** включить параметр **Панель сообщений**;

– включить заблокированное содержимое. На панели сообщений нажать кнопку **Параметры**. Откроется диалоговое окно **Параметры безопасности Microsoft Office**. Установить переключатель в положение **Включить это содержимое** и нажать кнопку **ОК**. Затем снова запустить запрос.

**Упражнение 5.** Создать запрос на удаление всех заказов, которые были оформлены ранее 2015 года.

Для решения задачи надо выполнить следующие шаги.

➤ Создать запрос выбора всех заказов, которые были оформлены ранее 2015 года. Для поля **Заказы** определите условие отбора **Year([Дата формирования заказа])<=2015**, используя построитель выражения.

➤ Проверить правильность отбора записей по этому условию.

➤ Вернуться в **конструктор запроса** и выполнить команду **Конструктор (работа с запросами)/ Тип запроса / Удаление**.

➤ Выполнить запрос и проверить результат его выполнения, открыв таблицу **Заказы**.

**Задание 1.** Выполнить конструирование запросов выбора по следующим критериям:

а) список всех постоянных клиентов, фамилия которых начинается на букву «К»;

б) список всех клиентов, проживающих в городе **Москва** и сделавших заказ в **течение последних двух недель**;

в) список **красных** автомобилей, купленных **постоянными** клиентами за **последние две** недели;

г) список **продавцов** с количеством выполненных **сделок**.

**Задание 2.** Выполнить проектирование запроса, который выполняет расчет наценки на специальную модель в размере 10% от общей стоимости автомобиля.

**Примечание.** Построение запроса выполняется на основе запроса **Стоимость автомобиля** и при проверке условия: является ли автомобиль специальной моделью. Для проверки указанного условия в вычисляемой поле следует ввести формулу:

**Наценка: Pf([Специальная модель]=Истина; [Цена]\*0,01;0)**

В формуле использована функция проверки условия **Pf**. Общий формат записи функции можно записать в виде: **Pf(условие; команда 1;команда 2)**. При истинности условия выполняется команда **1**, в противном случае команда **2**.

В скобках указаны параметры функции:

**[Специальная модель]** – поле таблицы, для которого выполняется проверка. Если используется поле из другой таблицы или в запросе используется несколько таблиц, то необходимо уточнить обращение до вида **[Название таблицы]![Название поля]**.

**Истина** – зарезервированная константа системы Access;

**[Цена]\*10/100** – расчет наценки в размере 10% от стоимости автомобиля;

**0** – отсутствие наценки на не специальную модель.

**Задание 3.** Выполнить конструирование запроса выбора, который определяет стоимость покупаемого автомобиля с учетом наценок на специальную модель и льготы для постоянный клиентов (для постоянных клиентов стоимость автомобиля уменьшается на 10%).

**Задание 4.** Выполнить построение запроса на изменение фамилии покупателя Иванов на Иванова, используя запрос на обновление данных.

**Примечание.** Обратит внимание на то, что после выбора команды **Конструктор (работа с запросами)/ Тип запроса / Обновление** появляется строка **Обновление**, где следует указать новое значение поля.

**Задание 5.** Выполнить построение запроса, который все затраты на предпродажную подготовку, превышающие 10 000 рублей, заменяются на 10 000 рублей.

**Задание 6.** Выполнить поиск продавцов с заданной фамилией.

**Задание 7.** Определить заказы, который заключал продавец с заданной фамилией по заданной модели автомобиля.

### **3 Контрольные вопросы**

- 1 Для чего предназначаются запросы?
- 2 Опишите принцип построения запроса.
- 3 Перечислите возможности запросов.
- 4 Назовите типы запросов и их назначение.
- 5 Какие функции использовались Вами при записи условий отбора?
- 6 Когда используется и как работает функция Pf ?

## Лабораторная работа №4

### Тема: Создание форм. Редактирование элементов форм

**Цель работы:** приобрести навыки создания форм в СУБД Access и работы с ними.

#### 1 Теоретическое введение

В Access 2007 можно вводить данные непосредственно в таблицу в режиме таблица. Но обычно для ввода данных в БД Access используют **формы**. Форма ускоряет работу с базой данных. Форма в БД - это структурированное интерактивное окно с элементами управления, в котором отображаются поля одной или нескольких таблиц или запросов.

**Форму** можно использовать для ввода, изменения или отображения данных из таблицы или запроса. В Microsoft Office Access предусмотрены новые средства, помогающие быстро создавать forms, а также новые типы форм и функциональные возможности.

Формы в БД Access можно создавать с помощью различных средств:

- инструмента Форма;
- инструмента Разделенная форма;
- инструмента Несколько элементов;
- инструмента Пустая форма;
- Мастера форм;
- Конструктора форм.

Все средства создания форм помещены в группу **Формы** на вкладке **Создание**, рисунок 4.1.

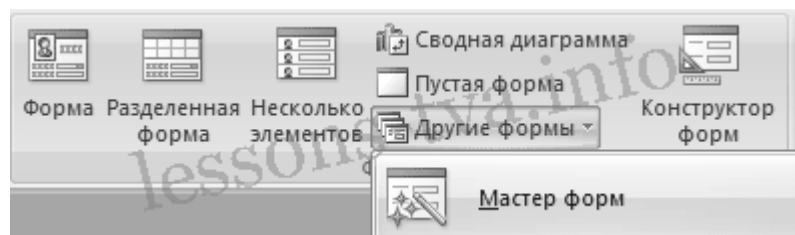


Рисунок 4.1 – Группа Формы, варианты создания форм

Формы, которые используют таблицы целесообразно выполнять с помощью Мастера или указанных инструментов, а дорабатывать их, т.е. вносить необходимые изменения, можно в режиме **макета** или **конструктора**. В Microsoft Access предусмотрено два режима внесения изменений и дополнений в формы: режим макета и режим конструктора. Переход между режимами (открыть, режим макета, конструктор) можно выполнить, щелкнув правой кнопкой мыши форму в области переходов, а затем выбрать нужный режим в контекстном меню.



Режим макета — это более наглядный режим редактирования (изменения) форм, чем режим конструктора. В режиме макета изменения выполняются фактически в реальной форме, поэтому в этом режиме целесообразно выполнять более простые изменения, связанные с ее внешним видом.

В тех случаях, когда в режиме макета невозможно выполнить изменения в форме, целесообразно применять режим конструктора. Режим конструктора предоставляет пользователю более широкие возможности для редактирования (изменения) форм, в этом режиме можно добавлять поля, настраиваемые элементы и составлять программы.

## 1.1 Инструмент "Форма"

Для быстрого создания формы, т.е. создания одним щелчком мыши можно воспользоваться инструментом **Форма**. В этом случае надо выделить таблицу в области объектов. Затем перейти на вкладку **Создание** и щелкнуть на пиктограмме **Форма**. На экране будет отображена **Форма**, рисунок 4.2.

Название модели	Код продавца	Код заказа	Дата формиро
kia soul	Ильина	1	10.04.2013
kia спорт	Сергеев	3	05.09.2012
kia венга	Ильина	4	01.07.2014
*			

Рисунок 4.2 – Результат создания формы

Если Access обнаруживает одну таблицу, связанную отношением «один-ко-многим» с таблицей или запросом, который использовался для создания формы, Access добавляет таблицу данных в форму, основанную на связанной таблице или запросе. Если таблица данных в форме не нужна, ее можно удалить.

## 1.2 Инструмент "Разделенная форма"

**Разделенная форма** — новая возможность в Microsoft Access, которая позволяет одновременно отображать данные в режиме формы и в режиме таблицы. В области объектов (переходов) выделить таблицу, например Покупатели. Далее щелкнуть на пиктограмме **"Разделенная форма"** на вкладке **Создать**. На экране будет отображена форма, рисунок 4.3.

Код покупателя:	2
Фамилия:	Иванов
Имя:	Сергей
Отчество:	Александрович
Телефон:	345271
Постоянный клиент:	<input checked="" type="checkbox"/>
город проживания:	Москва

Код покупа	Фамилия	Имя	Отчество	Телефон	Постоянный клиент	город прож
2	Иванов	Сергей	Александрович	345271	<input checked="" type="checkbox"/>	Москва
3	Иванова	Иван	Иван	123493	<input checked="" type="checkbox"/>	Новотроицк
4	Смирнов	Александр	Петрович	156786	<input checked="" type="checkbox"/>	Москва
5	Алексеев	Юрий	Андреевич	325657	<input checked="" type="checkbox"/>	Москва
6	Миронова	Анна	Семеновна	763249	<input type="checkbox"/>	Орск
(№)					<input type="checkbox"/>	

Рисунок 4.3 – Форма, созданная инструментом "Разделенная форма"

### 1.3 Инструмент "Несколько элементов"

Форму, в которой отображается не одна, а одновременно несколько записей, можно создать инструментом **"Несколько элементов"**, рисунок 4. Чтобы создать данную форму достаточно выделить в области объектов одну из таблиц (например, Покупатели). Затем перейти на вкладку **Создание** и щелкнуть на пиктограмме **"Несколько элементов"**. На экране будет отображена форма, рисунок 4.4, в режиме макета.

Код покупателя	Фамилия	Имя	Отчество	Телефон	Постоянный клиент	город проживания
2	Иванов	Сергей	Александрович	345271	<input checked="" type="checkbox"/>	Москва
3	Иванов	Иван	Иван	123493	<input checked="" type="checkbox"/>	Новотроицк
4	Смирнов	Александр	Петрович	156786	<input checked="" type="checkbox"/>	Москва
5	Алексеев	Юрий	Андреевич	325657	<input checked="" type="checkbox"/>	Москва
6	Миронова	Анна	Семеновна	763249	<input type="checkbox"/>	Орск
(№)					<input type="checkbox"/>	

Рисунок 4.4 - Форма, созданная инструментом "Несколько элементов"

Форма похожа на таблицу, в ней одновременно отображаются несколько записей. Но эта форма предоставляет возможности для настройки, так как она отображается в режиме макета. В режиме макета можно легко осуществлять доработку формы (например, добавлять элементы управления и т.д.).

## 1.4 Инструмент "Пустая форма"

Этот инструмент можно использовать в том случае, если необходимо быстро создать форму с несколькими полями. Форма открывается в режиме **"Работа с макетами форм"** и при этом отображается область **Список полей**, рисунок 4.5.

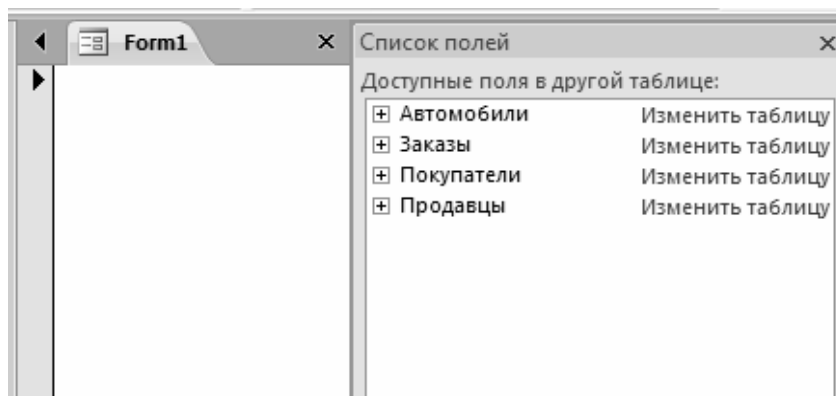


Рисунок 4.5 - Форма, созданная инструментом "Пустая форма"

## 1.5 Инструмент "Мастер форм"

Создание форм при помощи мастера форм осуществляется быстро, и это средство позволяет включить в форму поля из нескольких связанных таблиц или запросов. На вкладке **Создание** в группе **Формы** надо нажать кнопку **Другие формы**, а затем выбрать команду **Мастер форм**. Откроется окно диалога **Создание форм**, в котором необходимо отвечать на вопросы каждого текущего экрана **Мастера** и щелкать на кнопке **Далее**.

В **первом окне** необходимо выбрать поля из источника данных (таблиц или запросов). Для этого надо открыть список **Таблицы и запросы**, щелкнув на кнопку, справа. Например, выберем из списка таблицу **Покупатели**, рисунок 4.6.

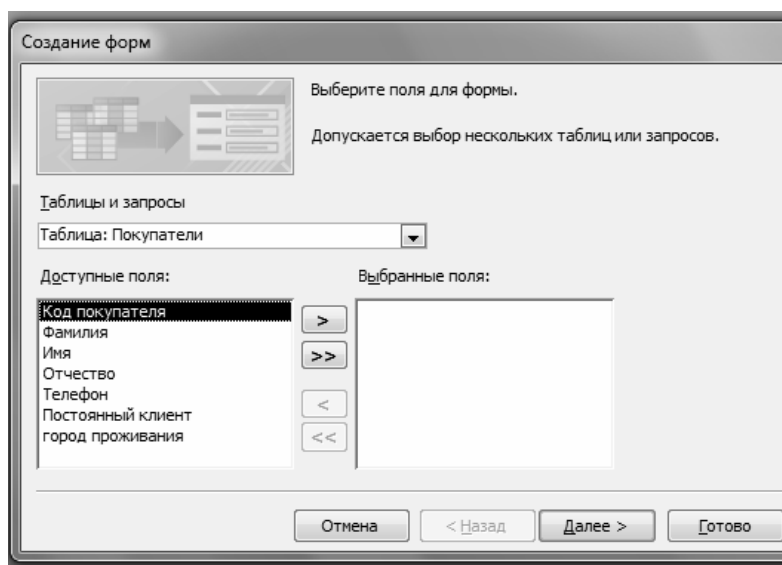


Рисунок 4.6 – Первое окно работы Мастера форм

Затем все "Доступные поля" переведем в "Выбранные поля", выделив их и щелкнув на кнопку >>. Необходимо отметить, что, если Форма создается на основе **нескольких** таблиц, необходимо повторить действия для **каждой таблицы** – источника. Затем необходимо щелкнуть на кнопке **Далее**. В следующем окне надо выбрать внешний вид, например **В один столбец** и щелкнуть **Далее**. В следующем окне выберем требуемый **стиль** – **Официальная**.

После выбора стиля, требуется перейти в последнее окно, щелкнув на кнопке **Далее**. В последнем окне **Мастера** требуется ввести имя формы (например, **Покупатель\_мастер\_форм**) и указать дальнейшие действия: **Открыть форму для просмотра и ввода данных**; **Изменить макет формы**.

После ввода **имени формы** (например, **Покупатели**), выбора режима: «**Открыть форму для просмотра и ввода данных**» и щелчка на кнопке **Готово**, получим следующую форму для ввода и просмотра записей в таблицу **Покупатели**, рисунок 4.7.

Код покупателя	
Фамилия	Иванов
Имя	Сергей
Отчество	Александрович
Телефон	345271
Постоянный клиент	<input checked="" type="checkbox"/>
Город проживания	Москва

Рисунок 4.7 – Форма, созданная с помощью Мастера форм

## 1.6 Инструмент "Конструктор форм"

Для создания новой пустой формы **Покупатели** необходимо выполнить ниже описанные шаги.

Шаг 1. В окне приложения Access выбрать вкладку **Создание**. Выполнить щелчок на пиктограмме "**Конструктор форм**". В окне редактирования появится окно **Form1** с пустой областью данных.

Шаг 2. Для отображения списка полей требуемой таблицы выполнить щелчок на пиктограмме "**Добавить существующие поля**", появится список таблиц. Щелкнув на знак "+" таблицы (например, **Покупатели**), откроется список необходимых полей, рисунок 4.8.

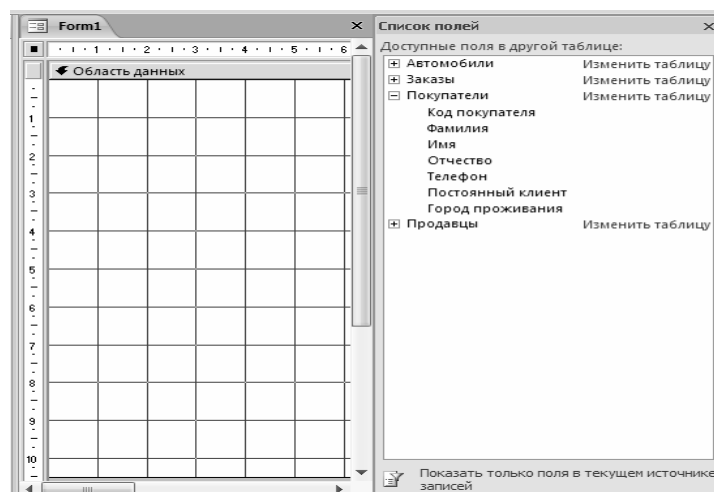


Рисунок 4.8 – Шаг 2, конструктора форм

Шаг 3. Поля из списка следует переместить на форму.

Перемещение полей и их имен **по форме** производится следующим образом:

- выделить поле с именем щелчком мыши. Вокруг него появятся маркеры перемещения и изменения размеров;

- для перемещения поместить указатель мыши на квадратик, находящийся в левом верхнем углу элемента. Указатель мыши в виде четырех направленной стрелки позволяет перемещать объект;

- нажать кнопку мыши и, удерживая ее, буксировать поле или его имя в нужное место в форме. Затем отпустить кнопку мыши;

- для изменения надписи, связанной с полем необходимо выполнить на ней двойной щелчок мышью и выполнить необходимые изменения. Затем закрыть окно;

- для изменения размеров поместить курсор на размерные маркеры, при этом курсор примет вид двунаправленной стрелки. Нажать кнопку мыши, буксировать в нужном направлении, затем отпустить кнопку мыши.

Результат перемещения списка полей на форму представлен на рисунке 4.9.

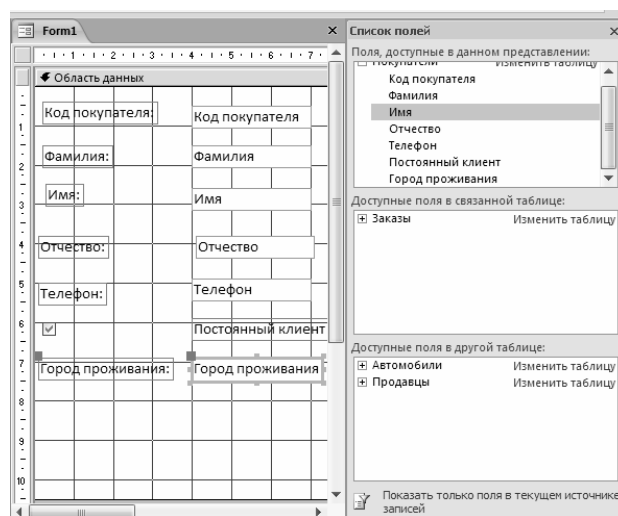


Рисунок 4.9 - Шаг 3, конструктора форм

Для удаления поля выделить его и нажать клавишу **Delete**.

Затем следует сохранить форму и просмотреть ее, выполнив на ней двойной щелчок в области переходов. Получается форма, представленная на рисунке 4.10.

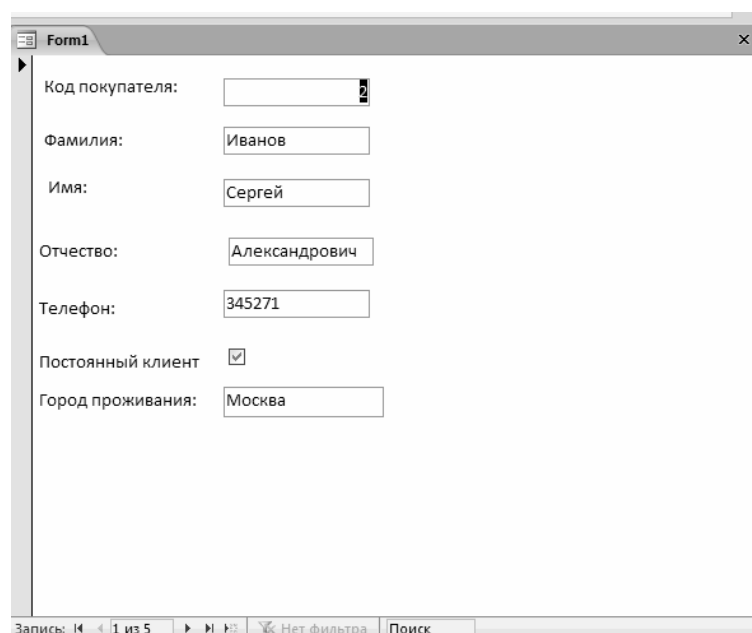
The image shows a screenshot of a software window titled "Form1". The window contains a form with several input fields and a checkbox. The fields are labeled as follows: "Код покупателя:" (empty), "Фамилия:" (Иванов), "Имя:" (Сергей), "Отчество:" (Александрович), "Телефон:" (345271), "Постоянный клиент" (checked), and "Город проживания:" (Москва). At the bottom of the window, there is a status bar with text: "Записи: 1 из 5", "Нет фильтра", and "Поиск".

Рисунок 4.10 - Форма, созданная Конструктором форм

Если вид формы не удовлетворяет, ее можно открыть в режиме Конструктор и внести необходимые изменения, затем сохранить.

## 1.7 Элементы управления формы

При проектировании формы в нее можно вставить управляющие элементы. Речь идет о кнопках, списках, иллюстрациях и пр. Вкладка с управляющими элементами автоматически открывается при переходе в конструктор формы. Каждый управляющий элемент может быть независимым, т.е. несвязанным, или зависимым – связанным.

Для размещения в форме несвязанного управляющего элемента следует выбрать на вкладке **Конструктор (инструменты конструктора форм)** в этот элемент, т.е. выполнить щелчок мышью на изображении этого элемента на панели и разместить этот элемент на прототипе формы, зафиксировав курсор мыши на предполагаемом месте размещения элемента. В дальнейшем можно изменить местоположение этого элемента и его размер.

## 2 Упражнения и задания к лабораторной работе

**Упражнение 1.** Выполнить создание **Формы** для таблицы **Автомобили**.

➤ Выделите таблицу **Автомобили**.

➤ Открыть вкладку **Создание**, группу **Форма**, выберите инструмент **Форма**. Созданная форма по умолчанию содержит все поля таблицы, имеет стандартный белый фон.

➤ При закрытии формы появится запрос на сохранение созданной формы ввести имя **Автомобили**.

**Упражнение 2.** Выполнить создание сложной формы, которая содержит данные об автомобиле и покупателе этой модели.

Согласно условию задачи для построения формы следует использовать составную форму, в которой главной будет являться таблица **Автомобили**, а подчиненной таблица – **Покупатели**. Для решения поставленной задачи воспользоваться **Мастером форм** и выполнить следующие действия.

➤ В окне базы данных выбрать вкладку **Создание /Формы/ Мастер форм**.

➤ В окне **Создание формы** в списке **Таблицы / Запросы** выбрать таблицу **Автомобили** и в списке **Доступные поля** выбрать все поля таблицы.

➤ Изменить в списке **Таблицы / Запросы** таблицу **Автомобили** на таблицу **Покупатели** и выбрать все поля этой таблицы, подтвердить выполненные действия кнопкой **Далее**.

➤ В следующем окне определить тип представления данных: по таблице **Автомобили** (в этом случае главной будет таблица **Автомобили**), проверить активность переключателя **Подчиненные формы**, подтвердить выбор кнопкой **Далее**.

➤ Выбрать внешний вид подчиненной формы, как **ленточный** (установить переключатель к соответствующему названию), подтвердить выбор кнопкой **Далее**.

➤ Определить внешний вид формы, например **Официальная**, подтвердить выбор кнопкой **Далее**.

➤ В последнем диалоговом окне определить название главной формы как **Автомобили и покупатели**, подчиненной – **Подчиненная форма Автомобили**, подтвердить выбор кнопкой **Готово**.

Следует обратить внимание на следующие факты: не все названия полей полностью помещаются в отведенном поле, в подчиненной форме отображаются только те записи, которые связаны с активной записью в главной, в окне базы данных появляются две формы – главная и подчиненная.

**Упражнение 3.** Выполнить корректировку **формы Автомобили** таким образом, чтобы все поля формы располагались в один столбец и все надписи полностью помещались бы в отведенном пространстве.

➤ Активизировать в окне базы данных **форму Автомобили** и нажать кнопку **Конструктор**.

➤ После выполненных действий откроется форма в режиме конструктора, рисунок 4.11.

➤ Установить курсор мыши на верхнюю границу **Примечание формы** и расширить **Область данных**.

➤ Зафиксировать курсор мыши на поле **Предпродажная подготовка** и перетащить его вниз под поле **Транспортная цена**, аналогично выполнить перенос поля **Специальная модель**.

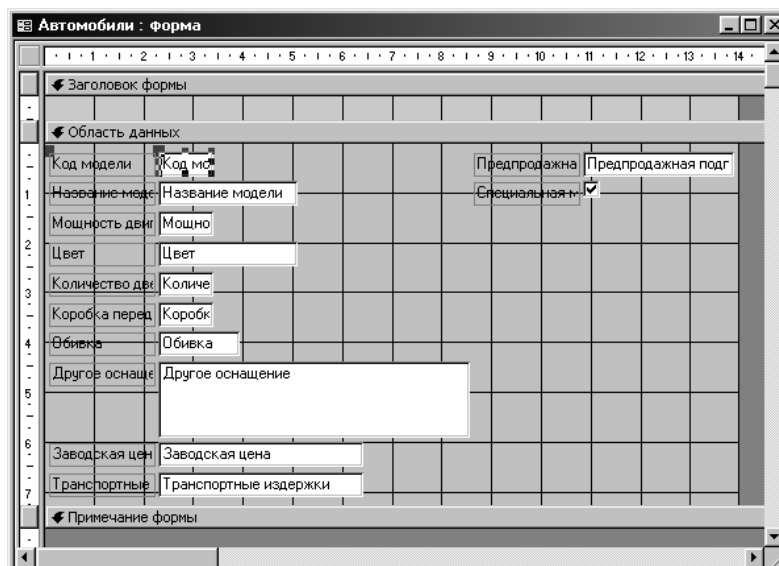


Рисунок 4.11 – Форма Автомобиля

- Установить курсор в левый верхний угол поля **Код модели**, курсор примет форму **Ладони** и перетащить это поле в начало четвертой клетки, аналогично перетащить все поля формы, выстроив их в один столбец.
- Расширить пространство для подписи полей, для этого зафиксировать курсор мыши на правой границе надписи и расширять до тех пор пока надпись полностью не появится в поле надписи.
- После всех преобразований форма должна принять вид, представленный на рисунке 4.12.

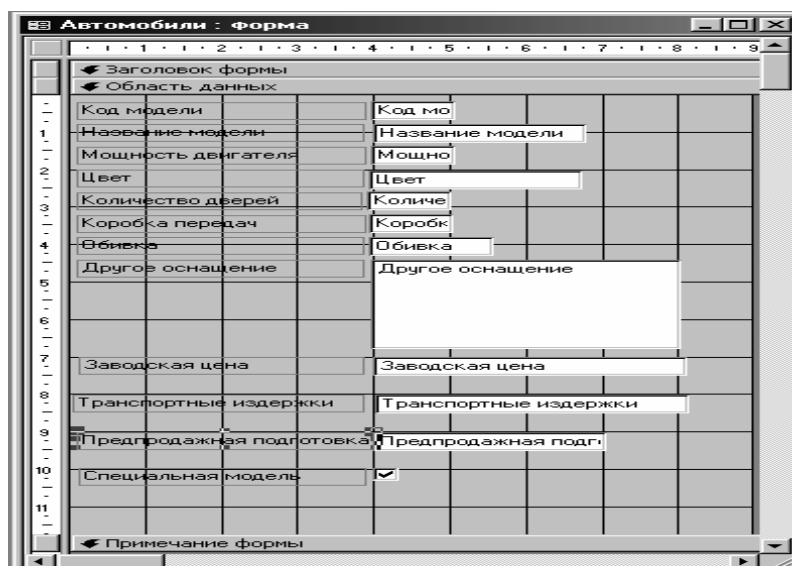


Рисунок 4.12 – Форма Автомобиля после преобразований

- Добавить в каждую таблицу по **пять** записей, используя возможности построенных форм.



**Упражнение 4.** Выполнить вставку вычисляемого поля в форму **Автомобили** для расчета общей стоимости автомобиля.

- Открыть форму **Автомобили** в режиме конструктора.
- Во вкладке **Конструктор** (Инструменты конструктора форм), в группе **Элементы управления** выбрать элемент **Поле** и поместить его после последнего поля, перед таблицей формы;
  - Двумя щелчками кнопкой мыши войти в режим редактирования **надписи поля** и ввести название: **Общая цена**.
  - Щелчком правой кнопки мыши выполнить вызов **контекстного меню для поля** и выбрать пункт **Свойства** или использовать элемент управления **Страница свойств** на вкладке **Конструктор** (Инструменты конструктора форм).
  - В появившемся **Окне свойств** выбрать закладку **Данные** и в поле **Данные** с помощью построителя выражений ввести формулу, используя **Построитель**:

[Заводская цена]+[Транспортные издержки]+[Предпродажная подготовка]


- Открыть закладку **Макет** и определить следующие значения полей: Формат поля – **Денежный**, Цвет текста – **Выделение**; Размер шрифта – **14**; Курсив – **Да**; остальные поля имеют значения по умолчанию.
- В закладке **Другие** определить значение поля **Всплывающая подсказка**:

*заводская цена, транспортные издержки и предпродажная подготовка*

- Для просмотра результата работы перейти в режим просмотра формы.

**Упражнение 5.** Определите всплывающие подсказки для всех полей формы **Автомобили**.

Создание подсказки, всплывающей при помещении указателя на элемент управления:

- Откройте форму в режиме **Конструктора**.
- Выберите элемент управления, и в группе **Сервис** нажмите кнопку  **Страница свойств**.

➤ В окне свойств перейти на вкладку **Другие** введите в поле **Всплывающая подсказка** текст сообщения, которое должно появляться на экране около элемента управления. Максимальная длина сообщения составляет 255 символов.

**Задание 1.** Создать **формы** для ввода данных. По таблицам **Продавцы** и **Заказы** - **Разделенная форма**. По таблице **Покупатели** – форма **Несколько элементов**. Сохранить формы, задав им имена соответственно **Продавцы**, **Заказы**, **Покупатели**.

**Задание 2.** Выполнить построение связанной формы, которая содержит

описание моделей автомобилей, реализованных различными продавцами. При построении формы главной должна быть таблица **Продавцы**, связанной с ней таблица **Автомобили**. Сохранить форму как **Продавцы-Автомобили**.

**Задание 3.** Выполнить построение формы, которая содержит сведения обо всех клиентах, с которыми работал данный продавец. Сохранить форму как **Продавец - Покупатели**

**Примечание:** Для каждой из построенных форм можно сделать несколько замечаний по размещению и размеру полей. Для внесения изменений в структуру формы можно воспользоваться **Конструктором форм**. Конструктор форм содержит ряд инструментальных средств, позволяющих как изменить свойства отдельных элементов формы, так и выполнить добавление и удаление элементов формы..

**Задание 4.** Выполнить корректировку форм для ввода данных в таблицы **Покупатели** и **Продавцы** таким образом, чтобы все поля формы располагались в один столбец и все надписи полностью помещались бы в отведенном пространстве.

**Задание 5.** В форму **Продавцы - Автомобили** добавить кнопку для печати текущей формы.

**Задание 6.** Создать форму для вывода на экран данных по цене автомобиля с учетом **НДС** в размере **18%**.

**Примечание:** Можно использовать формулу  
$$=([\text{Заводская цена}]+[\text{Транспортные издержки}]+[\text{Транспортные издержки}])+(([\text{Заводская цена}]+[\text{Транспортные издержки}]+[\text{Транспортные издержки}])*0,18$$

**Задание 7.** Создать форму на основе таблицы **Автомобили**, которая содержит **специальное поле со списком** моделей автомобилей и в полях формы выводятся данные удовлетворяющие условию этого поля.

**Примечание.** Использовать элемент **Поле со списком** и на первом шаге **обязательно** установить переключатель в позицию **Поиск записи в форме на основе значения, которое содержит поле со списком**.

**Задание 8.** Сконструировать форму по любому из построенных запросов, используя в форме рисунок созданный в любом графическом редакторе.

**Примечание.** Использовать элемент **Рисунок**.

### **3 Контрольные вопросы**

- 1 Для чего предназначены формы?
- 2 Перечислите способы создания форм и опишите каждый способ.
- 3 Как производится редактирование элементов форм?
- 4 Как создать всплывающие подсказки?

## Лабораторная работа №5

Тема: **Подготовка отчетов. Редактирование структуры отчета**

**Цель работы:** приобрести навыки создания отчетов и работы с ними.

### 1 Теоретическое введение

Отчет (report) — это объект базы данных, который используется для вывода на экран, в печать или файл структурированной информации. Отчеты позволяют извлечь из таблиц или запросов базы данных необходимую информацию и представить ее в удобном для восприятия виде. Отчет содержит заголовок, область данных, верхний и нижний колонтитулы, примечание и разбит на страницы.

В Microsoft Access 2007 для создания отчетов можно использовать различные средства, рисунок 5.1:

- мастер отчетов;
- конструктор отчетов;
- инструмент отчет;
- пустой отчет.

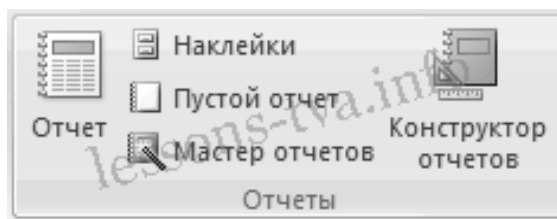


Рисунок 5.1 – Виды отчетов

Отчеты целесообразно создавать с помощью **Мастера отчетов** или других указанных инструментов, а дорабатывать их, т.е. вносить необходимые изменения можно в режиме макета или **Конструктора отчетов**. В Microsoft Access 2007 предусмотрено два режима внесения изменений и дополнений в отчет: режим макета и режим конструктора.

**Режим макета** — это более наглядный режим редактирования и форматирования (изменения) отчетов, чем режим конструктора. В тех случаях, когда в режиме макета невозможно выполнить изменения в отчете, то целесообразно применять режим конструктора.

### 2 Упражнения и задания к лабораторной работе

**Упражнение 1.** Выполнить конструирование отчета, на основе таблицы **Автомобили**, в котором подсчитываются общие суммы по заводской цене на автомобили, по предпродажной подготовке, по транспортным издержкам, рисунок 5.2.

*СПИСОК АВТОМОБИЛЕЙ*

*Сегодня  
(текущая дата)*

<i>Название модели</i>	<i>Заводская цена</i>	<i>Предпродажная подготовка</i>	<i>Транспортные издержки</i>
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Итого \_\_\_\_\_

Рисунок 5.2 – Предполагаемый результат упражнения 1

- Выполните **Создание/ Отчеты/ Конструктор отчетов**.
- В открытом окне **Отчета**, рисунок 5.3, необходимо выполнить размещение основных элементов. Определите скрытость верхнего колонтитула. Выделите **Верхний колонтитул** и установите свойство **Вывод на экран - Нет**, рисунок 5.4;

**Примечание:** Во избежание ошибок выполняйте просмотр отчета после добавления каждого элемента.

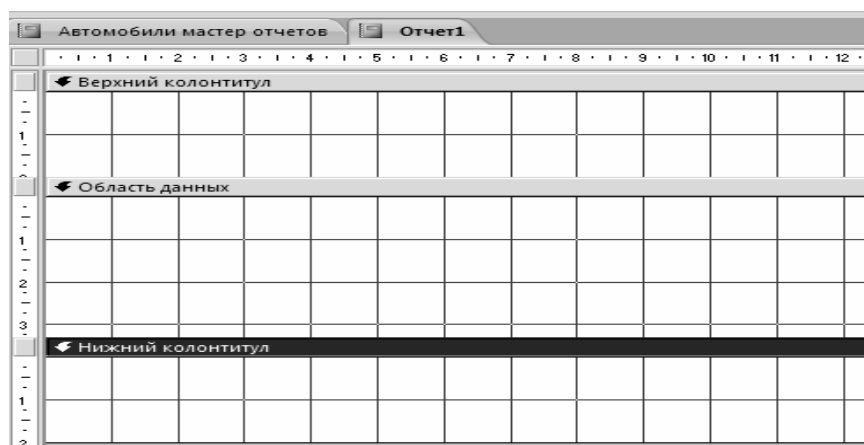


Рисунок 5.3 – Окно конструктора отчетов

- Выполните включение дополнительных разделов отчета – заголовка отчета. С помощью контекстного меню выполните **Заголовок/примечание отчета**. Добавится поле **Заголовок отчета**.
- Добавьте в заголовок отчета название отчета. Во вкладке **Конструктор** в группе **Элементы управления** выберите **Надпись** и разместите ее в заголовке отчета. В надпись ввести текст **СПИСОК АВТОМОБИЛЕЙ**.
- Выполните форматирование надписи. Выделите надпись и в окне свойств надписи установите: **Цвет текста – красный, Шрифт - Times New Roman Cyr, Размер шрифта – 14, Курсив – Да, Оформление – рельефное**.

Растяните рамку надписи до таких размеров, чтобы текст надписи был виден в полном объеме.

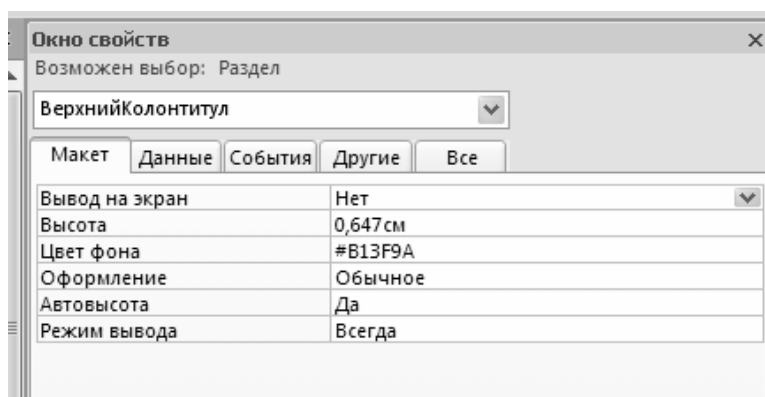


Рисунок 5.4 – Окно свойств верхнего колонтитула

➤ В поле заголовка отчета добавьте текущую дату. Расположите элемент управления **Поле**, из группы **Элементы управления**, поле заголовка отчета. Растяните название поля и поле ввода. В поле ввода введите формулу (используя построитель выражений в окне свойств данного поля, вкладка **Данные**) **=Now()**. В названии поля укажите **Сегодня**, по своему усмотрению установите форматирование шрифта для названия и содержания поля.

➤ Аналогично в область заголовка отчета добавьте надписи: **Название модели**, **Заводская цена**, **Транспортные издержки**, **Предпродажная подготовка** и выполните форматирование по своему усмотрению. Задайте выравнивание – **по правому краю**, рисунок 5.5.

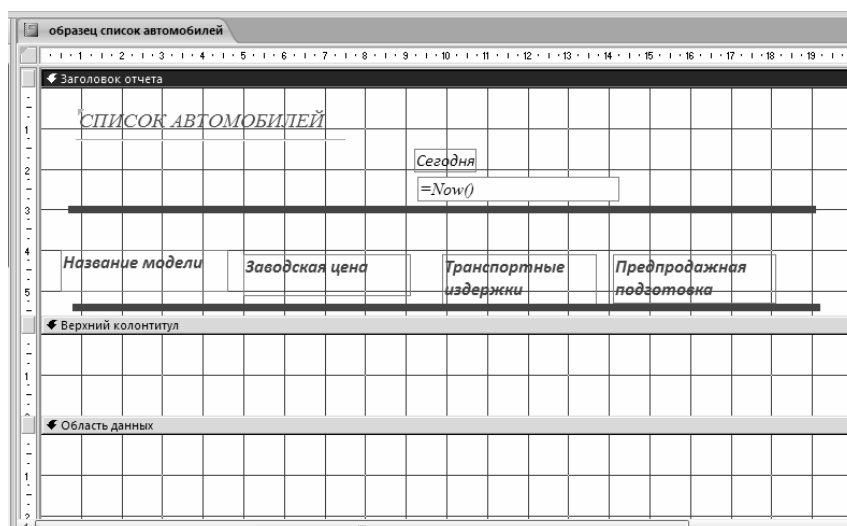


Рисунок 5.5 – Оформление заголовка отчета

➤ Во вкладке **Конструктор** в группе **Элементы управления** выберите инструмент **Линия** и проведите ограничение сверху и снизу заголовка таблицы, установить форматирование линии по своему усмотрению, рисунок 5.5.

➤ С помощью команды **Добавить поля (Конструктор \ Сервис)** вызовите окно, содержащее поля таблицы **Автомобили**. Выполните перетаскивание

соответствующих полей таблицы в **Область данных**. Обратите внимание на то, что при перетаскивании появляется **надпись** и **поле**, выполните выделение **надписи** и удалите ее для каждого из полей с помощью клавиши **Delete**.

- Установите высоту **Области данных** по размеру размещенных полей.

- Выполните вставку номера страниц в нижний колонтитул с помощью элемента управления **Номер страницы**, в окне **Номера страниц** установить переключатель области **Расположения** в положение **Нижний колонтитул**.

- Для подсчета общей суммы по столбцам необходимо в область **Примечание отчета** вставить три свободных поля и в каждое из полей соответственно ввести формулу с помощью **Построителя выражений**: **=Sum([Заводская цена]); =Sum([Транспортные издержки]); =Sum([Предпродажная подготовка])**. Слева от полей расположите **Надпись** и внесите в нее текст **Итого** рисунок 6.

- С помощью окна свойств установите для этих полей: **Формат поля** – **денежный**, **Число десятичных знаков** – **2**.

- В итоге окно **Конструктора отчета** примет вид, представленный на рисунке 5.6. Просмотрите отчет и сохраните его как **Список автомобилей**.

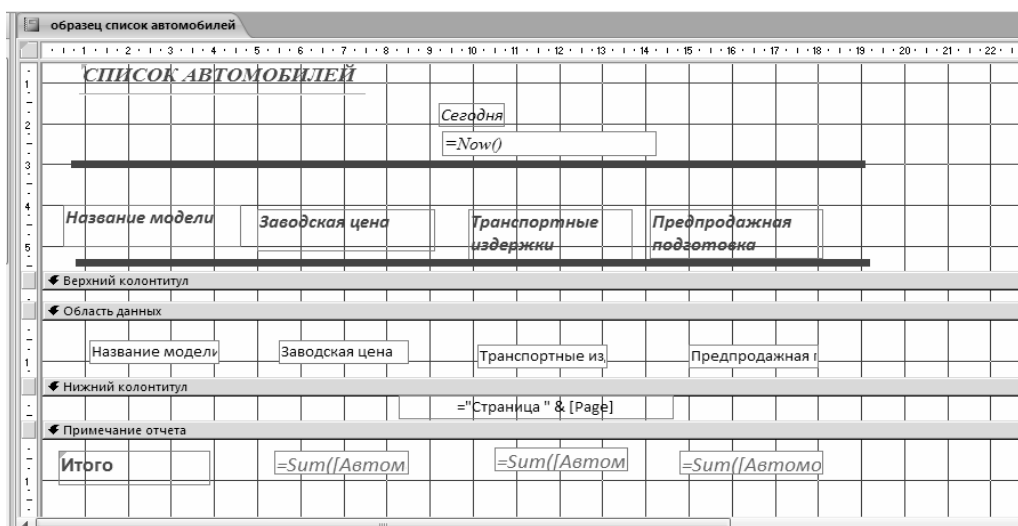


Рисунок 5.6 – Готовый отчет в режиме конструктора

**Задание 1.** Выполните построение отчета, который выполняет расчет итоговой стоимости каждого автомобиля, сохранить как **Итог по стоимости автомобилей**.

**Задание 2.** Выполнить построение отчета по общей сумме льгот при продаже машин постоянным клиентам, сохранить как **Льготы для постоянных клиентов**.

**Задание 3.** Выполнить построение отчета по расчету суммы НДС после продажи автомобилей, сохранить как **НДС**.

**Задание 4.** Рассмотреть работу **Мастера отчетов**. Сконструировать один отчет с его помощью, назвать **Отчет 5**.

**Задание 5.** Создать форму, которая содержит кнопки для демонстрации каждого из построенных отчетов.

### **3 Контрольные вопросы**

- 1 Для чего предназначены отчеты?
- 2 Какие средства СУБД Access можно использовать для создания отчетов?
- 3 Сформулируйте различия между формами и отчетами.
- 4 Что такое макет отчета?
- 5 Чем отличается создание отчета с помощью Мастера отчетов от режима Конструктор?
- 6 Когда и как используются элементы управления?
- 7 Как в режиме Конструктор установить свойства надписи?

## Список литературы

- 1 Базы данных. Microsoft Access 2007. [Электронный ресурс] URL: <http://www.lessons-tva.info/edu/inf-access/access.html> (дата обращения: 10.05.2015)
- 2 Быковец Н.П. Информационные технологии. Лабораторный практикум. [Текст]. Новотроицк: НФ МИСиС, 2007.
- 3 Бекаревич Ю. Б. Самоучитель Access 2010. [Текст]. - СПб.: БХВ-Петербург, 2011.
- 4 Киселев Г.М., Бочков Р.В., Сафонов В.И. Информационные технологии в экономике и управлении. [Текст] - М.: Дашков и К, 2010.



КАЗАККУЛОВА ЛАРИСА ГЕННАДЬЕВНА

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ  
В ЭКОНОМИКЕ И МЕНЕДЖМЕНТЕ**

Лабораторный практикум

для студентов направлений 38.03.01 "Экономика" и 38.03.02 "Менеджмент"  
очной и заочной форм обучения

Подписано в печать 18.11.2015		
Формат 60x90 $\frac{1}{16}$ Рег. № 69	Печать офсетная Тираж 25 экз.	Уч.-изд. л. 3,06

ФГАОУ ВПО

Национальный исследовательский технологический университет «МИСиС»

Новотроицкий филиал

462359, Оренбургская обл., г. Новотроицк, ул. Фрунзе, 8.

E-mail: [nfmisis@yandex.ru](mailto:nfmisis@yandex.ru)

Контактный тел. 8 (3537) 679729.