

Федеральное агентство железнодорожного транспорта
Уральский государственный университет путей сообщения
Кафедра «Информационные технологии и защита информации»

Е. А. Бармина
И. И. Данилина

Использование MS Access 2010 в практических задачах

Учебно-методическое пособие
для студентов направлений подготовки
230100, 090900, 080200, 221000, 190100, 231000, 100100,
220100, 270800, 190700, 280700, 100700, 100400, 080400,
220400, 080100, 190600, 140400 и специальностей 190300,
190901, 271501, 190401

Екатеринбург
Издательство УрГУПС
2013

УДК 681.3 (07)
Б25

Бармина, Е. А.

Б25 Использование MS Access 2010 в практических задачах : учеб.-метод. пособие / Е. А. Бармина, И. И. Данилина. — Екатеринбург : Изд-во УрГУПС, 2013. — 96 с.

Даны основные понятия по теме «СУБД Microsoft Access», предложен лабораторный практикум и варианты заданий к домашним контрольным работам по дисциплинам «Информатика» и «Практикум по информатике», даны рекомендации по их выполнению, рассмотрены примеры.

Пособие может быть использовано как для аудиторных занятий, так и для самостоятельной работы студентов направлений подготовки 230100, 090900, 080200, 221000, 190100, 231000, 100100, 220100, 270800, 190700, 280700, 100700, 100400, 080400, 220400, 080100, 190600, 140400 и специальностей 190300, 190901, 271501, 190401.

УДК 681.3 (07)

Печатается по решению редакционно-издательского совета университета

Авторы:

Е. А. Бармина, ст. преподаватель кафедры «Информационные технологии и защита информации», УрГУПС;

И. И. Данилина, доцент кафедры «Информационные технологии и защита информации», УрГУПС

Рецензенты:

Е. Н. Морозова, доцент кафедры «Информационные технологии и защита информации», канд. экон. наук, УрГУПС;

А. И. Газейкина, доцент кафедры информатики, вычислительной техники и методики обучения информатике Уральского государственного педагогического университета, канд. пед. наук

Учебное издание

Бармина Елена Алексеевна
Данилина Ирина Исаковна

Использование MS Access 2010 в практических задачах

Учебно-методическое пособие

Редактор Н. А. Попова
Верстка О. П. Игнатъевой

Подписано в печать 28.10.2013. Формат 60x84/16. Усл. печ. л. 5,6. Тираж 70 экз. Заказ 74.

Издательство УрГУПС
620034, Екатеринбург, ул. Колмогорова, 66

© Уральский государственный университет
путей сообщения (УрГУПС), 2013

Оглавление

Введение	4
Раздел 1. Основные понятия реляционных баз данных.....	8
Интерфейс MS Access 2010	9
Создание базы данных.....	12
Таблицы	13
Создание таблиц	19
Связи между таблицами	22
Запросы	24
Создание запросов.....	25
Виды запросов.....	27
Вычисления в запросах.....	32
Выполнение групповых расчетов.....	32
Формы	34
Главная и подчиненная формы.....	37
Главная кнопочная форма.....	39
Отчеты.....	42
Раздел 2. Лабораторный практикум	45
Лабораторная работа 1	45
Лабораторная работа 2	49
Лабораторная работа 3	52
Лабораторная работа 4	57
Лабораторная работа 5	63
Лабораторная работа 6	67
Лабораторная работа 7	70
Лабораторная работа 8	73
Лабораторная работа 9	77
Библиографический список.....	79
Приложение 1. Варианты заданий	80
Приложение 2. Домашняя контрольная работа	91
Приложение 3	93

Введение

Современные системы управления базой данных (СУБД) осуществляют хранение и обработку больших объемов информации. Результатами работы этих прикладных программных продуктов являются различные формы представления исходной информации и отчетности, анализ данных по разным параметрам и с достаточной скоростью вычислений.

База данных представляет собой поименованную совокупность данных, организованных по определенным правилам, включающим общие принципы описания, хранения и обработки.

Основой базы данных является модель, которая отражает предметную область в виде совокупности информационных объектов и их связей.

База данных находится под управлением определенной Системы Управления Базой Данных (СУБД), являющейся комплексом программ, предназначенных для обработки данных. Этот комплекс обеспечивает выполнение операций ввода, добавления, удаления, поиска и работы со связями по заданным правилам. Набор правил, определяющих организацию логической структуры хранения данных в базе, получил название модели данных.

Модели данных определяются тремя компонентами:
допустимой организацией данных;
ограничениями целостности;
множеством допустимых операций.

Процедуры хранения данных в БД должны подчиняться общим правилам, среди которых в первую очередь следует выделить:

- 1) целостность и непротиворечивость данных, под которыми понимается как физическая целостность данных, так и предотвращение неверного использования данных, поддержка допустимых значений сочетаний их значений, защита от структурных искажений и несанкционированного доступа;
- 2) минимальная избыточность данных; это означает, что любой элемент данных должен храниться в базе в единственном числе, что позволяет избежать необходимости дублирования операций, производимых с ним и уменьшение требуемого объема памяти.

Каждая БД и СУБД строятся на основе явной или неявной модели данных, т. е. способе представления данных об объектах реального мира и правилах организации этих данных. В теории СУБД выделяют три основные модели: иерархическую, сетевую и реляционную.

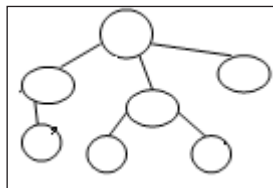


Рис. 1.1

В иерархической модели все элементы связаны отношением подчиненности, и при этом любой элемент может подчиняться только одному какому-либо другому элементу (рис. 1.1).

В базах данных этого типа записи упорядочиваются в определённую последовательность. Поиск данных осуществляется последовательным спуском по уровням. Иерархическая база данных по своей структуре — аналог иерархической файловой системе.

Сетевой подход является расширением иерархического.

В иерархических структурах запись — потомок должна иметь в точности одного предка, а в сетевой структуре потомок может иметь любое число предков (рис. 1.2).

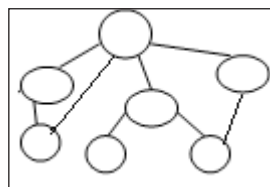


Рис. 1.2

Достоинствами СУБД, основанных на иерархической или сетевой моделях, являются компактность и быстродействие. Недостатками — высокая степень зависимости от конкретных данных.

Иерархическая и сетевая модели данных широко использовались при обработке данных на вычислительных машинах типа ЕС.

В основе реляционной теории БД лежит четко определенная математическая модель. Раздел математики, изучающий отношения, называется реляционной алгеброй. Отсюда и название реляционных баз данных — базы, информация в таблицах которых обрабатывается средствами реляционной алгебры. Реляционная модель данных получила название от английского термина relation — отношение. Её предложил в 70-е годы XX века сотрудник фирмы IBM Эдгар Кодд. При соблюдении определенных условий отношение представляется в виде двухмерной таблицы, привычной для человека (рис. 1.3).

Достоинствами реляционной модели являются простота, удобство реализации, наличие теоретического обоснования и возможность формирования гибкой системы БД, допускающей настройку при формировании запросов.

путь вка : таблица						
	код_тура	назв	Страна пребыв	вид	стоим недели	питание
+	1	Москва	Россия	экск	5 000,00р.	<input checked="" type="checkbox"/>
+	2	Китайская весна	Китай	экстри	15 000,00р.	<input checked="" type="checkbox"/>
▶	3	Страна орхидей	Бали	экск	30 000,00р.	<input type="checkbox"/>
*	0				0,00р.	<input type="checkbox"/>

Рис. 1.3

Реляционная модель данных используется в основном в базах данных среднего размера. При увеличении числа таблиц в базе данных падает скорость работы с ней.

База данных в MS Access представляет собой множество взаимосвязанных таблиц.

В последнее время начинают применяться объектно-ориентированные базы данных. Такие базы основываются на понятии объекта. Объектно-ориентированные базы данных служат для работы с данными в сложных предметных областях, для моделирования которых не хватает функциональности реляционной модели. Также важна была задача сохранения состояния объектов между повторными запусками прикладной программы. Большинство объектно-ориентированных баз данных представляют собой библиотеки, процедуры управления данными, которые включаются в прикладную программу.

Например, системы автоматизированного проектирования, издательские системы и т. д.

На сегодняшний день наиболее часто используются реляционные базы данных. Они наиболее просты и удобны в использовании, хотя и не всегда наиболее эффективны. СУБД Microsoft Access, рассматриваемая в данном пособии, относится именно к этому классу.

СУБД MS Access применяется, если прикладная задача требует хранения и обработки разнородной информации о большом количестве объектов и предполагает возможность многопользовательского режима работы. Примером может служить задача ведения бухгалтерского учета или работы с клиентами.

Основные функции СУБД MS Access — это определение, т. е. описание структуры базы данных, обработка данных и управление данными.

База данных в MS Access обычно содержит совокупность сведений о предметной области (о реальных объектах, процессах, событиях или явлениях). Она организована таким образом, чтобы обеспечить удобное представление этой совокупности, как в целом, так и любой ее части.

Перед началом создания базы данных необходимо продумать выполнение следующих шагов.

1. Проектирование структуры базы данных
 - 1.1. Постановка задачи (какая информация будет храниться в базе данных).
 - 1.2. Определение состава и структуры таблиц (для информации о различных объектах должны быть созданы различные таблицы).
2. Определение связей между таблицами.
3. Наполнение базы данных информацией.
4. Использование базы данных.
 - 4.1. Корректировка данных.
 - 4.2. Поиск нужной информации.
 - 4.3. Анализ информации.
 - 4.4. Представление информации в удобном для пользователя виде.

Раздел 1.

Основные понятия реляционных баз данных

Таблица — информация об однотипных объектах.

Запись — строка таблицы, информация об одном объекте.

Атрибут — столбец таблицы, информация о конкретном признаке всех объектов.

Поле — ячейка таблицы, информация о конкретном признаке конкретного объекта.

Ключевое поле или ключ — поле, однозначно определяющее конкретную запись, ее уникальный идентификатор; в ключевых полях не допускаются совпадения.

Вам предстоит научиться выполнять операции в СУБД Microsoft Access. В первую очередь познакомимся с перечнем объектов, из которых состоит база данных.

Таблицы. Основные объекты базы данных. Создание базы начинается с создания первой таблицы. Таблицы состоят из строк (записей), содержащих сведения об объектах. Каждая таблица хранит информацию об объектах одного вида. Структура таблицы определяется характером информации об этих объектах.

Запросы. Позволяют пользователю выполнять обработку данных, хранящихся в таблицах. Это может быть выборка данных в соответствии с некоторыми условиями или другие операции с данными.

Формы. Используются для просмотра, ввода и редактирования информации в полях базы данных. Они представляют собой более удобный способ просмотра и правки данных в таблицах, чем таблицы.

Отчеты. Предназначены для форматирования, вычисления итогов и печати выбранных данных.

Знакомство с остальными объектами выходит за рамки данного пособия.

Интерфейс MS Access 2010

Пользовательский интерфейс MS Access 2010 включает три основных компонента:

- Лента — полоса в верхней части окна приложения, содержащая группы команд.
- Представление Backstage — набор команд на вкладке Файл на ленте.
- Область навигации, расположенная в левой части окна MS Access, предназначенная для работы с объектами базы данных.

Три этих элемента формируют среду, в которой создаются и используются базы данных. Рассмотрим подробнее эти элементы.

Лента содержит основные вкладки с группами наиболее часто используемых команд, контекстные вкладки, которые появляются, когда их использование допустимо, и панель быстрого доступа — небольшую панель инструментов, на которую можно добавить самые нужные команды.

Лента заменила меню и панели инструментов. Она является основным командным интерфейсом в MS Access 2010. Одно из главных преимуществ ленты состоит в том, что на ней собраны средства выполнения задач, которые раньше находились в меню, на панелях инструментов, в областях задач и других компонентах пользовательского интерфейса. Благодаря этому нужную команду не приходится искать в нескольких разных местах.

При открытии базы данных лента появляется в верхней части главного окна MS Access. На ней отображаются команды активной вкладки команд (рис. 1.4).

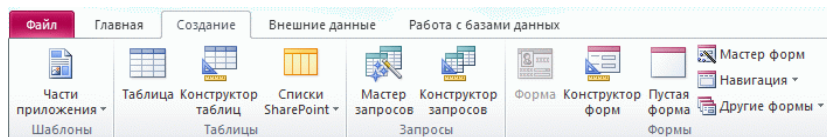


Рис. 1.4

В MS Access 2010 основные вкладки команд — Файл, Главная, Создание, Внешние данные и Работа с базами данных. Каждая вкладка содержит группу связанных команд, которые могут открывать другие новые элементы интерфейса.

На вкладке Главная можно осуществлять выбор режима представления объекта, работу с записями, копирование и вставку данных из буфера обмена, форматирование текста.

На вкладке Создание создание пустой таблицы в режиме конструктора, создание таблицы на основе шаблона, создание списка на сайте SharePoint, создание формы или отчета на основе активной таблицы или запроса.

На вкладке Внешние данные возможен импорт или связывание внешних данных, экспорт данных, запуск диспетчера связанных таблиц.

На вкладке Работа с базами данных выполняется перенос некоторых или всех частей базы данных на новый или существующий сайт SharePoint, создание и просмотр отношений между таблицами, показ или скрытие зависимостей объектов.

Команды ленты также соответствуют объекту, активному в настоящее время. Некоторые вкладки ленты появляются только в определенном контексте. Например, вкладка Конструктор появляется только при открытии объекта в режиме конструктора.

Если необходимо выделить на экране дополнительное пространство для работы, можно свернуть ленту и оставить только строку с вкладками команд. Чтобы скрыть ленту, дважды щелкните активную вкладку команд. Чтобы показать ее, снова дважды щелкните активную вкладку команд.



Рис. 1.5

Панель быстрого доступа (рис. 1.5), находящаяся рядом с лентой, обеспечивает доступ к командам одним щелчком мыши. Набор по умолчанию включает команды Сохранение, Отмена и Возврат. Пользователь сам может настроить панель быстрого доступа, добавляя в нее наиболее часто используемые команды.

Настройка панели быстрого доступа

1. Щелкните стрелку раскрытия списка в правой части панели.
2. В разделе Настройка панели быстрого доступа выберите команду, которую необходимо добавить.
3. Если нужной команды нет в списке, щелкните элемент Другие команды.

На ленте используется элемент управления Коллекция. С помощью этого элемента пользователю предлагается выбрать из предлагаемых наборов команд действия и сразу просмотреть результат выполнения этих команд.

Представление Backstage

Представление Backstage открывается при запуске приложения MS Access. Содержит команды и сведения, применимые ко всей базе данных, такие как создание или открытие базы данных, размещение в Интернете на сервере SharePoint Server, а также команды, которые в более ранних версиях содержались в меню Файл, например команда Печать.

Область навигации

Область навигации (рис. 1.6) позволяет организовывать объекты базы данных и является основным средством открытия или изменения объектов базы данных. Область навигации можно уменьшить или скрыть, но она не загромождается при открытии объектов базы данных поверх нее.

При открытии имеющейся или создании новой базы данных имена объектов базы данных появляются в области навигации. К объектам базы данных относятся таблицы, формы, отчеты, страницы, макросы и модули. Область навигации заменяет окно базы данных, доступное в более ранних версиях MS Access.

Чтобы открыть объект базы данных или применить к нему команду, щелкните его правой кнопкой мыши и выберите команду в контекстном меню. Команды контекстного меню зависят от типа объекта.

Все объекты базы данных в области навигации делятся на категории, которые содержат группы. Некоторые категории являются встроенными, но можно создавать и пользовательские группы.

При необходимости можно отключить отображение области навигации нажатием кнопки в правом верхнем углу области навигации (⏪) или нажав клавишу F11.

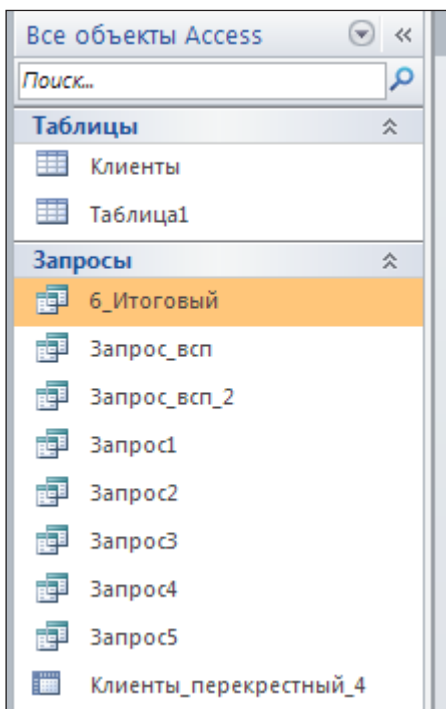


Рис. 1.6

Мини-панель инструментов для форматирования

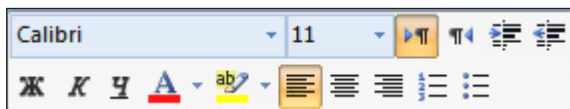


Рис. 1.7

При выделении текста для форматирования над ним автоматически появляется прозрачная мини-панель инструментов (рис. 1.7). При переносе курсора в другое место эта панель автоматически исчезает.

Создание базы данных

Создание базы данных начинается с обращения к представлению Backstage (вкладка Файл). Представление Backstage является первым шагом для создания новых баз данных и работы с уже существующими. Выбрав вкладку Создать, можно увидеть доступные в ней команды.

На этой вкладке отображены методы создания баз данных:

- 1) шаблоны, устанавливаемые вместе с программным продуктом (рис. 1.8);
- 2) шаблоны с сайта Office.com.

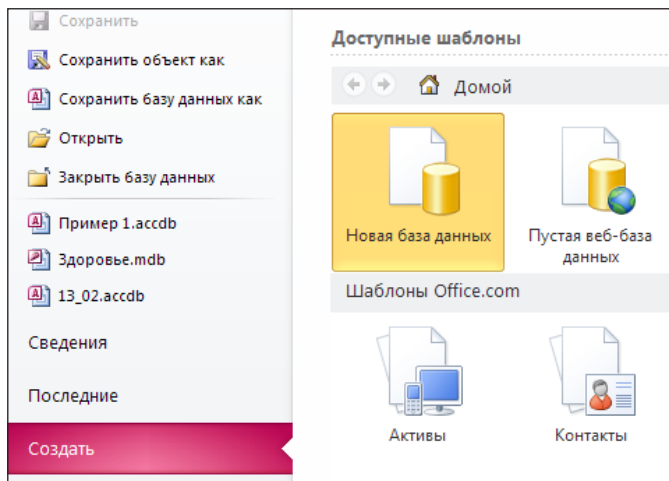


Рис. 1.8

Наиболее простой вариант создания базы данных — с помощью шаблона. Для этого в группе Доступные шаблоны щелкаем элемент Новая база данных:

- 1) справа в разделе Новая база данных в поле Имя файла вводим имя файла базы данных и определяем место хранения этого файла;
- 2) нажимаем кнопку Создать.

Будет создан новый файл базы данных и открыта новая таблица в режиме отображения таблицы.

Создание базы данных начинается с создания таблиц и проектирования связей между соответствующими полями.

Таблицы

Таблица — это объект базы данных, в котором хранятся сведения, относящиеся к конкретной предметной области. Например, можно создать таблицу «Клиенты» для хранения имен, адресов электронной почты и телефонных номеров клиентов фирмы.

Каждая запись содержит данные об одном элементе таблицы, например о конкретном сотруднике. Запись также часто называют строкой. Строка состоит из полей. Каждое поле содержит данные об одном аспекте элемента таблицы, например имя пользователя или адрес электронной почты. В MS ACCESS действуют ограничения на имена полей.

Имя поля:

- 1) должно содержать не более 64 символов;
- 2) может содержать любую комбинацию букв, цифр;
- 3) не может содержать точки (.), восклицательного знака (!), надстрочного символа (^) и прямых скобок ([]);
- 4) не должно начинаться с пробела;
- 5) не должно включать управляющие символы (с кодами ASCII от 0 до 31);
- 6) допускает пробелы, но они могут вызывать конфликты при работе с другими системами;
- 7) не должно совпадать с зарезервированными словами, например, не следует давать полю имя Count, Sum и т. п.;
- 8) должно быть уникальным в пределах таблицы.

База данных может содержать множество таблиц, в которых хранятся данные о различных объектах. Каждая таблица может содержать

множество полей различных типов, таких как текст, числа, даты, гиперссылки и другие.

В ACCESS 2010 при работе с таблицами существуют ограничения.

Таблица 1

Атрибут	Максимальное значение
Число знаков в имени таблицы	64
Число знаков в имени поля	64
Число полей в таблице	255
Число открытых таблиц	2048, включая таблицы, открытые для внутренних потребностей
Размер таблицы	2 ГБ за вычетом места, необходимого системным объектам
Число знаков в текстовом поле	255
Число знаков в поле MEMO	65535 при вводе данных через интерфейс пользователя; 2 ГБ для хранения знаков при программном вводе данных
Размер поля объекта OLE	1 ГБ
Число индексов в таблице	32
Число полей в индексе	10
Число знаков в сообщении об ошибке	255
Число знаков в условии на значение	2048
Число знаков в описании таблицы или поля	255
Число знаков в значении свойства поля	255

Типы данных

У каждого поля есть тип. Тип является свойством поля и задается в конструкторе таблиц. Если перевести курсор мыши в ячейку «Тип поля», то высвечивается список допустимых типов полей, из которого и следует выбрать подходящий для описываемого поля тип. Имя и тип поля должны задаваться обязательно. Графа «Описание» может не заполняться. Эта графа используется в целях документирования проекта.

Тип поля определяет допустимые символы, которые могут быть использованы при его заполнении. Если допущена ошибка в типе данных или неправильно введены данные, то необходимо исправить ошибку, так как программа не даст возможность продолжить работу.

Например, Текстовый тип — это комбинация букв, цифр и других символов. Максимальный размер поля (максимальное количество символов, которое можно ввести в это поле) — 255. Хранятся только

введенные в поле символы; позиции, не использованные в текстовом поле, не хранятся. Для управления максимальным числом вводимых символов определяется свойство Размер поля. По умолчанию размер текстового поля устанавливается 255 символов. Сравнение и упорядочение текстовых данных производится по алфавиту, причем, цифры располагаются раньше букв, а буквы латинского алфавита раньше, чем русского.

Примеры заполнения поля:

Иванов Иван Иванович

2-12-85-06

1000 р.

1234

NuNuCa Nuss-Nougat-Crème.

Возможно автоматическое определение типа данных, если поле создается в режиме таблицы путем ввода данных. В этом случае программа автоматически назначает полю тип данных на основе введенного значения, который впоследствии можно изменить.

Допустимые типы полей в MS ACCESS и их краткая характеристика приведены в таблице 2.

Таблица 2

Тип данных	Значения	Размер
Текст	Алфавитно-цифровые символы Используется для текста или чисел, не при- меняемых в вычислениях	До 255 симво- лов
МЕМО	Алфавитно-цифровые символы (более 255 символов) или текст. Используется для текста, размер которо- го превышает 255 символов, или для текста, в котором используется RTF-форматиро- вание. Поле МЕМО может использоваться, например, для заметок, длинных описаний и абзацев, в которых применяется формати- рование текста, такое как полужирное или курсивное начертание	До 2 гига- байт (2 бай- та на символ). Из них мож- но отображать до 65535 сим- волов в од- ном элементе управления
Число	Целые числовые значения или дробные	1, 2, 4 или 8 байт
Дата/время	Значения даты и времени. Используется для хранения значений дат и времени	8 байт

Тип данных	Значения	Размер
Валюта	Значения денежных сумм. Используется для хранения денежных значений (в рублях или другой валюте)	8 байт
Вычисляемый	Выражение, значением которого может быть текст, число, денежная сумма, дата и время, истина или ложь	Зависит от параметров выражения
Счетчик	Уникальное числовое значение, автоматически вставляемое при добавлении записи. Используется для формирования уникальных значений, которые могут применяться в качестве ключа	4 байта
Логический	Логические значения (да/нет). Можно использовать один из трех форматов: «Да/Нет», «Истина/Ложь» или «Вкл/Выкл»	1 бит
Объект OLE	Объекты OLE и другие двоичные данные. Используется для хранения объектов OLE из других программ Microsoft Windows	До 1 гигабайта
Вложение	Рисунки, изображения, двоичные файлы, файлы Office. Это предпочтительный тип данных для хранения цифровых изображений и двоичных файлов любого типа	2 гигабайта для сжатых вложений
Гиперссылка	Гиперссылки. Используется для хранения гиперссылок, обеспечивающих доступ с помощью одного щелчка к веб-страницам, заданным URL-адресом, или к файлам, заданным UNC-именем. Можно также хранить ссылки на объекты приложения Access, которые хранятся в базе данных	До 2 гигабайт (2 байта на символ). Из них можно отображать до 65535 символов в одном элементе управления
Мастер подстановок	Фактически не является типом данных. Используется для запуска мастера подстановок, позволяющего создавать поле, в котором в виде раскрывающегося списка отображаются значения из другой таблицы, запроса или списка значений	На основе таблицы или запроса: размер присоединенного столбца

В Свойствах поля сетки конструктора отображается список свойств активного поля. Перечень свойств будет зависеть от выбранного типа поля.

Некоторые типы полей имеют схожие наборы свойств полей.

Ниже перечислены основные свойства полей:

- размер поля
- формат поля
- маска ввода
- подпись
- значение по умолчанию
- условия на значения
- сообщение об ошибке
- индексированное поле.

Размер поля

Определяет максимальную длину текстового поля (по умолчанию 50 знаков) или тип данных числового поля. Рекомендуется задавать минимально допустимое значение этого свойства, потому что обработка данных меньшего размера выполняется быстрее. Если вы преобразуете поле в меньшее по размеру, может произойти потеря данных.

Формат поля

Определяет формат вывода значений данного поля.

Маска ввода

Использование масок ввода обеспечивает контроль использования допустимых символов и облегчает процесс ввода данных.

Например. Ввод в таблицу фамилий сотрудников.

Для соответствующего поля можно задать маску ввода. Маска позволит использовать только буквы при вводе, обеспечит преобразование первого символа в верхний регистр, всех остальных — в нижний, и не допустит ошибочного ввода фамилии, состоящей из одной буквы.

В MS ACCESS такая маска ввода будет выглядеть следующим образом:

>L<L??????????

Символ «L» в маске обозначает, что в данную позицию должна быть введена буква, символ «?» обозначает, что в данную позицию может быть введена буква. Символ «>» преобразует все символы, расположенные правее этого знака, к верхнему регистру, символ «<» преобразует все символы, расположенные правее этого знака, к нижнему регистру.

Подпись

Определяет текст, который будет отображаться по умолчанию в надписях для форм, отчетов или запросов.

Условия на значения

Предназначено для определения ограничений. Условия вводятся как выражения. Выражения могут быть простыми или сложными.

ми. Используя их можно задавать и диапазоны. Например, условие: >#1.92#, заданное как «Условие на значения» для поля «ДАТА_ПРИЕМА_НА_РАБОТУ», будет означать, что допустим ввод дат только после 1992 года.

Условия на значения могут задаваться для полей или записей. Выражения, определяющие условия на значения, не должны содержать функции, определяемые пользователем, статистические функции или функции по подмножеству, а также ссылки на формы, запросы и таблицы.

Условия на значения для записей задаются в окне свойств таблицы, вызываемом командой «Свойства» меню «Вид» в режиме конструктора таблицы.

Индексированное поле

Индексированное поле можно использовать для контроля уникальности. В MS ACCESS, как и во многих других системах, при определении для индексированного поля значения свойства «уникальный индекс» в это поле не допускается ввод повторяющихся значений.

Индексированные поля, или индексы, отличаются от обычных полей тем, что для них MS ACCESS создает специальные списки, позволяющие выполнять быструю сортировку и поиск по содержимому индексированного поля. Таблица может содержать несколько индексов.

Специальный индекс называется Ключ (первичный ключ) (рис. 1.9), идентифицирующий записи. Значения такого поля должны быть уникальными. Его наличие не обязательно, но оно так часто используется, что при отсутствии ключа MS ACCESS напоминает об этом и предлагает автоматически добавить ключевое поле.

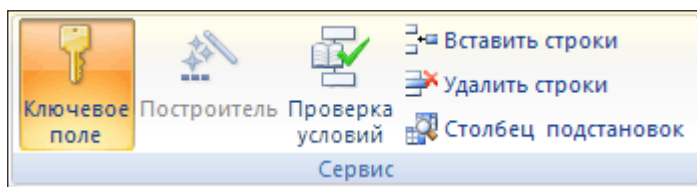


Рис. 1.9

В тех СУБД, которые поддерживают концепцию ключа (в том числе и MS ACCESS), после того как в таблице определяется ключ, по этому полю производится индексирование и запрещается ввод повторяющихся или пустых значений ключа.

При создании таблицы в режиме таблицы MS ACCESS автоматически создает поле первичного ключа с именем «Код» и типом данных «Счетчик».

Изменить или удалить первичный ключ и задать первичный ключ для таблицы, в которой его еще нет, можно в режиме конструктора.

Предполагаемый первичный ключ должен обладать указанными ниже характеристиками:

1. Значение данного поля или сочетания полей должно быть уникальным для каждой записи.
2. Поле или сочетание полей никогда не должно быть пустым или содержать значение NULL.
3. Значения не должны изменяться в процессе работы с БД.

Свойство «Индексированное поле» определяет, надо ли создавать индекс по этому полю. Не допускается создание индексов для полей MEMO, гиперссылок и объектов OLE.

Создание таблиц

Есть несколько режимов создания новой таблицы, отличающихся уровнем автоматизации (рис. 1.10). Независимо от того, в каком режиме начата работа, всегда можно переключиться в другой режим с помощью кнопок режимов в строке состояния окна MS ACCESS.

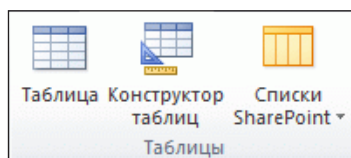


Рис. 1.10

1. Создание новой пустой таблицы

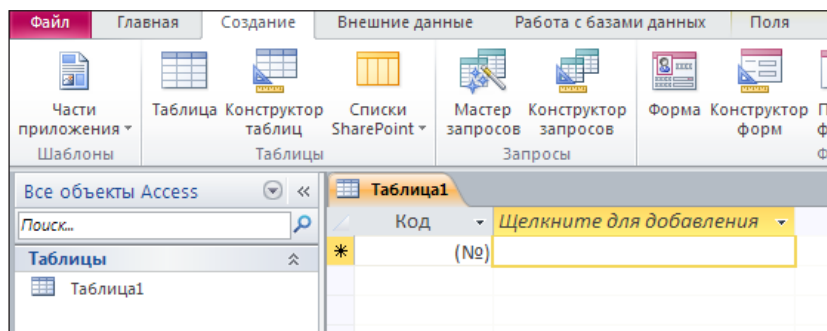


Рис. 1.11

MS ACCESS откроет пустую таблицу (рис. 1.11), в которой отображаются строки и столбцы, куда необходимо ввести данные. Первое поле имеет формальное имя Код. Тип данных поля определяется на основе вводимых данных. Структура таблицы создается при вводе данных. Если закрыть таблицу без сохранения, она будет удалена.

Для переименования столбца (поля) дважды щелкните заголовок столбца и введите новое имя. Для перемещения столбца выделите его, щелкнув заголовок столбца, а затем перетащите в нужное место. Можно также выделить несколько смежных столбцов и перетащить их все в другое место.

2. Создание таблицы в режиме конструктора Наиболее универсальный метод.

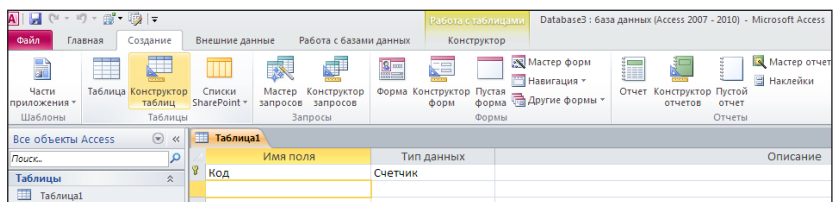


Рис. 1.12

В этом режиме можно самостоятельно задать имена полей, выбрать их тип и настроить свойства. В режиме конструктора (рис. 1.12) сначала следует создать структуру новой таблицы. Затем для ввода данных следует переключиться в режим таблицы. При желании можно ввести в столбце Описание дополнительные сведения для каждого поля. Это описание отображается в строке состояния, если курсор находится в данном поле. После добавления всех полей сохраните таблицу.

3. Создание таблицы на основе списка SharePoint

Создать таблицу можно путем импорта данных, хранящихся в другом месте, или создания связи с ними. Это можно сделать, например, с данными, хранящимися в файле Excel, в списке Windows SharePoint Services, XML-файле, другой базе данных MS ACCESS. Список SharePoint позволяет предоставить доступ к данным пользователям, у которых не установлено приложение MS ACCESS. При импорте данных создается их копия в новой таблице текущей базы данных. Последующие изменения, вносимые в исходные данные, не будут влиять на импортированные данные, и наоборот. Если осуществляется связи-

вание с данными, в текущей базе данных создается связанная таблица, обеспечивающая динамическое подключение к данным, хранящимся в другом месте. Изменения данных в связанной таблице отражаются в источнике, а изменения в источнике — в связанной таблице.

Создание поля подстановки

Поле подстановки предназначено для упрощения ввода данных и предотвращения ошибок.

Поле подстановки позволяет выбрать информацию из другого поля ранее созданной таблицы, что ускоряет процесс ввода и гарантирует правильность написания данных, или создать заранее определенный набор данных, вводимых в конкретное поле.

Имя поля	Тип данных
Отчество	Текстовый
Вид бизнеса	Текстовый
Дата начала работы	Текстовый
Скидка	Поле МЕМО
Место жительства	Числовой
Наличие кредита	Дата/время
	Денежный
	Счетчик
	Логический
	Поле объекта OLE
	Гиперссылка
	Вложение
	Вычисляемый
	Мастер подстановок.

Рис. 1.13

Поле подстановок можно создать, используя Мастер подстановок (рис. 1.13). Для этого:

1. Открыть таблицу в режиме конструктора.
2. В столбце «Тип данных» поля, где будет сделана подстановка, в раскрывающемся списке выбрать «Мастер подстановок».
3. Мастер создает столбец подстановки, в котором отображается список значений для выбора.

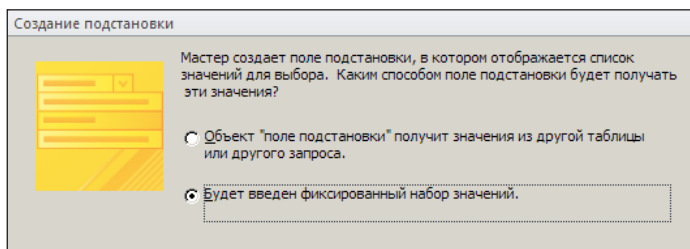


Рис. 1.14

4. Столбец может быть создан:
 из раннее созданной таблицы или запроса;
 из набора фиксированных значений (рис. 1.14).

В первом случае необходимо выбрать объект, из которого делаем подстановку столбца (таблица или запрос). Во втором случае ввести с клавиатуры набор данных, элемент только из этого набора сможет присутствовать в новом столбце.

По умолчанию списки значений используют текстовый тип данных.

В свойствах полей (рис. 1.15) всегда можно посмотреть параметры подстановки.

Общие	Подстановка	
Тип элемента управления	Поле со списком	
Тип источника строк	Список значений	
Источник строк	"крупный"; "средний"; "мелкий"	
Присоединенный столбе	1	

Рис. 1.15

Связи между таблицами

Основные преимущества систем управления базами данных реализуются при работе не с отдельными таблицами, а с группой взаимосвязанных таблиц. Используя связи, создаются запросы, получающие данные из двух или большего числа таблиц.

Для создания связей между таблицами применяют специальное окно Схема данных (рис. 1.16).

Окно Схема данных открывают на вкладке Работа с базами данных в группе Отношения.

Возможны два варианта:

- 1) если ранее при создании таблиц был использован Мастер подстановок, то связи между полями таблиц уже установлены и отображены в окне;
- 2) если никаких связей между таблицами базы не было, то при открытии окна Схема данных открывается окно Добавление таблицы, в котором нужно выбрать таблицы для включения в структуру межтабличных связей. Затем установить связь путем перетаскивания имени ключевого поля из одной таблицы в другую на соответствующее ему связываемое поле.

После построения связей между таблицами для каждой связи открыть окно Изменение связи, в котором можно задать свойства обра-

зованной связи (рис. 1.17). Флажки Каскадное обновление связанных полей и Каскадное удаление связанных записей обеспечивают одновременное обновление или удаление данных во всех подчиненных таблицах при их изменении в главной таблице. Включение флажка Обеспечение целостности данных позволяет защититься от случаев удаления записей из одной таблицы, при которых связанные с ними данные других таблиц останутся без связи.

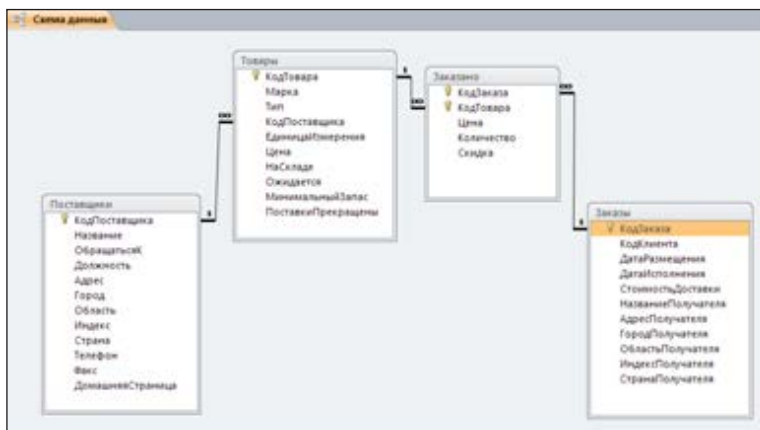


Рис. 1.16

The dialog box 'Изменение связей' (Change Relationships) contains the following information:

- Таблица/запрос:** Тарифы
- Связанная таблица/запрос:** Абоненты
- Primary Key (Таблица/запрос):** КодПлана
- Primary Key (Связанная таблица/запрос):** КодПлана
- Options (all checked):**
 - Обеспечение целостности данных
 - каскадное обновление связанных полей
 - каскадное удаление связанных записей
- Тип отношения:** один-ко-многим
- Buttons:** ОК, Отмена, Объединение..., Новое..

Рис. 1.17

Связи между таблицами отражают структуру связей (отношений) между взаимодействующими объектами и могут быть разделены на три типа.

– Один-к-одному

Каждому объекту одного типа соответствует один и только один объект другого типа. Например, человек и его паспорт. Если хранить какую-либо информацию о человеке в одной таблице, а паспортные данные в другой, то таблицы должны быть связаны именно таким образом. Возникает вопрос, нельзя ли хранить все в одной таблице? Можно, но не всегда удобно. Данные нужны разным людям в разных ситуациях, права доступа к их элементам могут различаться, да и работать с длинными записями не всегда удобно. Ключевые поля у таких таблиц совпадают, и связь осуществляется именно между ключевыми полями.

– Один-ко-многим

Встречается чаще других. При отношении один-ко-многим таблица, находящаяся на стороне отношения «один» называется *главной (адресуемой)* таблицей. Таблица на стороне отношения «многие» называется *внешней (адресующей)*. Каждая запись одной таблицы может быть связана с несколькими (в том числе ни с одной) записями другой таблицы. Примером может служить связь между таблицами **Товары** и **Поставщики**: каждый вид товара имеет только одного поставщика, но поставщик может поставлять несколько видов товаров. При такой связи ключевое поле таблицы на стороне «много» совпадает (и связывается) с не ключевым полем таблицы на стороне «один».

– Многие-ко-многим

Также встречается достаточно часто. Каждая из таблиц может иметь несколько значений в другой таблице. Например, связь между клиентами и товарами: каждый клиент может приобрести много товаров, и каждое наименование товара может быть куплено многими клиентами. Таблицы в СУБД MS Access не могут быть непосредственно связаны между собой по типу «многие-ко-многим». Вместо этого создается дополнительная связывающая таблица и устанавливается две связи «один-ко-многим» между связывающей и исходными таблицами.

Запросы

Основным назначением баз данных является работа с данными, т. е. добавление, поиск, просмотр, изменение, удаление данных. Для этой цели служат запросы. Запрос представляет собой обращение к данным для получения необходимой информации и (или) выпол-

нения действий с данными. Запросы, используемые для извлечения данных из таблицы или выполнения расчетов, называются запросами на выборку. Запросы, используемые для добавления, изменения или удаления данных, называются запросами на изменение.

Запрос получает требуемую информацию из одной таблицы или из двух и более связанных таблиц. Результатом выполнения запроса является новая таблица, поля которой выбираются из одной или нескольких таблиц. С помощью запросов также выполняют преобразование данных по заданному алгоритму и простейшие вычисления. Запросы, так же как таблицы, служат основой для форм и отчетов.

Создание запросов

Создание запроса к базе начинается с обращения к вкладке Создание окна База данных. Выбрав эту вкладку, можно увидеть доступные в ней команды.

На этой вкладке отображены методы создания запросов:

1. С помощью Мастера запросов, который упрощает создание простого, перекрестного запроса, а также запросов на поиск повторяющихся записей.

При этом:

- Простой запрос — Создает запрос к выбранным полям.
- Перекрестный запрос — суммирует данные в табличной форме.
- Повторяющиеся записи — обнаруживает повторяющиеся записи в таблице.

Записи без подчиненных — обнаруживает в таблице записи, у которых нет соответствующей записи в связанной таблице.

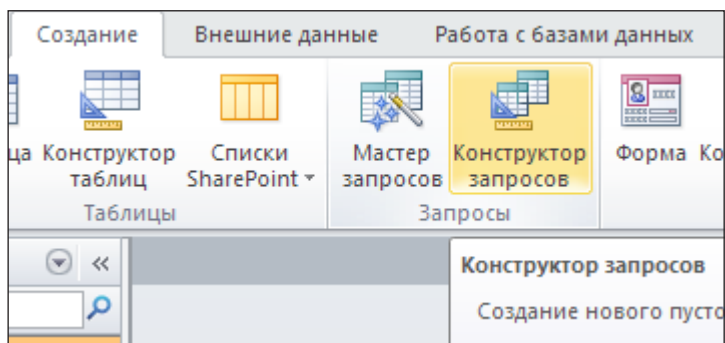


Рис. 1.18

2. Создания запроса с помощью Конструктора (рис. 1.18).
3. Это наиболее часто используемый способ создания запросов.

Конструктор открывает чистый бланк запроса.

Окно Конструктора состоит из двух частей. В верхней части располагаются объекты, данные из которых требуются в запросе (рис. 1.19). Это могут быть таблицы, запросы с названиями полей и связями между ними.

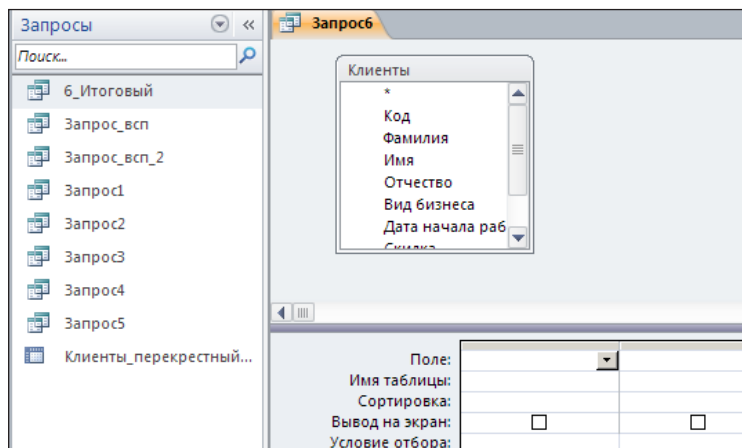


Рис. 1.19

Нижняя часть бланка предназначена для ввода параметров запроса:

- 1) первая строка бланка содержит имена полей, включенных в запрос;
- 2) вторая строка — объект, к которому принадлежит поле;
- 3) третья — групповая операция (если эта операция присутствует), позволяющая выполнять вычисления над значениями полей;
- 4) четвертая указывает принцип сортировки записей;
- 5) пятая — вывод на экран;
- 6) шестая — условия отбора записей;
- 7) седьмая — задает альтернативные условия.

Выражения, используемые в условиях отбора, вводятся вручную или создаются с помощью Построителя выражений (рис. 1.20).

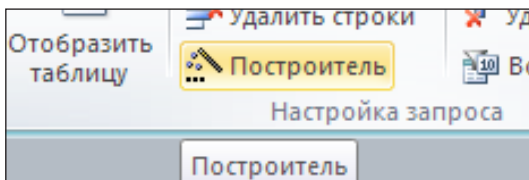


Рис. 1.20

Выражение — это аналог формул.

Выражения могут содержать:

- константы (1,5; 0,012; «АВ»);
- операторы (>; <; =; */; /; +; <>; between;&);
- идентификаторы (названия объектов или полей);
- встроенные функции (sum (); avg (); date ()).

Результатом вычисления любого выражения является некоторое значение.

Для открытия окна Построителя выражений нужно щелкнуть в поле условий отбора правой клавишей мыши или обратиться на ленту к инструменту Построитель.

Построитель предоставляет иерархические списки, содержащие большинство элементов (функций, операторов, полей), которые можно включить в выражение (рис. 1.21).

Выбрав нужный элемент, можно скопировать его в поле выражения, а затем щелкнуть по кнопке ОК, чтобы поместить его в бланк.

Можно выбирать нужные поля объектов БД (таблиц, запросов, форм), вставлять операторы и константы в выражения для вычислений и критериев.

В примере с помощью Построителя выражений получена формула с использованием выделенного поля запроса (рис. 1.22).

После ввода условий отбора запрос необходимо сохранить.

Виды запросов

Запрос на выборку

Цель запроса на выборку состоит в создании результирующей таблицы, в которой отображаются только нужные по условию запроса данные из связанных таблиц. В этом запросе можно сгруппировать записи для вычисления сумм и средних значений (рис. 1.23).

При помощи строки «Условие отбора», можно отобрать записи, используя следующие типы условий.

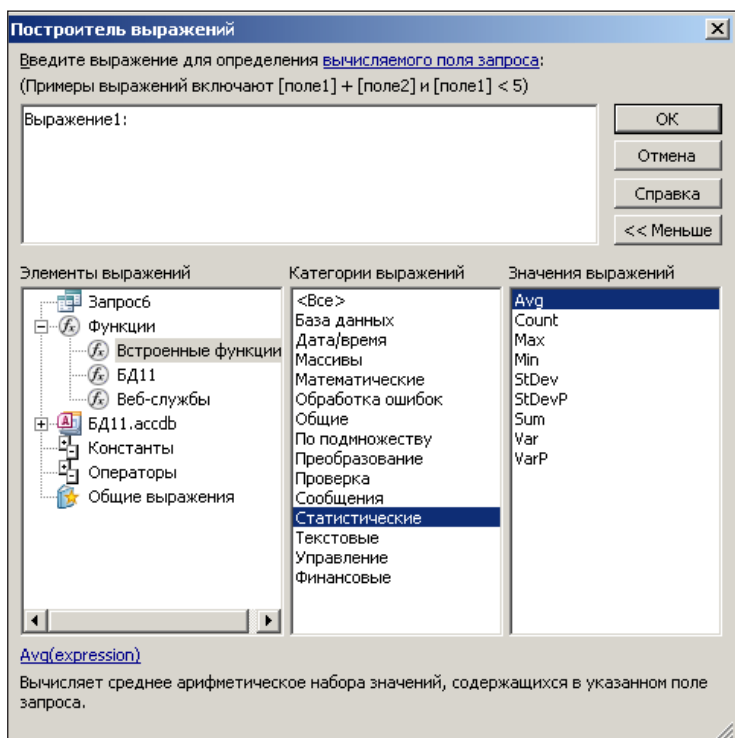


Рис. 1.21

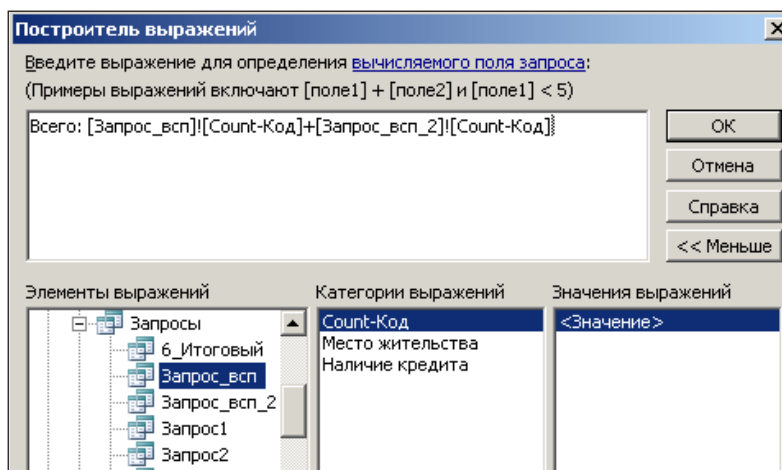


Рис. 1.22

Поле:	Фамилия	Дата начала работы	Место жительства	
Имя таблицы:	Клиенты	Клиенты	Клиенты	
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора: или:		> #01.03.2013#	< > "Сухой Лог"	

Рис. 1.23

1. Точное соответствие: вводится значение (текстовая строка или количество денег), которому должно соответствовать значение в поле.
2. Частичное соответствие. Используется комбинация цифр, букв и значков замены, которой должны соответствовать значения поля.

Таблица 3

*	Вместо любого количества любых символов
?	Вместо одного любого символа
#	Вместо одной цифры
[]	Для нахождения символов, совпадающих с одним из тех, что в скобках. Пример: A [BC], будет соответствовать AB или AC
!	Для нахождения символов, не совпадающих с теми, что в скобках
—	Для определения диапазона символов [A—K]
LIKE	Для поиска символов. LIKE « [A—Д]* » — поиск фамилий, начинающихся с букв А—Д

3. Критерии исключения. Используется оператор NOT для исключения записей, соответствующих этому критерию. Например: NOT AB.
4. Соответствие по дате. Используется точная дата или оператор DATE (), который представляет дату по часам ПК.
5. Пустые значения. Используется константа NULL для указания, что следует видеть только пустые ячейки. NOT NULL — только не пустые ячейки.
6. Операторы сравнения (>, <, =, <=, <>, =).
7. Множественные критерии. Для составления множественных критериев в внутри одного поля используются логические операторы (NOT, AND, OR, BETWEEN).

Запрос с параметром

Каждый раз при выполнении такого запроса появляется окно, в которое пользователь с клавиатуры может вводить различные вариан-

ты критериев поиска. Например, параметр — год рождения студентов группы. При первом обращении к запросу ищем студентов 1990 года рождения. При втором обращении вводим другое значение параметра — 1992. Получаем список студентов с 1992 годом рождения и т. д.

Параметров в запросе может быть несколько. Например, если нужно получать отчет, который определяется периодом. Логично в запросе задать два параметра: первый параметр — начало периода и второй параметр — конец периода.

Порядок действий при создании запроса с параметром:

- оторвать новое окно для построения запроса в режиме Конструктора;
- для каждого поля, которое предполагается использовать как параметр, ввести в ячейку строки «Условия отбора» текст приглашения, заключенный в квадратные скобки. Это приглашение будет выводиться при запуске запроса. Текст приглашения должен отличаться от имени поля, но может включать его.

Примеры:

- а) параметром в запросе является фамилия клиента, которая вводится с клавиатуры (рис. 1.24);

Поле:	Фамилия	Выражение1:	[Наз
Имя таблицы:	Клиенты		
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Условие отбора:	[Введите фамилию]		
или:			
		<input type="button" value="←"/> <input type="button" value="⌂"/>	

Рис. 1.24

- б) для ввода параметров временного интервала удобно использовать оператор BETWEEN «Выражение» and «Выражение». В качестве первого параметра «Введите начальную дату» и соответственно второго — «Введите конечную дату». Приглашения будут иметь вид BETWEEN [Введите начальную дату] and [Введите конечную дату];
- в) чтобы запросить у пользователя первый символ для поиска записей, которые начинаются с него:
LIKE [Введите первый символ для поиска] & «*».

Перекрестные запросы

В результирующей таблице отображаются данные, полученные на основе статистических расчетов, выполненных на основе одной или нескольких таблиц. Данные результирующей таблицы группируются по двум наборам: первый расположен в левом столбце таблицы, а второй — в верхней строке. Ячейки на пересечении строк и столбцов содержат вычисляемые значения (суммы, средние значения, количество).

Запрос на действие (активный запрос)

Вносит множественные изменения за одну операцию. Фактически это запрос на выборку, который выполняет определенные действия над результатом отбора (обновление, удаление, добавление записей и создание таблицы).

Для создания активного запроса необходимо создать запрос на выборку (рис. 1.25) в режиме Конструктора, а затем превратить его в активный запрос, выбрав тип запроса.

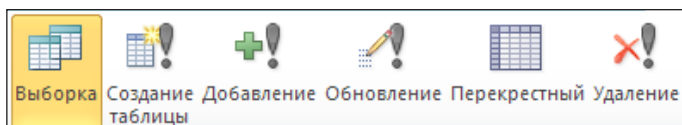


Рис. 1.25

Отмена внесенных изменений после запуска активного запроса будет невозможна, поэтому всегда при работе с активными запросами необходимо сохранять резервную копию объектов БД, которые находятся под воздействием активного запроса. Существует четыре формы запросов на действие:

1. Активный запрос на создание таблицы
 - Создаёт новую таблицу на основе всех или части данных из одной или нескольких таблиц.
 - Создаёт резервную копию таблицы.
 - Создаёт архивную копию, содержащую старые записи.
2. Активный запрос на добавление
 - Добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких других таблиц.
3. Активный запрос на обновление записей
 - Вносит общие изменения в группу записей одной или нескольких таблиц.

4. Активный запрос на удаление

Удаляет группу записей из одной или нескольких таблиц.

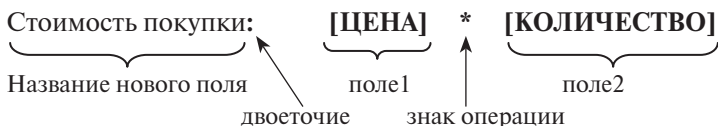
Вычисления в запросах

Поле, содержимое которого является результатом расчета по значению других полей, называется *вычисляемым полем*.

Вычисляемое поле существует только в результирующей таблице. В исходных таблицах такое поле не создается. С помощью запросов можно манипулировать данными и получать любые результаты, но при этом исходные таблицы остаются неизменными.

Для создания запроса, производящего вычисления, в столбце записывают имя нового поля и формулу. Имя нового вычисляемого поля отделяется от формулы двоеточием. В формулу могут входить числа, заключенные в квадратные скобки названия полей, участвующих в расчете, а также знаки математических операций и функции.

Например:



При записи длинной формулы при нажатой комбинации SHIFT+F2 открывается вспомогательное окно диалога, которое называется Область ввода. В нем можно ввести формулу произвольной длины — ОК.

Выполнение групповых расчетов

Microsoft Access позволяет выполнять вычисления для группы записей. Например, можно определить среднюю стоимость товара данного типа или найти работника с наибольшей заработной платой на предприятии. Для этого используются статистические функции в строке «Групповая операция» бланка запроса.

В бланке запроса раскрывающегося списка в ячейке строки «Групповая операция» двенадцать элементов:

1. Группировка — определяет группы, для которых выполняются вычисления.

2. Выражение — создает вычисляемое поле с использованием статистических функций.

Выражения, определяемые пользователем, дают возможность выполнять действия с числами, датами и текстовыми значениями в каждой записи с использованием данных из одного или нескольких полей.

При выполнении расчетов допускаются сложные выражения, например, Sum ([НаСкладе]+ [Заказано]).

Результаты вычислений не обязательно должны отображаться в поле. Вместо этого их можно использовать в условиях отбора для определения записей, которые выбираются в запросе, или для определения записей, над которыми производятся какие-либо действия.

3. Условие — Определяет условия отбора для поля, которое не участвует в группировке.
4. Девять статистических функций:

Таблица 4

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

При вычислении статистических функций не учитываются записи, содержащие пустые (Null) значения. Например, функция Count возвращает количество всех непустых полей.

Рекомендации

- Если вычисляемое поле определяется с помощью статистической функции, то Microsoft Access создает имя, под которым поле будет выводиться в режиме таблицы, путем объединения имени функции и имени поля, содержащего данные (например, «Avg_СтоимостьДоставки»).
- Если вычисляемое поле определяется с помощью выражения, которое включает одну или несколько статистических функций, необходимо выбрать в ячейке строки Групповая операция элемент «Выражение».
- Для каждого вычисляемого поля следует выбрать одно выражение.
- Выражения, определяющие вычисляемые поля, вводятся пользователем в строку «Групповая операция» бланка запроса, в которой допускается выбор статистических функций для выполняемых над полем вычислений.

Формы

MS Access позволяет с помощью форм создавать пользовательский интерфейс для работы с таблицами. Форму можно рассматривать как электронный документ, содержащий поля для ввода данных. Расположение полей может задаваться пользователем.

Проектирование формы состоит в размещении элементов управления на бланке формы и в задании связей между этими элементами и полями таблиц или запросов базы данных.

Структура форм состоит из разделов и элементов управления.

Форма имеет три раздела:

- раздел заголовка формы;
- область данных;
- раздел примечания формы.

В области данных содержатся элементы управления:

- связанное поле — то, что в него вводится, поступает и в одноименное поле таблицы, на базе которой создана форма;
- присоединенная надпись — называется так, поскольку перемещается вместе со своим элементом управления.

Фоновый рисунок, лежащий под элементами управления, показывает размер рабочего поля формы.

Редактирование форм состоит в создании новых или изменении имеющихся элементов управления, а также в изменении их взаимного расположения.

Для форматирования элемента управления его надо сначала выделить, щелкнув по нему. Вокруг элемента по периметру появляются маркеры. Теперь можно изменять параметры шрифта, выравнивание текста и другие элементы форматирования. Рамку можно растягивать или сжимать методом перетаскивания границ. При наведении на маркер указатель мыши меняет форму, принимая изображение открытой ладони. В этот момент рамку можно перемещать.

Можно также использовать дополнительные возможности изменения оформления, которые располагаются на панели форматирования: толщина линии/границы; оформление с тенью.

Элементы управления, в которых отображается содержимое полей таблицы, называются связанными полями.

Для их создания служит элемент «Поле» на панели элементов. При создании связанного поля вместе с ним одновременно образуется еще один элемент управления — присоединенная надпись. Оторвать поле от присоединенной надписи позволяет маркер, расположенный в левом верхнем углу выделенного поля. При наведении на него указатель мыши принимает форму указательного пальца. В этот момент связанное поле можно оторвать от присоединенной надписи и перемещать отдельно.

Перемещать элементы управления и изменять их размеры с помощью мыши не слишком удобно. Удобнее использовать курсорные клавиши и:

- shift — для изменения размеров элемента управления;
- ctrl — для изменения его расположения.

Форма создается на основе таблицы или запроса.

Формы могут быть выведены на экран в трех видах: режим формы, режим макета и режим конструктора (рис. 1.26).

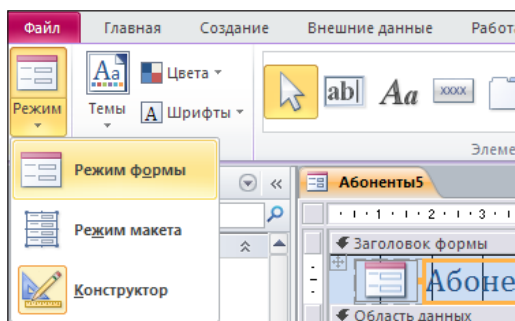


Рис. 1.26

Создание форм можно выполнять автоматически (автоформы), полуавтоматически (с помощью Мастера) или вручную (в режиме Конструктора).

Автоформы — самый простой вид форм. Для создания автоформы надо на панели навигации выделить объект, для которого создается форма, открыть вкладку Создать, **Формы**.

При этом получаем электронный бланк, содержащий все поля объекта, содержание которых можно проверять и редактировать (рис. 1.27), и также вводить новые. В нижней части бланка счетчик, с помощью которого можно найти требуемую запись.

Код клиента	1	Вид бизнеса	крупный
Фамилия	Алешин	Дата начала работы	01.03.2013
Имя	Иван	Скидка	0,1
Отчество	Петрович	Место жительства	Екатеринбург
		Наличие кредита	<input checked="" type="checkbox"/>

Код_заказа	дата заказа	товар	количество
*			

Рис. 1.27

Форма с помощью Мастера создается в четыре шага:

- 1) выделение объекта, для которого создается форма, выбор полей;
- 2) выбор внешнего вида формы;
- 3) выбор элементов форматирования формы;
- 4) ввод имени формы.

Готовую форму можно также использовать для просмотра существующих записей или для ввода новых.

В этом случае получаем электронный бланк, содержащий не все поля объекта, а только те, которые необходимы пользователю (рис. 1.28). Их также можно редактировать и также вводить новые данные.

Клиенты2

Код клиента	<input type="text"/>
Фамилия	Алешин
Имя	Иван
Отчество	Петрович
Место жительства	Екатеринбург

Записи: 1 из 9 Нет фильтра Поиск

Рис. 1.28

Главная и подчиненная формы

Существуют формы, в которых можно одновременно отобразить информацию из двух или более связанных таблиц. В этом случае внешняя форма называется главной, вложенные — подчиненными. В результате в окне такой формы выводятся только те записи подчиненной таблицы, которые связаны с текущей записью исходной (главной) таблицы.

Комбинацию «главная форма/подчиненная форма» часто называют также сложноподчиненной формой или комбинацией «родительской» и «дочерней» форм.

При создании главной формы и подчиненной формы, основанных на таблицах, связанных отношением «один-ко-многим», главная форма представляет сторону отношения «один», а подчиненная форма представляет сторону «многие». Подчиненные формы не отображаются, когда главная форма находится в режиме сводной таблицы или сводной диаграммы.

Главная форма может иметь любое число подчиненных форм, если каждая подчиненная форма помещается в главную форму. Имеется также возможность создавать подчиненные формы до семи уровней вложенности. Это означает, что можно иметь подчиненную форму внутри главной формы, а другую подчиненную форму внутри этой

подчиненной формы и т. д. Например, можно иметь главную форму, отображающую сведения о клиентах, подчиненную форму, отображающую сведения о заказах, и еще одну подчиненную форму, отображающую сведения о заказанных товарах. Однако в форме в режиме сводной таблицы или сводной диаграммы подчиненные формы не отображаются.

При использовании формы с подчиненной формой для ввода новых записей текущая запись в главной форме сохраняется при вводе данных в подчиненную форму. Это гарантирует, что записи из таблицы на стороне «многие» будут иметь связанную запись в таблице на стороне «один». Это также автоматически сохраняет каждую запись, добавляемую в подчиненную форму.

Создание главной и подчиненной форм:

- 1) на ленте выбрать вкладку Создание;
- 2) в Области навигации выделить связанные объекты, данные из которых размещаются в формах;
- 3) в группе Формы выбрать команду Мастер форм, открывается окно Создание форм (рис. 1.29);

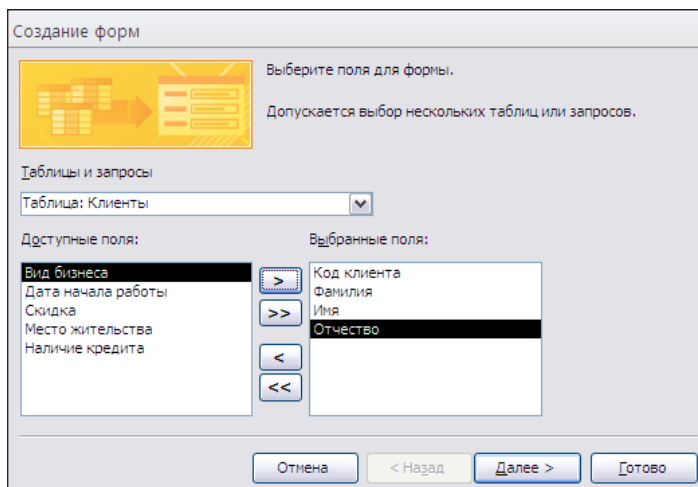


Рис. 1.29

- 4) в окне из раскрывающегося списка Таблицы и Запросы выбрать первый объект для главной формы;
- 5) в окне Доступные поля выбрать имена полей для главной формы;

- 6) вернуться в окно с раскрывающимся списком Таблицы и Запросы, выбрать следующий объект для подчиненной формы;
 - 7) вернуться в окно Доступные поля, выбрать имена полей для подчиненной формы;
- Далее; переключатель Подчиненные формы; Готово (см. рис. 1.).

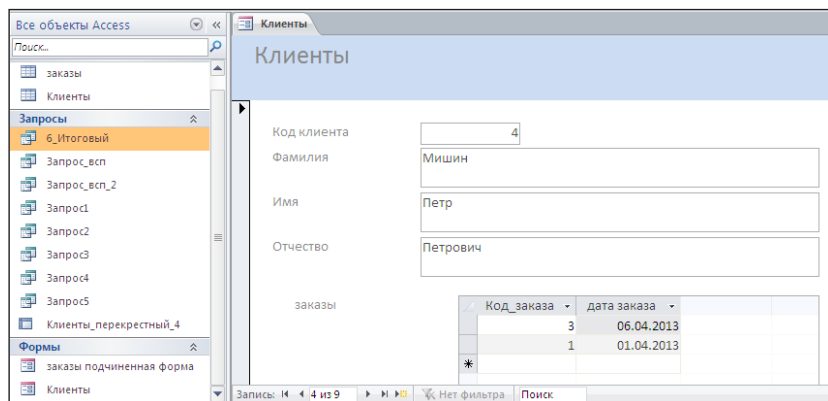


Рис. 1.30

Главная кнопочная форма

С помощью главной кнопочной формы можно осуществлять навигацию по формам и отчетам базы данных. Пользователю базы данных нет необходимости просматривать всю структуру базы. Пользователь с помощью форм может внести изменения в таблицы и затем получить отчеты с учётом новых данных. Запросы и таблицы не являются элементами главной кнопочной формы.

Для создания главной кнопочной формы нужно воспользоваться инструментом Диспетчер кнопочных форм. Если на вкладке Работа с базами данных этого инструмента нет, то его можно добавить на Панель быстрого доступа.

Для этого на вкладке Файл, выбрать Параметры, затем Панель быстрого доступа. После этого на вкладке Настройка панели быстрого доступа надо выбрать из раскрывающегося списка строку Вкладка «Работа с базами данных» и в списке команд выделить Диспетчер кнопочных форм (рис. 1.31).

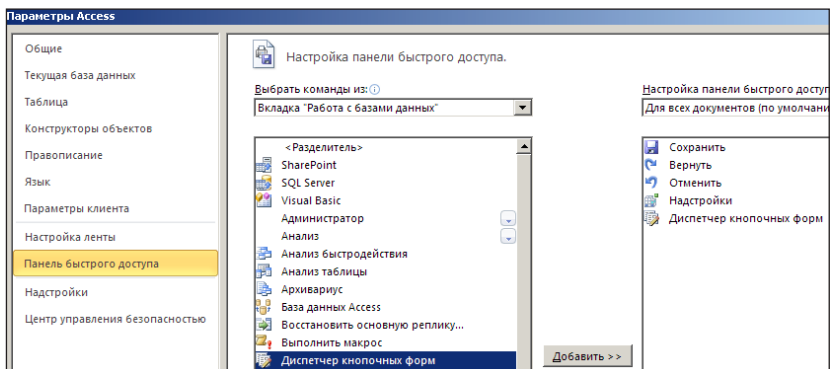


Рис. 1.31

Создание главной кнопочной формы:

- 1) открыть базу данных;
- 2) обратиться к «Диспетчеру кнопочных форм» (рис. 1.32);
- 3) если выводится запрос на подтверждение создания кнопочной формы, нажмите кнопку «Да».

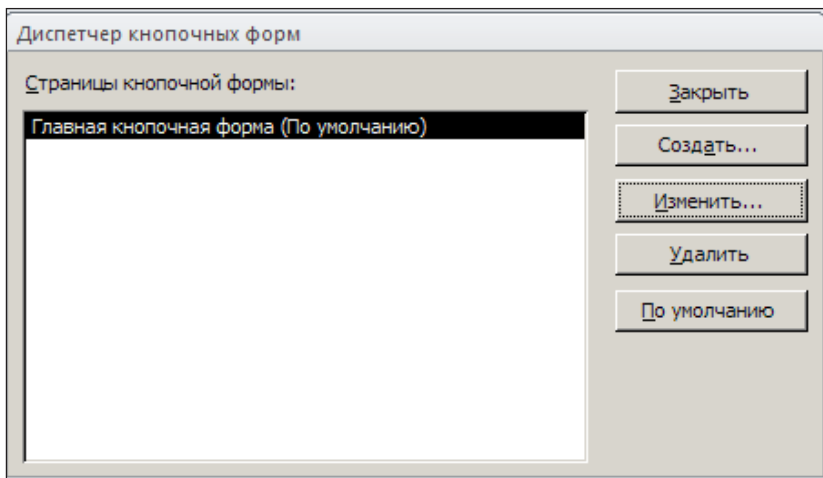


Рис. 1.32

- 4) дальнейшая работа будет производиться со стандартной кнопочной формой;

- 5) нажать кнопку «Изменить»;
- 6) нажать кнопку «Создать» (рис. 1.33).

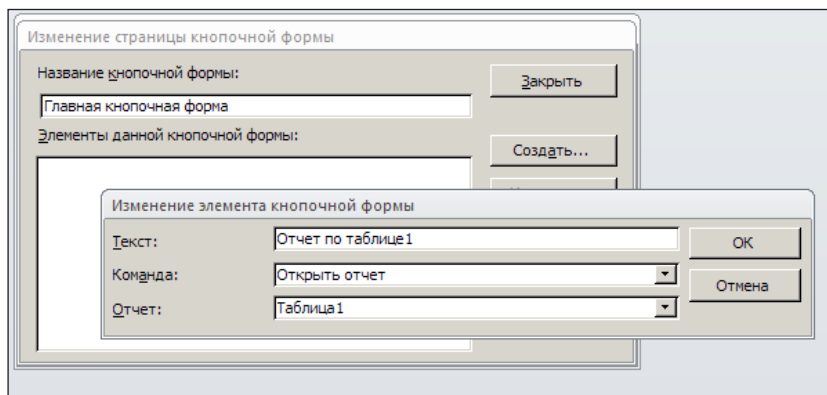


Рис. 1.33

- 7) в поле «Текст» введите текст названия первой кнопки кнопочной формы, а затем выберите для нее команду в поле Команда;
- 8) для создания второй кнопки используйте команду «Создать», после чего открывается окно «Изменение элемента кнопочной формы» для новой кнопки (рис. 1.34).

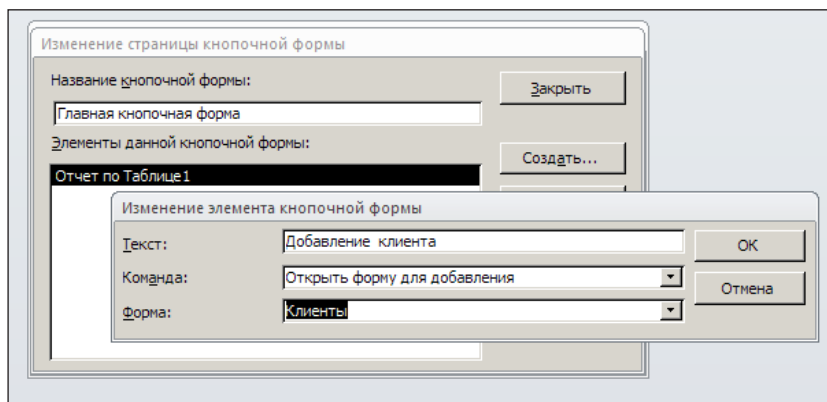


Рис. 1.34

Получаем главную кнопочную форму (рис. 1.35).

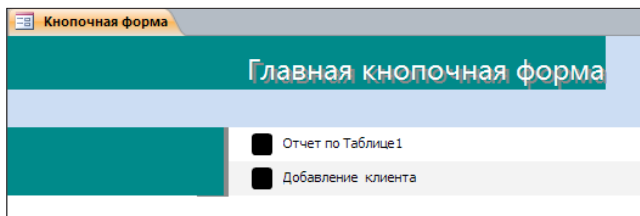


Рис. 1.35

Чтобы изменить или удалить какую-либо из созданных кнопок, выберите ее имя в списке «Элементы данной кнопочной формы» и нажмите кнопку «Изменить» или «Удалить». Если требуется изменить порядок элементов кнопочной формы, выберите элемент в списке и воспользуйтесь кнопками «Вверх» или «Вниз».

Закончив создание кнопочной формы, нажмите кнопку «Заккрыть».

Чтобы кнопочная форма открывалась при открытии базы данных, выберите имя кнопочной формы в диалоговом окне «Диспетчер кнопочных форм» и нажмите кнопку «По умолчанию».

При создании кнопочной формы с помощью диспетчера кнопочных форм создается таблица «Switchboard Items». Эта таблица описывает текст и действия кнопок созданной формы. Если позднее появится необходимость внести изменение в кнопочную форму, то сначала необходимо удалить соответствующую ей таблицу «Switchboard Items», а затем внести изменения.

Отчеты

Отчеты служат для форматированного вывода данных на печатающие устройства. При создании отчета MS Access всегда оперирует только с одной единственной таблицей или запросом. Если необходимо объединить информацию из нескольких таблиц и (или) запросов в одном отчете, то прежде следует собрать желаемые данные в одном запросе.

Структура отчетов состоит из разделов и элементов управления. Проектирование отчета состоит в создании структуры его разделов и в размещении элементов управления внутри этих разделов, а также в задании связей между этими элементами и полями таблиц или запросов базы данных.

Структура готового отчета отличается от структуры формы увеличенным количеством разделов. Кроме разделов заголовка, примечания и данных, отчет может содержать разделы верхнего и нижнего колонтитулов. Если отчет занимает более одной страницы, эти разделы необходимы для печати служебной информации, например номеров страниц. Чем больше страниц занимает отчет, тем важнее роль данных, выводимых на печать через эти разделы. Если для каких-то полей отчета применена группировка, количество разделов отчета увеличивается, поскольку оформление заголовков групп выполняется в отдельных разделах.

Редактирование структуры отчета выполняют в режиме Конструктора. Приемы редактирования те же, что и для форм. Элементы управления в данном случае выполняют функции элементов оформления, поскольку печатный отчет не интерактивный объект, в отличие от электронных форм.

Со структурой отчета проще всего ознакомиться, создав автоотчет, а затем открыв его в режиме Конструктора.

1. Структура отчета состоит из пяти разделов: заголовка отчета, верхнего колонтитула, области данных, нижнего колонтитула и примечания.
2. Раздел заголовка служит для печати общего заголовка отчета.
3. Раздел верхнего колонтитула можно использовать для печати подзаголовков, если отчет имеет сложную структуру и занимает много страниц. Здесь можно также помещать и колонцифры (номера страниц), если это не сделано в нижнем колонтитуле.
4. В области данных размещают элементы управления, связанные с содержимым полей таблиц базы. В эти элементы управления выдаются данные из таблиц базы, которые могут быть использованы для вывода на бумагу.
5. Раздел нижнего колонтитула используют для тех же целей, что и раздел верхнего колонтитула.
6. Раздел примечания используют для размещения дополнительной информации.

Автоотчет — самый простой вид отчета (рис. 1.36). Для создания автоотчета надо на панели Навигации выделить объект, для которого создается отчет, открыть вкладку Создать, Отчет.

Код клиента	1	Вид бизнеса	крупный
Фамилия	Алешин	Дата начала работы	01.03.2013
Имя	Иван	Скидка	0,1
Отчество	Петрович	Место жительства	Екатеринбург

Рис. 1.36

При этом получаем электронный вид отчета, содержащий все поля объекта.



С помощью Мастера отчетов.

В этом случае создается простой настраиваемый отчет в четыре шага:

- 1) выделение объекта;
- 2) в группе Отчеты выбрать Мастер отчетов;
- 3) выбрать поля для отчета;
- 4) выберите требуемый порядок сортировки полей;
- 5) выберите вид макета отчета.

При этом получаем электронный вид отчета, содержащий только требуемые поля объекта (рис. 1.37).

Код клиента	Фамилия	Имя	Отчество
1	Алешин	Иван	Петрович
2	Иванов	Артём	Васильевич
3	Илюшин	Олег	Михайлович
4	Мишин	Петр	Петрович
5	Васильева	Ольга	Федоровна
6	Макеева	Елена	Николаевна
7	Сидоров	Иван	Васильевич
8			

Рис. 1.37

Раздел 2.

Лабораторный практикум

Лабораторная работа 1

Задание 1

Сделайте для себя копию файла `Торговля.acsbd`, для того, чтобы вы могли экспериментировать с базой, не затрагивая первоисточник и не мешая своим товарищам. Откройте этот файл. В левой части экрана вы увидите перечень групп перечисленных выше основных объектов базы данных. Раскрывая каждый список, вы увидите перечень объектов соответствующей группы.

Поскольку главными объектами являются таблицы (без них база данных просто не существует), начнем знакомство именно с них.

Задание 2

Откройте таблицу `Товары` и рассмотрите ее. Таблица, как и можно было предположить по ее названию, содержит информацию о товарах, которыми торгует некая фирма. В таблице хранится 77 записей (это можно увидеть на панели внизу таблицы Запись: 1 из 77).

Каждая запись состоит из 10 полей, т.е. товар описывается десятью атрибутами. Их названия видны в верхней строке таблицы.

В большой таблице бывает необходимо быстро отыскать нужную запись. Для этого служит кнопка `Найти` на главной вкладке меню. Найдите в таблице `Товары` марку `Ravioli Angelo` и определите код этого товара.

Задание 3


Изначально строки таблицы упорядочены по возрастанию ключевого поля (кода товара). Для удобства поиска информации существует возможность сортировки таблицы по любому из атрибутов. Для этого рядом с названием атрибута имеется управляющая кнопка. Нажав ее, вы можете выбрать способ сортировки. Обратите внимание, что способы сортировки зависят от типа данных (от `A` до `Я` для текстовых полей, от максимального к минимальному — для числовых, от установленных к снятым — для логических).

Отсортируйте таблицу по типам товаров. Обратите внимание, что строки таблицы перемещаются целиком, не разрушаясь, т. е. при сортировке не нарушается целостность данных.

Отмените сортировку — товары вновь будут расположены по возрастанию кодов. Отсортируйте таблицу по убыванию цены.

Еще раз отмените сортировку и отсортируйте таблицу по логическому полю Поставки прекращены.

Задание 4


Для просмотра некоторой части таблицы в Access существует возможность фильтрации данных. Она выполняется с помощью той же кнопки, что и сортировка. Отфильтруйте данные, чтобы увидеть только кондитерские изделия. Значок  рядом с названием атрибута говорит о том, что на него установлен фильтр.

Отмените фильтр и установите новый так, чтобы видеть одновременно мясные (мясо/птица) и рыбные продукты.

Возможна фильтрация одновременно по нескольким атрибутам. Отберите в таблице приправы, на которые прекращены поставки (у вас должно остаться две строки).

Отметим, что фильтры и сортировки не сохраняются при закрытии файла, поэтому если выборку требуется выполнять часто, следует сохранять ее в виде запроса.

Задание 5

Добавление данных можно производить непосредственно в таблицу. Добавьте в конец таблицы еще одну произвольную запись. Для того, чтобы быстро, не прокручивая все строки, попасть на первую свободную запись, нажмите внизу таблицы кнопку : новая (пустая) запись.

При вводе вы заметите разницу между типами данных различных атрибутов. Так, атрибут КодТовара не вводится вручную, а формируется автоматически (тип данных Счетчик).

Атрибуты Марка и ЕдиницаИзмерения вводятся произвольно (тип данных Текстовый).


Атрибуты Тип и Поставщик выбираются из предложенного списка (использован Мастер подстановок).

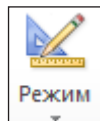
Атрибут Цена вводится произвольным дробным числом без наименования, никакие символы, кроме цифр и запятой не допускаются. Наименование «р» и пробелы добавляются автоматически (тип данных Денежный).

Атрибуты НаСкладе, Ожидается и МинимальныйЗапас вводятся произвольным целым числом, никакие символы, кроме цифр, не допускаются (тип данных Числовой целый).

И наконец, атрибут ПоставкиПрекращены вводится с помощью флажка (тип данных Логический).

Задание 6

Для того чтобы увидеть или указать тип данных для атрибута, нужно перейти в режим конструктора. Для этого служит кнопка на панели инструментов (в левом верхнем углу). Нажмите ее. Вы перестанете видеть содержимое таблицы, зато увидите структуру (рис. 2.1). При этом вид кнопки изменился, она стала выглядеть так:  и ее нажатие переводит нас снова в режим просмотра таблицы.



В первом столбце теперь перечислены имена полей, во втором указаны типы данных (они выбираются из предлагаемого перечня). В третьем могут содержаться пояснения к атрибутам. Ключик рядом с именем поля указывает, что это поле — ключевое.

Товары		
Имя поля	Тип данных	Описание
КодТовара	Счетчик	Уникальное число, автоматически присваиваемое новой записи.
Марка	Текстовый	
Тип	Текстовый	
КодПоставщика	Числовой	Совпадает со значением поля "КодПоставщика" в таблице "Поставщики".
ЕдиницаИзмерения	Текстовый	(например, в ящиках по 24 бутылки, 1-литровая бутылка и т.п.).
Цена	Денежный	
НаСкладе	Числовой	
Ожидается	Числовой	
МинимальныйЗапас	Числовой	Минимально допустимый складской запас.
ПоставкиПрекращены	Логический	"Да" означает, что товар больше не поставляется.

Свойства поля	
Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Список значений
Источник строк	"Напитки"; "Приправы"; "Кондитерские изделия"; "Молочные продукты"; "Хлебобулочные изделия"; "Мясо/птица"
Присоединенный столбец	1
Число столбцов	1
Заглавия столбцов	Нет
Ширина столбцов	3,519см
Число строк списка	16
Ширина списка	3,519см
Ограничиться списком	Нет
Разрешение нескольких зл	Нет
Разрешить изменение слв	Да
Формат изменения элемет	
Только значения источни	Нет

Рис. 2.1

Выберите поочередно элемент Тип данных для каждого атрибута и рассмотрите в нижней части окна общие свойства для каждого из них. Затем перейдите на вкладку Подстановка и снова просмотрите все атри-

буты. В большинстве из них на этой вкладке ничего интересного не обнаружится. Однако для атрибута Тип вы увидите список всех возможных значений поля. Именно это позволяло нам ускорить ввод данных. Такую возможность обеспечивает нам Мастер подстановок. Подробнее вы познакомитесь с ним, когда будете создавать свою базу данных.

Вспомним, что для ввода поля КодПоставщика мы тоже выбирали значение из списка. Однако на вкладке Подстановка мы видим несколько другую картину (рис. 2.2). В качестве типа источника строк указан не список, а Таблица или запрос, а источник строк отсылает нас к другой таблице. Мастер подстановок может таким образом связывать таблицы между собой. Подробно об этом речь пойдет в последующих лабораторных работах.

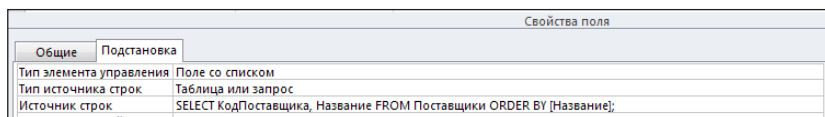


Рис. 2.2

Задание 7

Теперь пришло время взглянуть на базу данных в целом. Понятно, что торговля не ограничивается одними товарами. Должны быть и другие объекты — поставщики, клиенты, заказы и др. А значит, база должна состоять из нескольких взаимосвязанных таблиц. Увидеть ее структуру можно, перейдя в главном меню на вкладку Работа с базами данных и нажав кнопку Схема данных (рис. 2.3).

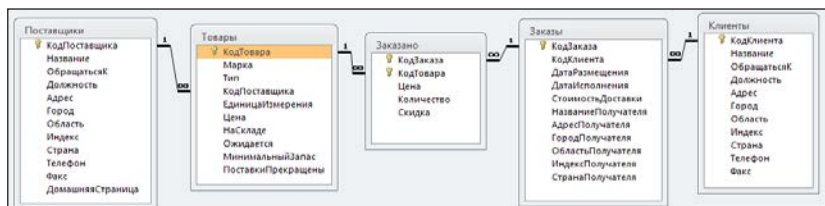


Рис. 2.3

Из этой схемы мы видим, какие информационные объекты фигурируют в деятельности торговой фирмы, какими атрибутами они описываются и как связаны между собой. Кроме того, на схеме видно, какое поле (может быть более одного) является ключевым в каждой таблице

Лабораторная работа 2

Создадим базу, содержащую одну таблицу с данными клиентов фирмы. Такие базы могут быть полезны при получении статистических данных о деятельности фирмы.

Задание 1

Подготовим все необходимое для создания простой базы данных.

Создание базы данных начинается с обращения к представлению Backstage (вкладка Файл), Создать, Новая база данных (рис. 2.4).

В правой части окна, в разделе Новая база данных, в поле Имя файла вводим имя файла базы данных «БД1» и определяем место хранения этого файла (рис. 2.5).

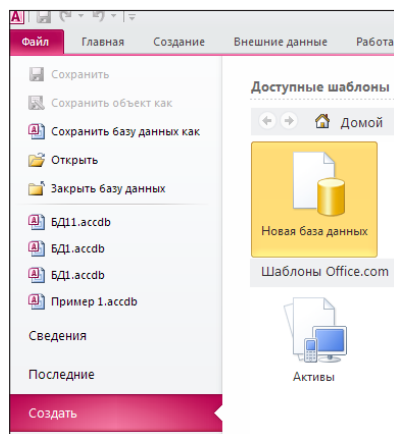


Рис. 2.4

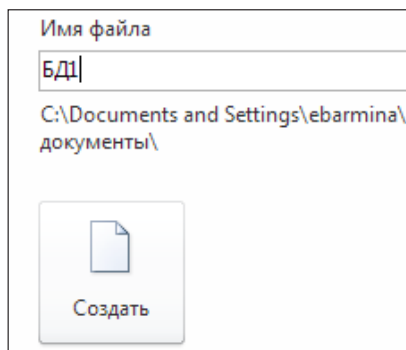


Рис. 2.5

Открывается таблица в режиме отображения содержания (рис. 2.6).

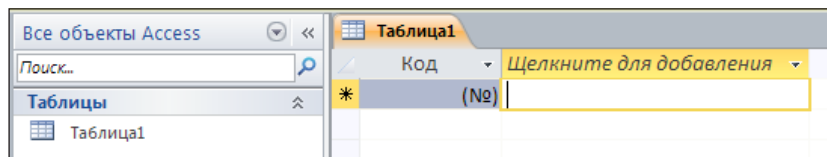



Рис. 2.6

Задание 2

Создаем структуру таблицы Клиенты.

Нажатием на кнопку  переходим в режим Конструктора, при этом Access предлагает ввести имя таблицы (рис. 2.7).

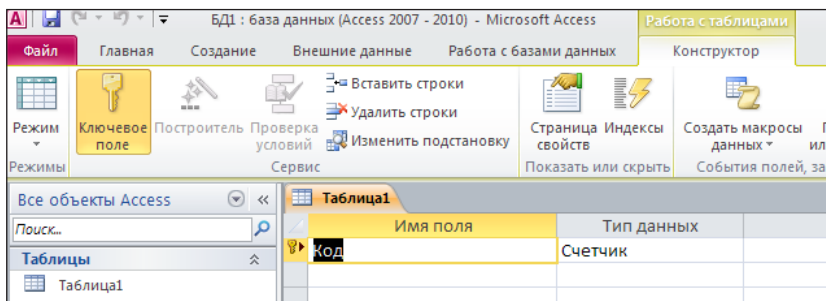


Рис. 2.7

В режиме Конструктора, в окне создания таблицы первое поле автоматически назначено ключевым и получило название «Код». Для нашей таблицы удобно назвать его «Код клиента» — изменим его. Это поле однозначно определяет клиента. Повторяться значения в нем не могут, и оно не может быть равно нулю. В этом случае тип поля лучше взять «Счетчик». При этом заполняться поле будет автоматически.

Второе, третье и четвертое поля — «Фамилия», «Имя», «Отчество» — текстовые. Для ускорения ввода и избежания ошибок ввода повторяющиеся данные лучше копировать.

Поле «Вид бизнеса» содержит ограниченный набор элементов. Для ускорения ввода и избежания ошибок ввода это поле следует заполнять с использованием Мастера подстановок.

В поле «Дата начала работы» в свойствах поля, в строке «Формат поля» выбрать «Краткий формат даты» (рис. 2.8). Вводить данные мы будем с помощью всплывающего «Календаря».

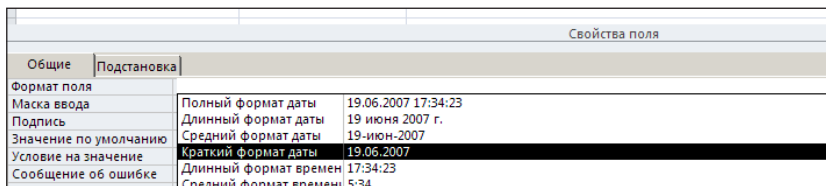


Рис. 2.8

По содержанию задачи поле «Скидка» может содержать значения больше 0 и меньше 1, поэтому в свойствах поля в строке «Размер поля» выбрать «Одинарное с плавающей точкой».

Поле «Место жительства» имеет текстовый тип данных и заполняется аналогично полям «Фамилия», «Имя», «Отчество».

В поле «Наличие кредита» проставляется факт использования кредита «Да» или «Нет». Поэтому удобно выбрать тип данных — «Логический».

В результате работы таблица в режиме Конструктора примет вид, показанный на рисунке 2.9.

Имя поля	Тип данных
Код	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Вид бизнеса	Текстовый
Дата начала работы	Дата/время
Скидка	Числовой
Место жительства	Текстовый
Наличие кредита	Логический

Рис. 2.9

Задание 3

Переходим в режим просмотра таблицы и вводим данные. Способ ввода зависит от типа вводимых данных. Часть данных вводится вручную, часть выбирается из списка, код формируется автоматически, наличие кредита устанавливается с помощью флажка. В итоге таблица должна иметь вид, показанный на рисунке 2.10.

Код	Фамилия	Имя	Отчество	Вид бизнеса	Дата начал	Скидка	Место жите	Наличие кредита
1	Алешин	Иван	Петрович	крупный	01.03.2013	0,1	Екатеринбург	<input checked="" type="checkbox"/>
2	Иванов	Артем	Васильевич	средний	05.03.2013	0,05	Реж	<input type="checkbox"/>
3	Илюхин	Олег	Михайлович	мелкий	28.02.2013	0,05	Екатеринбург	<input checked="" type="checkbox"/>
4	Мишин	Петр	Петрович	средний	26.01.2013	0	Сухой Лог	<input type="checkbox"/>
5	Васильева	Ольга	Федоровна	мелкий	21.03.2013	0,04	Сухой Лог	<input checked="" type="checkbox"/>
6	Макеева	Елена	Николаевна	крупный	19.01.2013	0,06	Екатеринбург	<input type="checkbox"/>
7	Сидоров	Иван	Васильевич	средний	12.03.2013	0,01	Сухой Лог	<input checked="" type="checkbox"/>
8	Воробьева	Ирина	Федоровна	мелкий	01.03.2013	0	Екатеринбург	<input checked="" type="checkbox"/>
9	Огнев	Вадим	Михайлович	средний	20.03.2013	0	Реж	<input checked="" type="checkbox"/>
(№)						0		<input type="checkbox"/>

Рис. 2.10

Лабораторная работа 3

Рассмотрим, как связаны между собой таблицы в базе данных «Торговля». В лабораторной работе 1 вы уже видели схему данных этой БД (см. рис. 2.3), сейчас разберем ее более детально.

Задание 1

Откройте свою копию БД «Торговля», перейдите в главном меню на вкладку Работа с базами данных и нажмите кнопку Схема данных. На схеме связи между таблицами изображены в виде линий, соединяющих одноименные поля таблиц. Линии снабжены маркерами «1» (один) и «∞» (много). Как видно на схеме, все связи в нашей БД относятся к типу «один-ко-многим».

Выделите мышью связь между таблицами Поставщики и Товары и удалите ее. Теперь установите связь заново путем перетаскивания мышью имени поля из одной таблицы в другую на соответствующее ему связанное поле. После перетаскивания открывается окно диалогов Изменение Связей, в котором можно задать свойства образующейся связи (рис. 2.11).

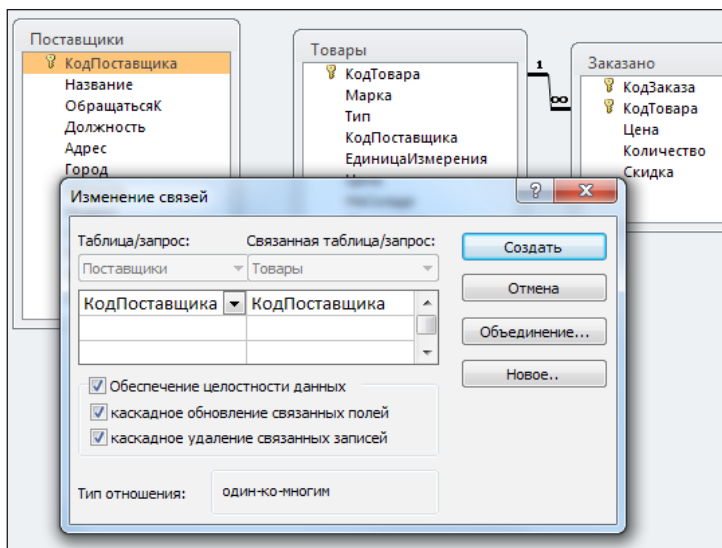


Рис. 2.11

Включение флажка **Обеспечение целостности данных** позволяет защититься от случаев удаления записей из одной таблицы, при которых связанные с ними данные других таблиц останутся без связи.

Флажки **Каскадное обновление связанных полей** и **Каскадное удаление связанных записей** обеспечивают одновременное обновление или удаление данных во всех подчиненных таблицах при их изменении в главной таблице.

Нажмите кнопку Создать, не включая флажки. Вы увидите, что между таблицами установилась связь, но тип ее не определен. Выделите связь и с помощью правой кнопки мыши выберите режим Изменить связь. На этот раз включите все флажки, и вы увидите, что теперь связь имеет тип «один-ко-многим».

Закройте базу данных Торговля.

Задание 2

Начиная с этого задания, мы будем разрабатывать базу данных «Телефонная компания» для обслуживания фирмы, предоставляющей услуги сотовой связи. Разумеется, мы смоделируем лишь небольшую часть функций, выполняемых компанией, в реальности подобная БД устроена гораздо сложнее.

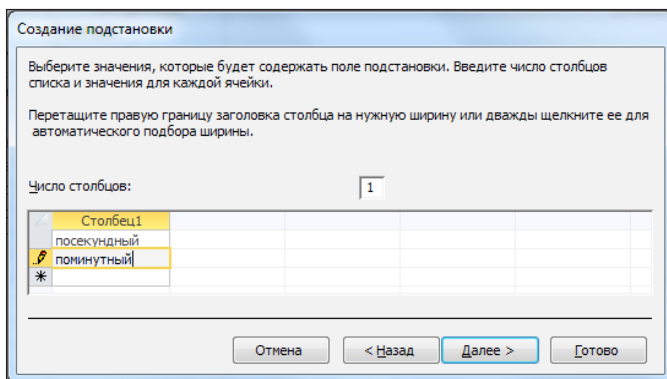
Сначала зададимся вопросом, какие классы объектов участвуют в бизнес-процессах, т. е. какие таблицы нам предстоит создать. Во-первых, у телефонной компании должны быть абоненты (без них она существовать не может). Во-вторых, абоненты будут совершать звонки (для этого они и пользуются услугами компании). В третьих, компания разработала для своих клиентов несколько тарифных планов. Ограничимся этими тремя видами объектов и будем создавать БД, состоящую из трех таблиц.

Выберите в MS Access в меню Файл режим Создать — Новая база данных, в правой части окна укажите имя файла, его местоположение и нажмите кнопку Создать.

Выберите режим конструктора для появившейся автоматически таблицы, начнем конструировать первую таблицу. Для того чтобы у нас появились абоненты, мы должны им что-то предложить. Поэтому начнем с таблицы тарифных планов.

Ключевым полем назначим порядковый номер **КодПлана** (тип данных — счетчик). Каждому плану дадим привлекательное название, т. е. следующим полем будет **Название** (тип данных — текстовый). Чтобы не слишком усложнять задачу, договоримся, что тарифные планы будут описываться двумя атрибутами — стоимостью секунды разгово-

ра внутри своей сети и вне ее (тип данных — денежный). И наконец, последний атрибут тарифного плана **Тип тарификации** — посекундный или поминутный. Чтобы не вводить длинные слова многократно и не допускать ошибок, при выборе типа данных воспользуемся мастером подстановок (рис. 2.12). Выберите вариант «Будет введен фиксированный набор значений», нажмите кнопку Далее. Введите в столбец нужные значения, как показано на рисунке 2.12, и нажмите кнопку Готово.



Тарифы	
Имя поля	Тип данных
КодПлана	Счетчик
Название	Текстовый
Стоим_сек_внутр	Денежный
Стоим_сек_внеш	Денежный
Тип_тарификации	Текстовый
	Текстовый
	Поле MEMO
	Числовой
	Дата/время
	Денежный
	Счетчик
	Логический
	Поле объекта OLE
	Гиперссылка
	Вложение
	Вычисляемый
	Мастер подстановок.

Рис. 2.12

Для всех полей таблицы задайте свойству Обязательное поле значение Да, поскольку никакие поля не должны оставаться пустыми.

Закроем таблицу, дав ей название, например Тарифы. Теперь откроем ее в режиме таблицы и убедимся, что она готова для ввода данных (столбец Тип_тарификации снабжен кнопкой для выбора одного из двух значений, атрибут Код формируется автоматически, остальные значения можно вводить произвольно). Заполнять таблицу пока не будем, лучше сначала установить все связи. Это поможет избежать ошибок при вводе, возникающих из-за несоответствия типов данных.

Задание 3

Следующим шагом будет создание таблицы Абоненты. Ключевым полем здесь удобно объявить **НомерТелефона** — повторения в этом поле невозможны. Хотя номер состоит из цифр, тип данных предпочтительнее текстовый. Вычисления с номерами нам не потребуются, а вот различать номера, начинающиеся с одинаковых цифр, будет необходимо. Номера телефонов имеют 10 цифр, поэтому в строке Размер поля укажем 10. Договоримся, что все наши абоненты будут иметь номер, начинающийся с 999, и для упрощения ввода установим в строке Значение по умолчанию «999» (вместе с кавычками).

Для абонентов мы будем хранить в таблице **фамилию** (тип данных текстовый). Некоторым абонентам будут предоставляться **льготы**: тип данных числовой с плавающей точкой, формат поля — процентный, значение по умолчанию — 0, условие на значение — ≥ 0 and < 1 . Еще один атрибут абонента — **тарифный план**. Он устанавливается с помощью мастера подстановок последовательным выбором «Объект ... получит значение из другой таблицы или запроса», далее выбрать таблицу **Тарифы**, выбрать из доступных полей **КодПлана** и завершить работу мастера.

Все поля таблицы сделайте обязательными. Закройте таблицу, сохранив ее с нужным именем.

Задание 4

Чтобы наша компания работала, абоненты должны звонить, а мы должны хранить информацию о звонках. Создайте таблицу Звонки следующей структуры.

Ключевое поле **КодЗвонка**, тип данных — счетчик.

НомерТелефона звонившего — через Мастер подстановок связывается с ключевым полем **НомерТелефона** таблицы Абоненты.

НомерВызываемого — тип данных текстовый, размер поля 10, данные вводятся произвольно, звонить можно абонентам как своей сети, так и любой другой.

Продолжительность разговора — тип данных числовой целый, измеряется в секундах.

Все поля таблицы сделайте обязательными и закройте таблицу, сохранив ее с нужным именем.

Задание 5

Установите связи между таблицами. Для этого, как вы уже знаете, нужно открыть окно Схема данных на вкладке Работа с базами данных. На схеме вы увидите, что связи между таблицами установились автоматически, потому что при создании таблиц был использован мастер подстановок. Однако тип связей не определен. Обе связи в нашей БД относятся к типу «один-ко-многим» (одним тарифным планом пользуется много абонентов, один абонент совершает много звонков). Выделите поочередно каждую связь мышью, нажмите правую кнопку мыши, выберите режим Изменить связь и установите флажки во всех пунктах окна Изменение связи (Обеспечение целостности данных, Каскадное обновление связанных полей и Каскадное удаление связанных записей). Затем нажмите кнопку Создать, и вы увидите, что установился нужный нам тип отношения, так как связывается ключевое поле с не ключевым.

Если ранее никаких связей между таблицами базы не было, то при открытии окна Схема данных одновременно открывается окно Добавление таблицы, в котором можно выбрать нужные таблицы для включения в структуру межтабличных связей.

Если некоторые связи между таблицами уже были заданы, то для введения в схему данных новой таблицы надо щелкнуть правой кнопкой мыши на схеме данных и выбрать Добавить таблицу. Введя в схему данных все необходимые таблицы, перечень таблиц нужно закрыть.

Закройте окно Схема данных, сохранив макет.

Задание 6

Заполните таблицы произвольными данными. Сначала введите три тарифных плана с различными характеристиками.

Затем в таблицу абонентов введите 7–8 записей так, чтобы были использованы все тарифные планы (выбираем их из раскрывающегося списка). Часть абонентов должна иметь льготу, остальные абоненты не имеют льгот, т. е. это поле остается без изменения.

И наконец, в таблицу звонков введите 15 записей. Обязательно предусмотрите, чтобы там присутствовали звонки как внутри сети (номер вызываемого начинается с 999), так и абонентам других операторов (номер вызываемого начинается с других цифр). Часть звонков должны быть короткими (1–2 секунды), остальные звонки большей длительности.

Лабораторная работа 4

Построение запросов для простейшей базы

Продолжаем работать с базой данных, которая была сохранена под именем БД1. В этой базе создана одна таблица Клиенты. Все последующие запросы строятся на основе этой таблицы.

Задание 1

Построим запрос на выборку. Предположим, нам требуется найти клиентов, начавших работать с фирмой после 1.03.2013 года и проживающих не в Сухом Логу.

В режиме Конструктора в строку «Условия отбора» в соответствующие поля вводим два критерия (рис. 2.13)

Поле:	Фамилия	Дата начала работы	Место жительства
Имя таблицы:	Клиенты	Клиенты	Клиенты
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		>#01.03.2013#	<>'Сухой Лог''
или:			

Рис. 2.13

Результат выполнения запроса в режиме просмотра показан на рисунке 2.14.

Фамилия	Дата начал	Место жите
Огнев	20.03.2013	Реж
Иванов	05.03.2013	Реж
*		

Рис. 2.14

Задание 2

Постройте запрос на выборку, позволяющий найти коды клиентов, имеющих средний или мелкий бизнес и какую-нибудь скидку.

Поле:	Код	Вид бизнеса	Скидка
Имя таблицы:	Клиенты	Клиенты	Клиенты
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		"средний"	>0
или:		"мелкий"	>0

Рис. 2.15

Для этого в режиме Конструктора в строку «Условия отбора» введите в соответствующие поля критерии отбора, как показано на рисунке 2.15. Результат выполнения запроса приведен на рисунке 2.16.

Код	Вид бизнеса	Скидка
5	мелкий	0,04
7	средний	0,01
2	средний	0,05
3	мелкий	0,05
*	(№)	0

Рис. 2.16

Задание 3

Постройте запрос с параметром. В качестве параметра будет выступать атрибут «Место жительства». Вводя с клавиатуры значение параметра, можно получить списки клиентов, проживающих в любом городе.

При создании этого запроса необходимо ввести в строку «Условие отбора» фразу [Введите название города] вместе с квадратными скобками, как показано на рисунке 2.17. Скобки означают, что при выполнении запроса стоящая в них фраза будет высвечиваться в окне приглашения для ввода параметра.

Поле:	Фамилия	Имя	Отчество	Место жительства
Имя таблицы:	Клиенты	Клиенты	Клиенты	Клиенты
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				[Введите название города]
или:				

Рис. 2.17

Сохраните запрос и запустите его на выполнение. Чтобы найти клиентов, проживающих, например, в городе Сухой Лог, следует ввести это название в окно приглашения и нажать кнопку ОК. Окно приглашения и результат выполнения запроса можно увидеть на рис. 2.18.

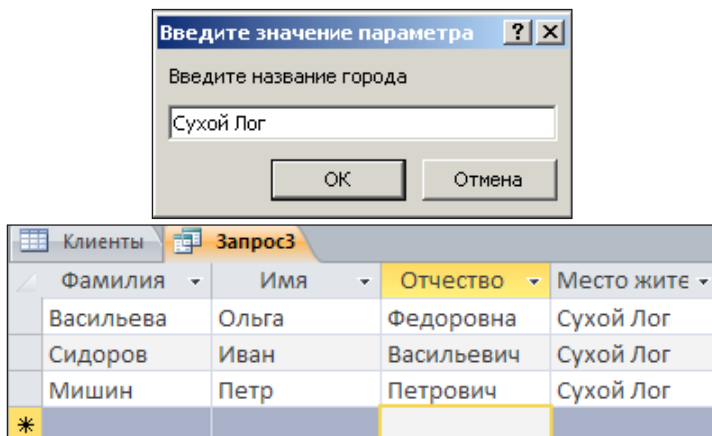



Рис. 2.18

Задание 4

Создадим запрос, подсчитывающий количество клиентов по каждому виду бизнеса.

Для построения запроса используем строку «Групповые операции». Если она не видна в бланке запроса, нужно включить ее с помощью кнопки , расположенной на ленте в режиме конструктора запросов.

Выполним группировку по полю «Вид бизнеса» и подсчитаем количество клиентов по полю «Фамилия», как показано на рисунке 2.19.

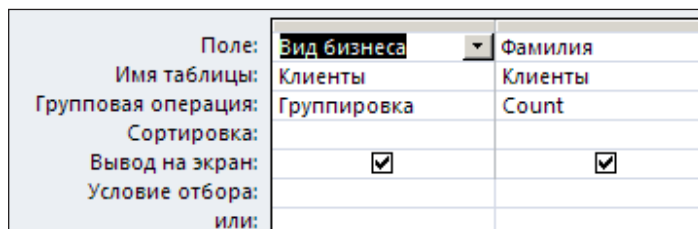
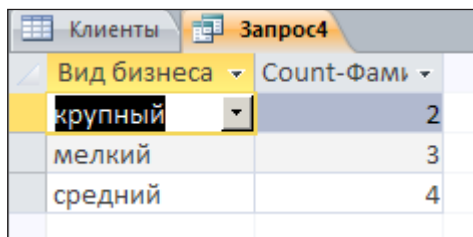


Рис. 2.19

Результат выполнения запроса можно увидеть на рисунке 2.20.



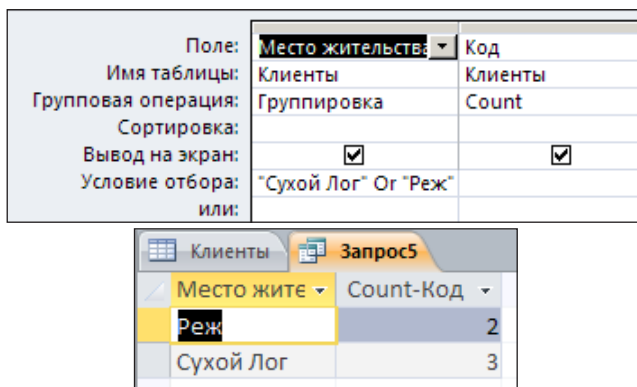
Вид бизнеса	Count-Фами
крупный	2
мелкий	3
средний	4

Рис. 2.20

Задание 5

Просчитайте количество клиентов фирмы в городах Сухой Лог и Реж (отдельно для каждого города).

Для построения запроса снова используем «Групповые операции». На рисунке 2.21 показан запрос в режиме конструктора и в режиме просмотра результатов.



Поле:	Место жительства	Код
Имя таблицы:	Клиенты	Клиенты
Групповая операция:	Группировка	Count
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	"Сухой Лог" Or "Реж"	
или:		

Место жите	Count-Код
Реж	2
Сухой Лог	3

Рис. 2.21

Задание 6

Создадим перекрестный запрос, показывающий распределение клиентов по месту жительства и видам бизнеса. В результирующей таблице нужно отобразить в названии столбцов — место жительства, в названии строк — «Вид бизнеса». Кроме того, требуется вывести итоговое значение, показывающее общее количество клиентов по каждому виду бизнеса.

Такой запрос создается с использованием Мастера запросов. В ходе построения следует выполнить следующие шаги: вызов Мастера запросов, распределение параметров по строкам и столбцам, выбор функции для вычисления итогового значения. На рисунке 2.22 показано последовательное выполнение шагов и результат выполнения запроса.

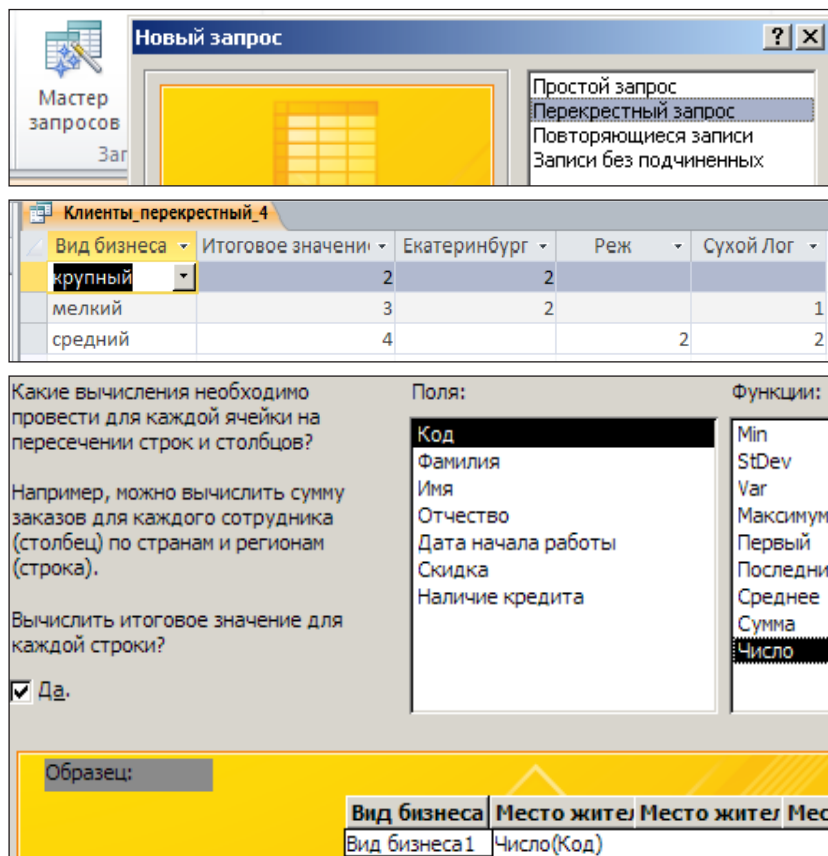


Рис. 2.22

Задание 7

Построим запрос для подсчета количества клиентов, имеющих кредит и проживающих в Екатеринбурге и Реже. Для этого нам потребуется создавать запрос с вычисляемым полем.

Рассмотрим один из способов решения поставленной задачи. Для нахождения искомого значения построим два вспомогательных запроса. В первом подсчитаем количество клиентов, имеющих кредит и проживающих в Реже (рис. 2.23). Второй аналогичным образом построим для Екатеринбурга.

Поле:	Код	Место жительства	Наличие кредита
Имя таблицы:	Клиенты	Клиенты	Клиенты
Групповая операция:	Count	Группировка	Группировка
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		"Реж"	Истина
или:			

Рис. 2.23

Итоговый запрос будем строить на базе двух вспомогательных запросов.

Для его построения необходимо создать вычисляемое поле с именем «Всего». Затем с помощью Построителя выражений записать формулу, показанную на рисунке 2.24.

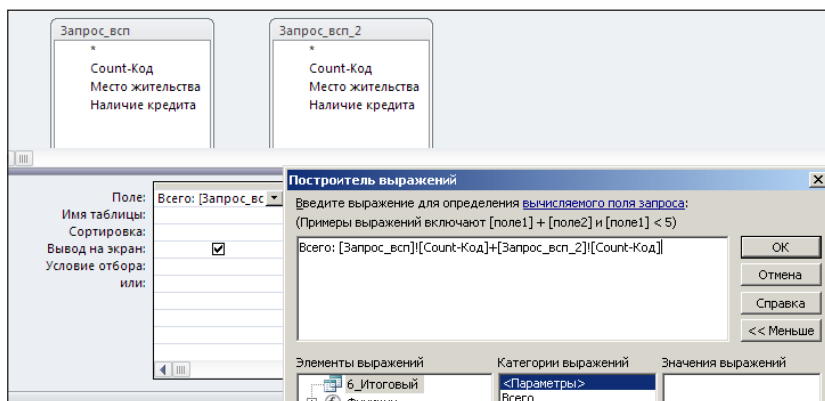


Рис. 2.24

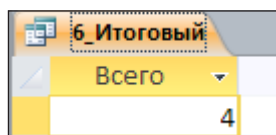


Рис. 2.25

На рисунке 2.25 виден результат выполнения запроса. Результатом в данном случае будет единственное числовое значение.

Лабораторная работа 5

В этой работе вы будете строить различные запросы к базе данных **Телефонная компания**, содержащей связанные таблицы. Поскольку приемы построения запросов различных видов вам уже знакомы, значительную часть работы вы выполните самостоятельно. Подробно останемся лишь на построении запросов на основе нескольких таблиц.

Задание 1

Постройте запрос на выборку абонентов-льготников. Условием отбора будет наличие льготы, т. е. *Льгота* > 0.

Задание 2

Постройте запрос для поиска номера телефона по фамилии абонента.

Задание 3

Постройте запрос на основе таблицы звонков для отбора вызовов внутри сети. Это значит, что номер вызываемого абонента должен начинаться с 999. Для этого в строке Условие отбора для атрибута НомерВызываемого укажите либо шаблон «999*», либо специальный оператор сравнения Like «999*» (см. приложение). Такое условие приведет к отбору записей, у которых поле НомерВызываемого начинается с трех девяток.

Задание 4

Постройте запрос на основе таблицы звонков для отбора вызовов вне своей сети. Это значит, что номер вызываемого абонента НЕ должен начинаться с 999. Для этого в строке Условие отбора для атрибута НомерВызываемого введите not Like «999*». Такое условие приведет к отбору записей, у которых поле НомерВызываемого НЕ начинается с трех девяток.

Задание 5

Постройте запрос на основе таблицы звонков для отбора вызовов абонентов выбираемой пользователем компании. Это значит, что номер вызываемого абонента должен начинаться с введенных пользователем в диалоговом окне цифр (знакомый вам запрос с параметром). Условие отбора в данном случае должно быть таким: Like [Введите значение]+*»*».

Задание 6

Постройте запрос на основе таблицы звонков для отбора вызовов продолжительностью от 3 до 30 секунд.

Задание 7

Постройте запрос на основе таблицы звонков для подсчета количества внутрисетевых вызовов (*подсказка: используйте групповую операцию Count*).

Задание 8

Постройте запрос на основе таблицы звонков для подсчета средней продолжительности разговора (*подсказка: используйте групповую операцию Avg*).

Задание 9

Постройте запрос на основе таблицы звонков для подсчета суммарной продолжительности разговоров с указанного номера (*подсказка: используйте групповые операции*).

Задание 10

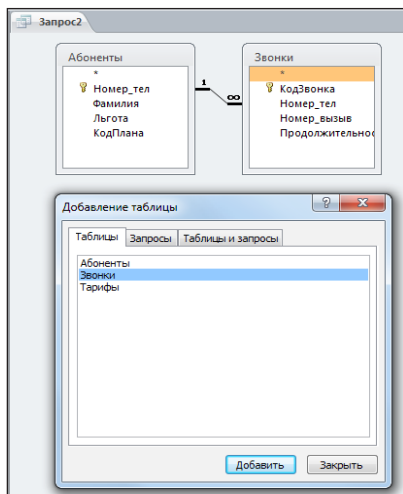


Рис. 2.26

Постройте запрос для отбора всех звонков абонента по указанной фамилии. Для этого запроса нам потребуется информация из двух таблиц, так как данные о звонках содержатся в одной таблице, а фамилии абонентов — в другой. После вызова конструктора запросов в окне Добавление таблицы нужно последовательно выбрать две нужные таблицы (рис. 2.26), после чего закрыть окно. В окне запроса отобразятся две связанные таблицы, после чего в запрос можно включать поля из них обеих. Далее строится знакомый вам запрос с параметром. Включите в него поля НомерВызываемого и Продолжительность.

Задание 11

Постройте запрос для отбора звонков по указанному тарифному плану. Здесь тоже нужны две таблицы — Тарифы и Звонки. Однако эти таблицы не связаны между собой, как это видно на рис. 2.27. Запрос, построенный на несвязанных таблицах, будет работать некорректно (можете это проверить). Для того чтобы добиться правильного результата, нужно включить в запрос связывающую таблицу Абоненты. После чего строится запрос с параметром, как и в предыдущем задании.

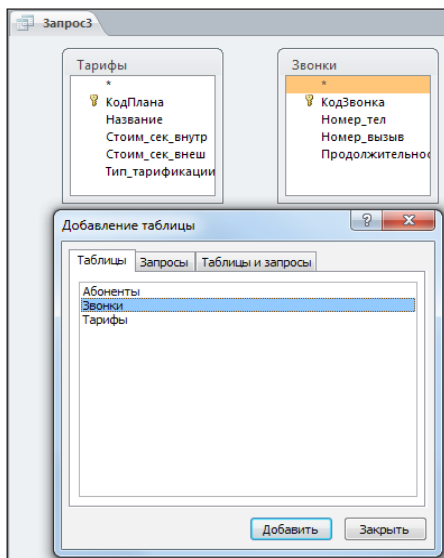


Рис. 2.27

Задание 12

Постройте запрос для вычисления стоимости каждого звонка внутри сети с учетом льготы. Для этого запроса потребуются все три таблицы (из таблицы тарифов берем стоимость секунды, из таблицы звонков — продолжительность, номер телефона звонящего и номер вызываемого, из таблицы абонентов — льготу). Задайте условие отбора, как в задании 3, чтобы отобразить только внутренние звонки. Затем создайте вычисляемое поле, в которое с помощью построителя выражений постройте формулу для расчета:

стоим_внутр: [Звонки]! [Продолжительность] *
[Тарифы]! [Стоим_сек_внутр] * (1- [Абоненты]! [Льгота])

Задание 13

Для поставщика услуг связи необходимо выставлять клиентам счета за обслуживание, в которые включена плата за все звонки, как внутренние, так и внешние. Можно вычислить стоимость внешних звонков подобно предыдущему заданию, но вычислить и просуммировать таким способом стоимость всех звонков не удастся. Придется с помощью построителя выражений строить сложную формулу с использованием встроенной функции If (аналогична функции ЕСЛИ в MS Excel). Общий вид функции If (условие; значение если истина; значение если ложь)

В нашей задаче условие:

[Звонки]! [Номер_вызов] Like «999*»,

значение если истина: [Звонки]! [Продолжительность]* [Тарифы]!
[Стоим_сек_внеш]* (1- [Абоненты]! [Льгота]),

значение если ложь: [Звонки]! [Продолжительность]* [Тарифы]!
[Стоим_сек_внеш]* (1- [Абоненты]! [Льгота]).

На рисунке 2.28 показано, как выбрать описанную функцию (Встроенные функции → Управление → If) и как выглядит построенное выражение.

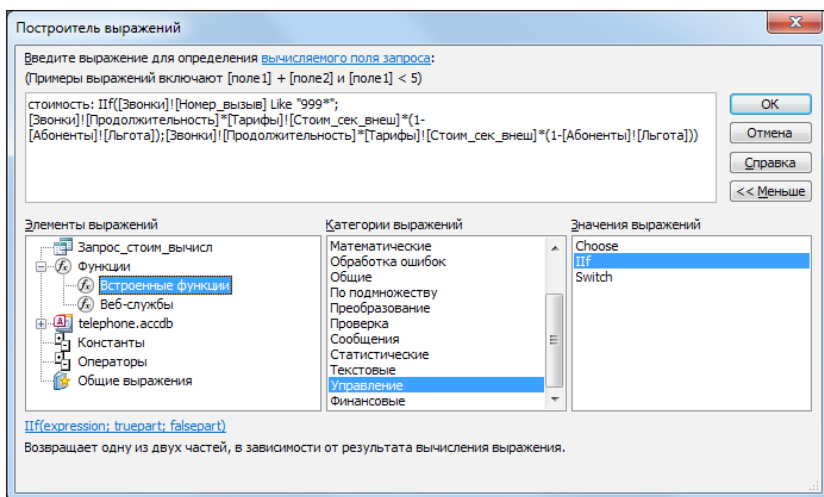


Рис. 2.28

Постройте запрос описанным образом, включите в него также фамилию абонента. Убедитесь, что полученный запрос правильно вычисляет стоимость всех звонков. Для этого подсчитайте вручную стоимость звонков какого-либо абонента, имеющего льготу и сделавшего более одного звонка. Такая проверка позволит убедиться, что суммирование и учет льгот выполняются правильно.

Задание 14

На основе запроса из предыдущего задания постройте запрос для вычисления стоимости всех звонков для конкретного абонента по указанному номеру телефона.

Лабораторная работа 6

Таблицы и запросы, построенные нами для БД «Телефонная компания», представляют собой основу базы данных, можно сказать, ее кухню. Они видны только разработчику базы, но не ее пользователю. Для общения с пользователем служат формы и отчеты — экранные и печатные документы, позволяющие выполнять информационное обслуживание бизнес-процессов, не зная структуры и способов работы в СУБД MS Access.

В этой лабораторной работе мы создадим формы для ввода и просмотра данных нашей БД.

Задание 1

Откройте БД «Телефонная компания» и выберите вкладку Создание, а на ней Мастер форм. Создадим форму для ввода данных в таблицу абонентов. Выберем нужную таблицу и перенесем все содержимое из окна Доступные поля в окно Выбранные поля, как показано на рисунке 2.29.

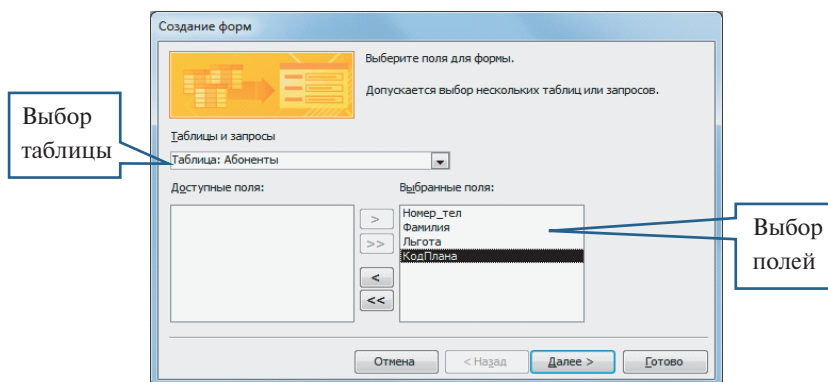


Рис. 2.29

После нажатия кнопки Далее выбираем внешний вид формы. Для формы абоненты пусть это будет «в один столбец». Нажимаем кнопку Готово и открываем форму для просмотра. С помощью этой формы можно просматривать записи таблицы поочередно, перемещаясь от одной к другой при помощи кнопок внизу окна, а также перейти к вводу новой записи. На рисунке 2.30 показаны управляющие кнопки формы.

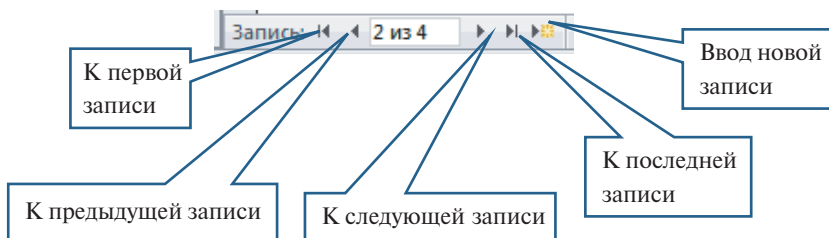


Рис. 2.30

Задание 2

Создайте с помощью Мастера форм ленточную форму для ввода данных в таблицу Тарифы.

Задание 3

Создайте с помощью Мастера форм форму в один столбец для ввода данных о звонках.

Задание 4

Иногда бывает необходимо на одной форме отображать данные сразу двух таблиц, например, выбрав конкретного абонента, видеть перечень его звонков (главная и подчиненная форма). Создадим с помощью Мастера такую форму. Для этого на этапе выбора перенесем в окно Выбранные поля последовательно поля из таблиц Абоненты и Звонки. Далее выделим таблицу Абоненты, после чего Access предложит нам соответствующий нашей задаче вид представления данных (рис. 2.31).

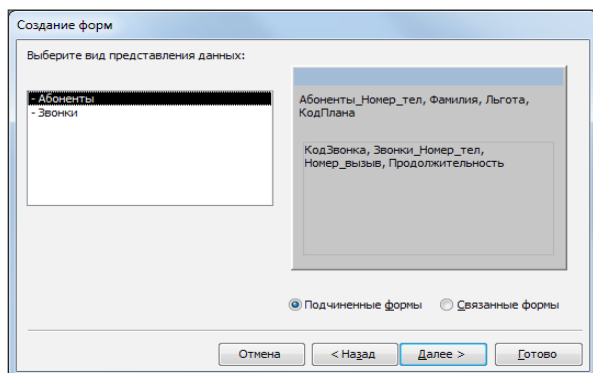


Рис. 2.31

Выберем вариант Подчиненные формы, далее выберем внешний вид подчиненной формы — табличный и — готово, увидим результат (рис. 2.32). Переходя с помощью управляющих кнопок от одного абонента к другому, мы можем просматривать его звонки.

КодЗвонка	Номер_выз	Продолжит
1	99900012333	20
3	34343434	6

Рис. 2.32

Задание 5

Создайте таким же образом сложную форму, где главной частью будут тарифные планы, а подчиненной — абоненты, которые подключены к данному тарифному плану. Внешний вид формы, созданной мастером, можно при необходимости изменить. Для этого откройте сохраненную форму в режиме конструктора и отредактируйте ее (измените размеры полей, добавьте дополнительную надпись).

Задание 6

Форма может базироваться не только на таблице, но и на запросе. Постройте форму на основе запроса из задания 12 предыдущей лабораторной работы (с вычислением стоимости каждого звонка внутри сети с учетом льготы).

Лабораторная работа 7

Эта лабораторная работа посвящена созданию отчетов, т. е. итоговых документов, которые предполагается выводить на печать. Отчеты, подобно формам, могут базироваться как на таблицах, так и на запросах. MS Access предоставляет несколько способов создания отчетов. Мы воспользуемся **Мастером отчетов**, а затем отредактируем внешний вид с помощью конструктора.

Порядок создания отчета с помощью Мастера.

1. Откройте вкладку **Создание** и нажмите на ленте кнопку **Мастер отчетов**.
2. В качестве источника данных выберите таблицу или запрос → Далее.
3. Выберите поля для отчета → Далее.
4. При необходимости установите уровни группировки, т. е. укажите, по каким признакам будут группироваться записи в отчете → Далее.
5. Можно выбрать порядок сортировки записей в отчете по одному или нескольким полям (пропустите этот шаг) → Далее.
6. Выберите вид макета для отчета → Далее.
7. Задайте название отчета. Затем можно открыть отчет для просмотра либо для редактирования — выберите просмотр → Готово.
8. Просмотрите и закройте отчет.

Затем созданный отчет можно открыть в режиме конструктора и внести необходимые изменения.

Задание 1

В базе данных *Телефонная Компания* с помощью **Мастера** создайте отчет на основе запроса о льготниках. Сохраните отчет под именем **Льготники1** и просмотрите полученный результат.

Задание 2

Сделайте копию отчета **Льготники1** (скопируйте и вставьте в этом же окне). Назовите полученную копию **Льготники2**. Откройте этот отчет в режиме конструктора для внесения изменений. В разделе **Заголовок** измените заголовок отчета на такой: «Абоненты, имеющие право на льготы». В разделе **Примечание отчета** добавьте надпись: «Ознакомлен _____ инспектор Иванов И. И.».

Для добавления надписи сначала расширьте мышью раздел **Примечание отчета** — он находится в самом низу и может быть вообще не виден. Затем на ленте в разделе Элементы управления выберите элемент Надпись и поместите его в разделе примечаний, как показано на рисунке 2.33. Введите в рамку нужный текст. Сохраните и закройте отчет. Просмотрите полученный результат.

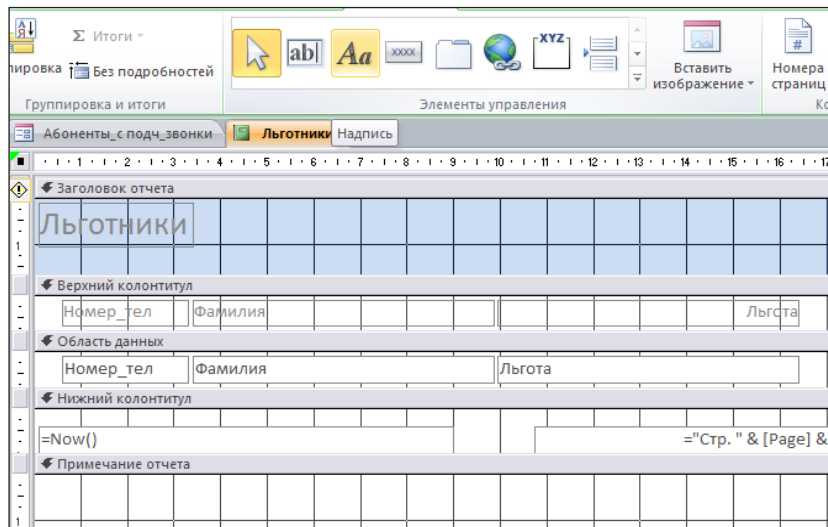


Рис. 2.33

Задание 3

Сделайте еще одну копию отчета **Льготники1** и назовите **Льготники3**. Откройте этот отчет в режиме конструктора. В раздел примечаний добавьте итоговую функцию для подсчета количества льготников. Для этого в разделе Элементы управления выберите элемент Поле и поместите его в разделе примечаний. Введите в рамку, как показано на рисунке 2.34, вместо слова Поле поясняющий текст «Количество льготников», а вместо слова Свободный — формулу `=count ([Фамилия])`. Если для поясняющего текста недостаточно места, рамки можно расширить и переместить. Сохраните и закройте отчет. Просмотрите полученный результат.

Аналогично можно использовать в отчетах другие итоговые функции — `sum` (сумма), `avg` (среднее) и т. п.

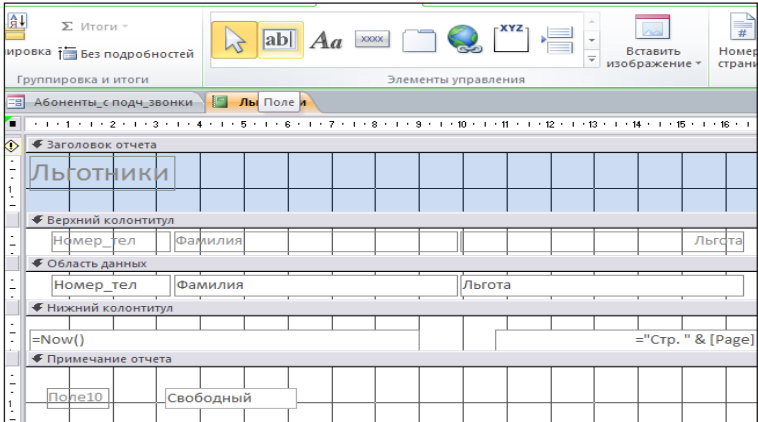


Рис. 2.34

Задание 4

На основе таблицы звонков создайте отчет с группировкой по номеру телефона. В разделе примечаний отчета вычислите среднюю продолжительность разговора.

Задание 5

На основе той же таблицы звонков создайте отчет с группировкой по тарифному плану. Обратите внимание: в таблице звонков нет информации о тарифном плане, она находится в таблице абонентов. Для того чтобы включить ее в отчет, нужно на этапе выбора источника данных (шаг 2 Мастера отчетов) после выбора полей таблицы звонков добавить еще одно поле из таблицы абонентов (рис. 2.35).

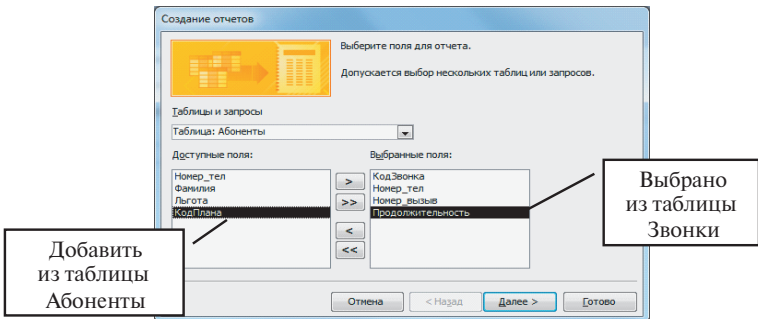


Рис. 2.35

Задание 6

Создайте отчет для предоставления абоненту информации обо всех его звонках с расчетом стоимости каждого и итоговой суммой по всем звонкам. Создавать его следует на основе запроса из **задания 13** лабораторной работы 7 с добавлением итоговой суммы в разделе примечаний отчета.

Лабораторная работа 8

В этой работе вам предстоит самостоятельно создать базу данных для фирмы, занимающейся продажей товаров.

Задание 1. Таблицы

а. Создать таблицу КЛИЕНТ, содержащую следующие поля:

- код клиента;
- имя клиента (ФИО);
- название фирмы клиента;
- дата первого заказа;
- скидка клиента;
- город;
- телефон;
- оборот;
- надёжность (средняя, высокая, низкая).

В таблицу ввести восемь клиентов.

б. Создать таблицу ТОВАР, содержащую следующие поля:

- артикул товара;
- группа товара (например, 8 С);
- цена;
- количество;
- страна (по умолчанию Китай);
- сертификат качества (документ MSWord).

В таблицу ввести десять записей. Артикул не повторяется.

с. Создать таблицу ЗАКАЗ, содержащую следующие поля:

- номер заказа;
- код клиента;
- артикул;
- количество в заказе;
- дата заказа;
- выполнение (ДА, НЕТ);

- оплата (ДА, НЕТ);
- договор (ссылка на документ MSWord).

В таблицу ввести двенадцать записей. Клиенты могут делать несколько заказов.

Задание 2. Схема данных

- d. Создать Схему Данных. Установить типы связи между таблицами.
- e. Упорядочить записи:
 - 1. В таблице КЛИЕНТ по именам клиентов по возрастанию.
 - 2. В таблице ЗАКАЗ по датам заказов по возрастанию.
 - 3. В таблице ТОВАР по полю КОЛИЧЕСТВО по убыванию.
- f. Создать расширенные фильтры к таблице КЛИЕНТ:
 - 1. Клиенты из Москвы с высокой степенью надежности. Сохранить фильтр как запрос под именем КФ1.
 - 2. Список клиентов, упорядоченный по городам, а внутри городов по именам клиентов по возрастанию. Сохранить фильтр как запрос под именем КФ2.
- g. Создать фильтры к таблице ТОВАР:
 - 1. Список товаров, имеющих сертификаты. Сохранить фильтр как запрос под именем ТФ1.
 - 2. Список товаров всех групп за исключением 5 А и 3 А. Сохранить фильтр как запрос под именем ТФ2.
- h. Создайте фильтр к таблице ЗАКАЗ:
 - 1. Список выполненных и неоплаченных заказов 2012 года. Сохранить фильтр как запрос под именем ЗФ1.
 - 2. Список заказов, в которых товары заказаны в количестве более 50 штук и есть договор (или ссылка на него) ЗФ2.

Задание 3. Запросы к одной таблице

- 1. Создать запрос к таблице ЗАКАЗ, содержащий сведения о заказах, сделанных в течение последних 200 дней. Включить в запрос данные: номер заказа и количество. Сохранить под именем З1.
- 2. Создать запрос к таблице КЛИЕНТ, содержащий сведения о клиентах, с которыми фирма начала работать в 2007 году. Включить в запрос данные: год начала работы с клиентом, имя клиента, город, телефон, оборот. Сохранить под именем К1.
- 3. Создать запрос к таблице КЛИЕНТ, содержащий сведения о клиентах из Москвы и Екатеринбурга, упорядоченный по го-

роду и имени клиента. Включить в запрос данные: имя клиента, город, телефон, оборот, год начала работы с клиентом. Сохранить под именем К2.

4. Создать запрос к таблице ЗАКАЗ, включающий номер клиента, общий объем заказов клиента фирме в денежном выражении, количество заказов клиента фирме. Сохранить под именем З2.

Задание 4. Запросы к связанным таблицам

1. Список клиентов (имена, города, телефон, номер заказа, дата заказа, артикул товара), делавших заказы на товар с указанным артикулом (сделать артикул параметром запроса). Сохранить под именем ЗСТ1.
2. Список всех заказов, их стоимость с учетом скидки. Сохранить под именем ЗСТ2.
3. Список неоплаченных выполненных заказов с указанием номера заказа, имен клиентов, сделавших заказ, даты заказа, артикула товара, группы товара и объёма заказа в денежном выражении. Сохранить под именем ЗСТ3.
4. Список клиентов-должников (номер клиента, имя клиента, город) с указанием общей суммы долга каждого клиента фирме и количества неоплаченных им заказов, упорядоченный по общей сумме долга. Сохранить под именем ЗСТ4.
5. Список всех товаров, включающих поля артикул, количество, продано, остаток. Сохранить под именем ЗСТ5.
6. Список всех клиентов с указанием количества заказов и общей суммы заказов каждого клиента фирме. Сохранить под именем ЗСТ6.

Задание 5. Перекрёстные запросы

1. Построить таблицу распределения объемов заказов по клиентам (по строкам номер клиента) и годам. Сохранить под именем ЗП1.
2. Построить таблицу распределения количества клиентов по городам и степеням надежности. Сохранить под именем ЗП2.
3. Построить таблицу распределения объемов заказов по годам и месяцам, включив в запрос итоги по строкам — общие продажи в каждом году. Сохранить под именем ЗП3.
4. Построить таблицу распределения продаж (в денежном выражении) каждого артикула в каждом году. Сохранить под именем ЗП4.

Задание 6. Запросы на изменение

1. Создать таблицу КЛИЕНТ1 на основе таблицы КЛИЕНТ, содержащую список клиентов из Москвы. Сохранить под именем ИЗМ1.
2. Создать запрос на обновление к таблице КЛИЕНТ1, в котором заменить значение поля «город» Москва на Уфу, номера клиентов увеличить на 50 и увеличить значение поля «оборот» на 5000 рублей. Сохранить под именем ИЗМ2.
3. Создать запрос на добавление записей о клиентах с высокой степенью надежности из таблицы КЛИЕНТ1 в таблицу КЛИЕНТ. Сохранить под именем ИЗМ3.
4. Создать запрос на удаление из таблицы КЛИЕНТ записей о клиентах из Уфы. Сохранить под именем ИЗМ4.

Задание 7. Формы

1. Построить форму на дополнение списка клиентов. Сохранить под именем ФК1.
2. Построить автоформу на дополнение заказов. Сохранить под именем Ф3.
3. Построить форму для просмотра суммарной стоимости заказов каждого клиента. Сохранить под именем ФК2.
4. Построить форму, содержащую артикул товара и номера заказов, его содержащих. Сохранить под именем ФС1.

Задание 8. Отчеты

1. Создать отчет за указанный период, содержащий фамилию клиента, его заказы и их стоимость, предварительно создав необходимые запросы.
2. Создать отчет, отображающий продажи товаров по времени.
3. Создать отчет о долгах фирме, содержащий номер заказа, фамилию клиента и величину долга.

Задание 9. Кнопочная форма

Построить кнопочную форму, содержащую четыре кнопки. Первая — для вывода формы на добавление клиента. Вторая — для вывода формы на добавление заказов. Третья — для просмотра отчета по выполнению и оплате заказов (отчет должен быть сформирован на альбомной странице, содержать столбцы: номер заказа, дата заказа, выполнение и оплата). Четвертая — для выхода из приложения.

Лабораторная работа 9

Результатом работы должна стать база данных для небольшой библиотеки. Она должна содержать следующую информацию.

1. О книжном фонде: автор, название, год издания, издательство, инвентарный номер, тип издания — учебное, художественное, справочное.
2. О читателях: фамилия, имя, отчество, номер читательского билета, категория — студент, преподаватель, сотрудник.
3. О выдаче книг: что, когда и кому выдано, возвращено ли обратно. При возврате запись о выдаче книги не удаляется, а хранится до специального указания.
4. Для каждого типа изданий определены сроки пользования: учебные — 300 дней, художественные — 10 дней, справочные — 3 дня.

В каждой из таблиц целесообразно определить ключевое поле — одно из заданных полей или дополнительное поле — счетчик.

Задание 1

Подготовьте нужные таблицы, установите связи между ними. Заполните таблицы информацией: 5 читателей разных категорий, 10 книг разных типов, 25 записей о выдаче книг (часть должна быть уже возвращена, часть еще на руках у читателей).

Задание 2

На основании созданных таблиц подготовьте следующие запросы.

1. Перечень всех изданий, выпущенных указанным издательством за последние пять лет.
2. Перечень всех изданий, находящихся на руках у читателей с указанием даты истечения срока пользования.
3. Перечень всех задержанных книг (срок пользования истек, а книга не возвращена).
4. Список всех читателей — должников (если читатель задержал несколько книг, то он должен перечисляться в этом списке один раз).
5. Перечень всех книг, находящихся у указанного читателя.
6. Данные о читателе, взявшем данную книгу (по указанному инвентарному номеру).
7. Перечень всех книг, у которых сегодня истекает срок пользования.

8. Информация о читателе (фамилия, имя, отчество и перечень книг, которыми он когда-либо пользовался).
9. Информация о книге (автор, название, перечень читателей: кто и когда ею пользовался)
10. Спрос на различные виды изданий — количество выданных изданий каждого вида за указанный период.

Задание 3

Для удобства работы персонала библиотеки подготовьте следующие формы.

1. Добавление читателя.
2. Добавление издания.
3. Информация об указанном читателе (перечень книг, находящихся у него на руках с указанием, когда книга должна быть возвращена).
4. Информация обо всех читателях с перечнем всех книг, которыми каждый из них когда-либо пользовался (главная форма — читатели, подчиненная — выданные книги).

Задание 4

Подготовьте следующие отчеты.

1. Информация о читателе (фамилия, имя, отчество и перечень книг, которыми он когда-либо пользовался).
2. Информация о книге (кто и когда ею пользовался).
3. Спрос на различные виды изданий (рис. 2.36).

Спрос на литературу по типам	
Тип издания	Количество
Научная	4
Справочная	10
Художественная	6

Рис. 2.36

Задание 5

Подготовьте кнопочную форму для удобства работы с базой данных.

Библиографический список

1. Информационные технологии / Д. Н. Угринович. — СПб., 2002.
2. Microsoft Office 97 / Р. Винтер, П. Винтер. — М.: ВНУ — Санкт-Петербург, 1998.
3. Новые информационные технологии / под ред. В. П. Дьяконова. — СПб.: ВНУ — Санкт-Петербург, 2004.
4. Проектирование баз данных в СУБД Access / О. А. Житкова, М. А. Журина, Е. К. Кудрявцева. — М.: Интеллект-Центр, 2006.
5. Работа с приложением Microsoft Access / Е. А. Бармина. — Екатеринбург: УрГУПС, 2009.
6. Экономическая информатика / под ред. П. В. Конюховского и Д. Н. Колесова. — СПб., 2001.

Приложение 1

Варианты заданий

Вариант 1

База данных «Конференция»

База данных содержит три связанные таблицы:

1. Участники (ФИО, возраст, регион, заявка на гостиницу, количество дней пребывания).
2. Проезд из регионов в днях.
3. Доклады участников (ФИО участника и тема доклада).

Создать запросы:

1. ФИО докладчика и его регион.
2. Темы докладов из определенного региона (регион — параметр).
3. Список участников, которым требуется гостиница.
4. Список молодых специалистов (возраст до 30 лет).
5. Участники без докладов.
6. Количество дней, затраченных участниками с учетом дней проезда.
7. Просчитать количество участников по каждому региону.
8. Создать таблицу, содержащую список участников с пребыванием на конференции более двух дней.

Создать формы:

1. Добавление участника.
2. Добавление доклада.
3. Докладчики по темам.
4. Молодые специалисты.
5. Тема и докладчики (главная и подчинённая формы).

Создать отчеты:

1. Участники из определённого региона.
2. Заявки на гостиницу.
3. Молодые специалисты.

Создать кнопочную форму, содержащую четыре кнопки: две для открытия форм на добавление, третья — для открытия отчета по заявкам на гостиницу и четвертая — для выхода из приложения.

Вариант 2

База данных «Зоопарк»

База данных содержит три связанные таблицы:

1. Животные: название, возраст, стоимость содержания в месяц, сектор нахождения.
2. Рацион: животное, основное питание (сено, мясо, рыба, овощи), номер клетки.
3. Сектора: сектор (А, В, С...), вид животных (крупные хищники, крупные травоядные, медведи, ...). Каждый вид животных располагается в своем секторе.

Создать запросы:

1. Список всех животных в возрасте от 7 до 15 лет.
2. Список всех травоядных животных.
3. Сумма, идущая на содержание всех травоядных животных в месяц.
4. Сумма, идущая на содержание всех животных указанного сектора (сектор параметр) в месяц.
5. Список всех животных, питающихся мясом, на содержание которых уходит меньше 50 000 р. в год.
6. Информация о всеядных животных (стоимость содержания, сектор, возраст).
7. Создать таблицу, содержащую список животных старше 12 лет.

Создать формы:

1. Форму в один столбец для добавления животного.
2. Ленточную форму для просмотра списка всех травоядных.
3. Главную форму, содержащую название сектора, и подчиненную, в которой указаны все животные, находящиеся в нем.
4. Кнопочную форму для вызова всех форм и отчетов.

Создать отчеты:

1. Сумма, идущая на содержание всех травоядных животных.
2. Список всех животных, питающихся мясом, на содержание которых уходит меньше 50 000 р. в год.

3. Информация о всеядных животных.

Создать кнопочную форму, содержащую четыре кнопки: две для открытия форм, третья — для открытия отчета и четвертая — для выхода из приложения.

Вариант 3

База данных «Туристическая фирма»

Фирма организует экскурсионные, экстремальные и оздоровительные туры по Южной Америке, Европе, России. Клиенту предоставляется возможность выбрать длительность путешествия самостоятельно.

База данных состоит из трех связанных таблиц: «Клиент», «Путевка» и «Договор»:

1. Таблица «Клиент»: код клиента, ФИО, телефон, город, номер паспорта, номер страховки.
2. Таблица «Договор»: номер договора, код клиента, код тура, дата отъезда, продолжительность тура в неделях.
3. Таблица «Путевка»: код тура, название тура, место пребывания, вид путевки, стоимость недели пребывания, питание — включение/невключение в стоимость путевки.

Создать запросы:

1. Список всех туристов с продолжительностью тура более одной недели.
2. Список туров, где питание включено в стоимость путевки.
3. Перечень договоров и стоимость соответствующих туров.
4. Список туристов и стоимость купленных ими туров. Некоторые клиенты обращались в фирму несколько раз.
5. Сумма, полученная фирмой за реализацию всех путевок указанного тура (код тура — параметр).
6. Перечень названий туров и количество клиентов, их выбравших.
7. Информация о востребованности разных видов туров.
8. Создать таблицу, содержащую номера договоров с продолжительностью тура две недели.

Создать формы:

1. Форма для добавления клиента
2. Форма для добавления тура
3. Главную форму, содержащую название тура, и подчиненную, в которой указаны все клиенты, которые им воспользовались.

Создать отчеты:

1. Информация о клиенте и дате отъезда.
2. Информация о номере страховки клиента.
3. Информация о стоимости путевки за неделю.

Подготовить кнопочную форму для всех форм и отчетов.

Вариант 4

База данных «Образовательные услуги»

Фирма оказывает платные образовательные услуги по специальностям. База данных состоит из четырех связанных таблиц:

1. Таблица «Студенты»: ФИО, номер зачетки, возраст в годах.
2. Таблица «Преподаватели»: ФИО, специальность.
3. Таблица «Предлагаемые специальности»: название, стоимость.
4. Таблица «Обучение»: ФИО студента, специальность, сроки обучения — начало и конец.

Создать запросы:

1. О студентах определенной специальности. Специальность — параметр.
2. Список студентов, которые начали обучение за последние 200 дней. В запросе использовать функцию date ().
3. Распределение студентов по специальностям.
4. О прибыльности обучения по различным специальностям.
5. Об общей стоимости всех услуг.
6. Окончание студентами курсов по месяцам и специальности (перекрестный).
7. Создать таблицу, содержащую ФИО студента и специальность.

Создать формы:

1. Для добавления студента.
2. Ленточная форма для просмотра специальностей.
3. Главную форму, содержащую название специальности и подчиненную, содержащую список всех студентов по этой специальности.
4. Кнопочная форма для вызова форм и отчетов.

Создать отчеты:

1. О прибыльности различных видов специальностей.
2. О студентах определенной специальности

3. О суммарной стоимости оказанных услуг.

Создать кнопочную форму, содержащую четыре кнопки: две для открытия форм, третья — для открытия отчета и четвертая — для выхода из приложения.

Вариант 5

База данных «Гастроли»

Фирма организует гастроли артистов по городам России. Артист может принимать участие в организации разных концертов.

База данных состоит из трех связанных таблиц: «Участники», «Концерты» и «График концертов».

1. Таблица «Участники»: ФИО, возраст в годах, наличие фонограммы, город проведения. При оплате концертов учитываются звания «Заслуженный артист» и «Народный артист». В первом случае оплата повышается на 25 %, а во втором — на 40 %.
2. Таблица «Концерты»: название, стоимость.
3. Таблица «График концертов»: название концерта, город и дата проведения.

Создать запросы:

1. Перечень городов, в которых будут проведены концерты за указанный период (период — параметр).
2. Список участников старше 25 лет, которые используют на концерте фонограмму.
3. Число участников по разным названиям концертов.
4. Список участников, имеющих званий «Заслуженный артист» и «Народный артист».
5. Количество участников: мужчины старше 40 лет и женщины старше 35 лет.
6. Названия концертов и их общее количество.
7. График проведения концертов: название концерта, месяц проведения и город (перекрестный).

Создать формы:

1. Для добавления артиста.
2. Ленточная форма для просмотра города проведения концертов, даты и названия.
3. Главную форму, содержащую название концерта и подчиненную, содержащую список всех участников этого концерта.

Создать отчеты:

1. О выплатах участникам концертов.
2. Об участниках определенного концерта.
3. О стоимости каждого концерта.

Подготовить кнопочную форму, открывающую отчеты.

Вариант 6

База данных «Отдел материально-технического снабжения»

Отдел организует поставку необходимых материалов для функционирования предприятия и оплату работ. Сотрудники отдела делают закупки и пишут отчеты о затраченных средствах и оплаченных работах.

База данных состоит из трех связанных таблиц: «Сотрудники», «Виды затрат» и «Отчеты».

1. Таблица «Сотрудники»: ФИО, должность, наличие электронного адреса, рабочий телефон. При оплате закупок некоторые сотрудники имеют личные льготы от 1 до 10 % (по умолчанию — 0), которые впоследствии учитываются в их заработной плате.
2. Таблица «Виды затрат»: наименование, стоимость.
3. Таблица «Отчеты»: код сотрудника, наименование затрат, дата проведения операции, предварительная оплата.

Создать запросы:

1. Перечень затрат, за которые был выполнен окончательный расчет за указанный период (период — параметр).
2. Список сотрудников, имеющих льготы и электронный адрес.
3. Число сотрудников, оформивших отчеты на разные виды затрат.
4. Список сотрудников, которые при выполнении операции оформляли предварительную оплату.
5. Окончательный расчет по каждой операции с учетом предварительной оплаты и льгот сотрудника.
6. Виды затрат и общие потраченные на них суммы с учетом льгот и предварительной оплаты.
7. Количество видов затрат по месяцам выполнения (перекрестный).

Создать формы:

1. Для добавления вида затрат.
2. Ленточная форма для просмотра льгот в денежном выражении при оплате по отчетам.
3. Главную форму, содержащую код сотрудника, и подчиненную, содержащую название видов затрат, по которым он делал отчет.

Создать отчеты:

1. О сотрудниках отдела, имеющих более одного отчета.
2. О сумме затрат по каждому отчету.
3. О количестве разного вида затрат.

Подготовить кнопочную форму, открывающую отчеты.

Вариант 7

База данных «Санаторий»

База данных состоит из трех связанных таблиц: «Отдыхающие», «Услуги» и «Назначения»:

1. Таблица «Отдыхающие»: код, ФИО, возраст в годах, диагноз, форма размещения (с оплаченным проживанием и лечением в санатории или с оплаченным курсом лечения — курсовка), дата заезда, количество дней, льгота по умолчанию — 0.
2. Таблица «Услуги»: название, стоимость.
3. Таблица «Назначения»: ФИО, название услуги, количество.

Создать запросы:

1. Перечень услуг, которые будут получены отдыхающими за указанный период (период- параметр).
2. Список отдыхающих старше 25 лет, которые пользуются льготой.
3. Список отдыхающих и дата их отъезда.
4. Число отдыхающих, приехавших по курсовкам.
5. Суммарная стоимость льгот, предоставленных отдыхающим.
6. Суммарная стоимость услуг каждого отдыхающего.
7. Заезд отдыхающих по датам и по форме размещения (перекрестный).
8. Создать таблицу, содержащую список отдыхающих с пребыванием в санатории более десяти дней.

Создать формы:

1. Для добавления отдыхающего.
2. Ленточная форма для просмотра списка отдыхающего, даты его заезда и отъезда.
3. Главную форму, содержащую название услуги и подчиненную, содержащую список всех отдыхающих, получивших эту услугу.

Создать отчеты:

1. О стоимости услуг санатория для каждого отдыхающего.
2. О количестве назначенных услуг каждого вида.
3. Список отдыхающих на дату (дата — параметр).

Создать кнопочную форму, содержащую четыре кнопки: две для открытия форм, третья — для открытия отчета и четвертая — для выхода из приложения.

Вариант 8

База данных «Продажа автомобилей»

Фирма организует продажу автомобилей.

База данных состоит из трех связанных таблиц: «Клиенты», «Автомобили» и «Наличие в продаже»:

1. Таблица «Клиенты»: код клиента, ФИО, год рождения, номер счета, оборот.
2. Таблица «Автомобили»: марка автомобиля, год выпуска, цвет, число скоростей, мощность, вид горючего, инжектор.
3. Таблица «Продажа»: код записи, код клиента, марка автомобиля, продающая фирма, количество, цена, изображение.

Создать запросы:

1. Марки автомобилей, имеющих в продаже на фирме (фирма-параметр) и стоимость которых превышает 300 000 р.
2. Список клиентов старше 25 лет, оборот которых не меньше 500 000 р.
3. Суммы в денежном выражении, которые фирмы вложили в продаваемые машины.
4. Стоимость машин с учетом наценки фирмы.
5. Число клиентов, которые могут купить автомобиль дороже 400 000 р.

6. Количество автомобилей по маркам и по году выпуска (перекрестный).
7. Создать таблицу, содержащую список автомобилей, выпущенных за последние три года.

Создать формы:

1. Для добавления клиента.
2. Ленточная форма для просмотра остатка денежных средств клиентов, которые купили автомобиль дороже 400 000 р.
3. Главную форму, содержащую название фирмы, и подчиненную, содержащую список автомобилей, продаваемых фирмой.

Создать отчеты:

1. О стоимости автомобилей после наценки.
2. Список фирм, которые продают модификации ВАЗа.
3. О клиентах, которые могут купить автомобиль дороже 400 000 р.

Создать кнопочную форму, содержащую четыре кнопки: две для открытия форм, третья — для открытия отчета о стоимости автомобилей после наценки и четвертая — для выхода из приложения.

Вариант 9

База данных «Ателье»

База данных состоит из трех связанных таблиц: «Ассортимент услуг», «Ткани» и «Изготовление»:

1. Таблица «Изготовление»: код, клиент, изделие, коэффициент сложности, ткань, количество ткани в метрах, дата заказа, выполнение. Коэффициент сложности имеет три значения: 1; 1,4; 1,8.
2. Таблица «Ассортимент услуг»: название, базовая стоимость, срок изготовления в днях.
3. Таблица «Ткани»: ткань, цена за метр.

Создать запросы:

1. Перечень изделий, заказанных за указанный период (период — параметр).
2. Список клиентов, заказавших изделия с коэффициентом сложности 1,4 или 1,8. Повторяющихся строк в запросе быть не должно.
3. Список заказанных изделий и даты их готовности.

4. Список клиентов и количество их заказов.
5. Список клиентов и стоимость каждого изделия, заказанного ими с учетом базовой стоимости изготовления, коэффициента сложности, стоимости и количества ткани.
6. Суммарная стоимость услуг каждого клиента.
7. Распределение изделий по клиентам и коэффициентам сложности (перекрестный).
8. Создать таблицу, содержащую список заказов, имеющих коэффициент сложности 1,8.

Создать формы:

1. Для добавления ассортимента.
2. Ленточная форма для просмотра списка клиентов и даты заказов изделий.
3. Главную форму, содержащую название ткани и подчиненную, содержащую список изделий, выполненных из этой ткани.

Создать отчеты:

1. О стоимости изделий ателье для каждого клиента.
2. О количестве изделий ателье каждого вида ассортимента.
3. Список незаконченных изделий на определенную дату (дата — параметр).

Создать кнопочную форму, содержащую четыре кнопки: две для открытия форм, третья — для открытия отчета о количестве изделий ателье каждого вида ассортимента и четвертая — для выхода из приложения.

Вариант 10

База данных «Транспортные перевозки»

База данных состоит из трех связанных таблиц: «Перевозки», «Транспортные средства» и «Клиенты»:

1. Таблица «Заказы»: код, клиент, транспортное средство, коэффициент комфортности, продолжительность аренды в часах, дата заказа, предварительная оплата. Коэффициент комфортности имеет четыре значения: 0,7; 1; 1,3; 1,9.
2. Таблица «Транспортные средства»: вид, базовая стоимость аренды за один час.

3. Таблица «Клиенты»: ФИО, вид деятельности, начало работы с фирмой, льгота от 5 % до 15 % (по умолчанию — 0), долг в денежном выражении.

Создать запросы:

1. Перечень заказов с предварительной оплатой за указанный период (период — параметр).
2. Список клиентов, которые пользуются транспортными средствами коэффициентом комфортности 1 и 1,3.
3. Список клиентов, код заказа и его стоимость с учетом комфортности и льготы.
4. Количество заказанных транспортных средств каждого вида.
5. Суммарная стоимость заказов каждого клиента с учетом предварительной оплаты.
6. Окончательный расчет каждого клиента с фирмой с учетом долга.
7. Выбор комфортности в заказах клиентов по видам транспортных средств (перекрестный).

Создать формы:

1. Для добавления клиента.
2. Ленточная форма для просмотра окончательного расчета каждого клиента с фирмой.
3. Главную форму, содержащую название транспортного средства, и подчиненную, отражающую список всех заказов с ним.

Создать отчеты:

1. Окончательный расчет каждого клиента с фирмой.
2. О количестве заказанных транспортных средств каждого вида.
3. Список заказов, где транспортное средство используется более 10 часов (количество часов — параметр).

Создать кнопочную форму, содержащую четыре кнопки: две для открытия форм, третья — для открытия отчета о количестве заказанных транспортных средств каждого вида и четвертая — для выхода из приложения.

Приложение 2

Домашняя контрольная работа

Задание 1

В программе Microsoft Word составить и распечатать на одном листе формата А4 тезаурус из двадцати основных понятий СУБД Microsoft Access.

Задание 2

Выполнить проектирование базы данных с учетом поставленных запросов. База данных должна содержать:

1. Таблицы по заданной предметной области. Количество таблиц определяется в процессе проектирования, но не менее трех. Каждая таблица должна содержать восемь записей.
2. Десять запросов:
 - один на выборку к каждой таблице;
 - два перекрестных;
 - два на выборку к связанным таблицам;
 - два с параметрами к связанным таблицам;
 - один с вычисляемым полем;
 - два с групповыми операциями.

При оформлении формулировка запроса обязательна.

3. Формы:
 - главную и подчиненную;
 - одну автоформу.
4. Отчеты:
 - один автоотчет;
 - один, созданный Мастером отчетов или в режиме Конструктора.
5. Кнопочную форму.

В базе данных при создании объектов должны быть использованы объекты OLE (картинки, фотографии), возможности Мастеров, Конструктора и Построителя.

Требования к оформлению и расположению заданий домашней контрольной работы

- Титульный лист по стандарту — с. 1
- Оглавление — с. 2
- Задание 1 — с. 3
- Задание 2, формулировка — с. 4
- Таблицы, описание полей — с. 4
- Схема данных — с. 5
- Формулировка запросов, их вид в Конструкторе и результат выполнения — с. 6–9
- Формы — с. 10
- Отчеты — с. 11
- Литература — с. 12.

Выполнить автоматическую нумерацию страниц, не ставя номер на титульном листе.

Домашняя контрольная работа должна быть представлена в печатном и электронном варианте на диске.

- Вариант 1. Проведение семинара.
- Вариант 2. Поликлиника.
- Вариант 3. Гастроли.
- Вариант 4. Компьютерный салон.
- Вариант 5. Склад товаров магазина «Детский мир».
- Вариант 6. Склад продовольственных товаров.
- Вариант 7. Учет товаров оптовой базы промышленных товаров.
- Вариант 8. Продажа книг.
- Вариант 9. Начисление заработной платы.
- Вариант 10. Оплата штрафов ГИБДД.

Приложение 3

Использование подстановочных знаков для поиска значений

Подстановочные знаки используются в качестве прототипов для других символов при указании образца поиска в следующих случаях:

- известна только часть значения;
- требуется найти значения, начинающиеся с конкретной буквы или соответствующие определенному шаблону.

В базах данных Microsoft Access для поиска значений полей, записей или имен файлов в диалоговых окнах **Поиск** и **Замена** или в запросах, командах и выражениях допускается использование символов, указанных в таблице.

Символ	Использование	Примеры
*	Соответствует любой цифре или любому символу. Может использоваться в качестве первого или последнего символа текстовой строки	wh* — поиск слов what, white и why и т. п.; *ол — поиск слов вол, волейбол, пол и т. п.; *ло* — поиск слов горло, полотенце, Фролова и т. п.; 2* — поиск слов (чисел) 2 доа, 2-я задача (22, 209)
?	Соответствует любому текстовому символу	B?ll — поиск слов ball, bell и bill и т. п.; 2??7? — поиск слов (чисел) 2 ав7ы, 2 пр76 (24379, 22377) и т. п. 2?.?.7? — двадцатые числа 70-х годов (для дат в формате дд.мм.гг)
[]	Соответствует любому одному символу из заключенных в скобки	B [ae]ll — поиск слов ball и bell, но не bill; [12]5 [34] — 153, 253, 254, 253, но не 354, не 155; [М-Ф]* — строки, начинающиеся с буквы «М» по «Ф»; 12.0 [36].7 [13] — 12-е число июня или марта 1971 или 1973 года

Символ	Использование	Примеры
!	Соответствует любому одному символу, кроме заключенных в скобки	b [! ae]ll — поиск слов bill и bull, но не bell или ball; [! Оля]* — все строки, не начинающиеся с «Оля»; 4 [!8]? — все трехзначные числа, на первом месте которых цифра 4, на втором — любая цифра кроме 8, а на третьем — любая цифра (456, 421, но не 482)
-	Соответствует любому символу из диапазона. Необходимо указывать этот диапазон по возрастанию (от A до Z, но не от Z до A)	b [a-c]d — поиск слов bad, bbd и bcd; 7 [1-5] — числа от 71 до 75 включительно; [П-Т]* — строки, начинающиеся с букв «П» по «Т», например Петя, Толик, Татьяна, Т-34 и т. п.;
#	Соответствует любой цифре	1#3 — поиск значений 103, 113, 123

Функции даты и времени

Функция	Описание
Day (дата)	Возвращает значение дня месяца в диапазоне от 1 до 31
Year (дата)	Возвращает значение года в диапазоне от 100 до 9999
Month (дата)	Возвращает значение месяца года в диапазоне от 1 до 12
Weekday (дата)	По умолчанию возвращает целое число от 1 (Воскресенье) до 7 (Суббота), соответствующее дню недели
Hour (дата)	Возвращает целое число от 0 до 23, представляющее значение часа
Now (0)	Возвращает дату и время часов компьютера
Date ()	Возвращает текущую системную дату

Специальные операторы сравнения

Кроме обычных операторов сравнения Access предоставляет специальные операторы для отбора данных — операторы Between и Like.

Логический оператор BETWEEN

Оператор BETWEEN определяет принадлежность значения выражения указанному диапазону.

Синтаксис: выражение [Not] Between значение_1 And значение_2

Ниже перечислены аргументы оператора Between... And:

Элемент	Описание
---------	----------

Выражение	Выражение, определяющее поле, значение которого проверяется на принадлежность к диапазону
Значение_1, значение_2	Выражения, задающие границы диапазона

Дополнительные сведения

Если значение поля, определенное в аргументе выражения, попадает в диапазон, задаваемый аргументами значение_1 и значение_2 (включительно), оператор `Between... And` возвращает значение `True` (истина); в противном случае возвращается значение `False` (ложь). Логический оператор `Not` позволяет проверить противоположное условие (что выражение находится за пределами диапазона, заданного с помощью аргументов значение_1 и значение_2).

Оператор `Between... And` часто используют для проверки, попадает ли значение поля в указанный диапазон чисел.

В следующей таблице приведены примеры определения диапазонов значений с помощью оператора `Between... And`.

Выражение	Описание
<code>Between #02.02.93# And #01.12.93#</code>	Даты в диапазоне от 2-фев-93 до 1-дек-93. Означает то же самое, что и условие <code>>= #02.02.93# and <=#01.12.93#</code>
<code>Between 2 And 9</code>	Числа от 2 до 9. Означает то же самое, что и условие <code>>= 2 and <=9</code>
<code>Not Between 17 and 87</code>	Все числа кроме диапазона от 17 до 87. Означает то же самое, что и условия <code><17 and >87</code> или <code>Not (>=17 and <=87)</code>

Логический оператор LIKE

Оператор `Like` полезен при поиске образцов в текстовых полях. Используется для сравнения строкового выражения с образцом в выражении.

Синтаксис: выражение Like «образец»

Ниже перечислены аргументы оператора `Like`:

Элемент	Описание
Выражение	Выражение, используемое в условии на значение (<code>></code> , <code><</code> , <code>>=</code> , <code><=</code> , <code>Not</code> и т. д.)
Образец	Строка, с которой сравнивается выражение

Дополнительные сведения

Для аргумента образец можно задавать полное значение (например, `Like «Иванов»`) или использовать подстановочные знаки для по-

иска диапазона значений (например, Like «Ив*»).

Оператор Like используется в выражении для сравнения значений поля со строковым выражением. Например, если в условии на значение поля ввести Like «С*», фильтр или запрос возвратит все значения поля, начинающиеся с буквы «С».

Следующая таблица содержит примеры использования оператора Like для тестирования выражений с помощью разных образцов.

Тип совпадения	Образец	Совпадение (True)	Несовпадение (False)
Несколько символов	a*a	aa, aBa, aBBBa	aBC
ab	abc, AABb, Xab	aZb, bac	
Специальный символ	a [*]a	a*a	aaa
Несколько символов	ab*	abcdefg, abc	cab, aab
Одиночный символ	a?a	aaa, a3a, aBa	aBBBa
Одиночная цифра	a#a	a0a, a1a, a2a	aaa, a10a
Диапазон символов	[a-z]	f, p, j	2, &
Вне диапазона	[! a-z]	9, &, %	b, a
Не цифра	[!0-9]	A, a, &, ~	0, 1, 9
Комбинированное выражение	a [! b-m]#	An9, az0, a99	abc, aj0