Саратовский государственный университет имени Н.Г. Чернышевского

В.Ф. Кабанов, А.В. Бурмистров

Основные принципы работы с базами данных

Учебное пособие для студентов Саратовского государственного университета

Саратов

2015

УДК 004.41/.42 ББК 32.973 - 018.1

 \mathbf{C}

Кабанов В.Ф., Бурмистров А.В.

Основные принципы работы с базами данных: учебн. пособие для \mathbf{C} студ. Саратовского гос. ун-та. – Саратов, 2015. - 94 с.

ISBN

В пособии в доступной форме рассмотрены основные понятия информационных технологий, баз данных, систем управления базами Рассмотрены основные объекты Ms Access и приведены практические примеры применения СУБД Ms Access 2007.

Учебное пособие предназначено для студентов факультета нано- и биомедицинских технологий, а также для студентов физических и инженерных специальностей других факультетов и институтов Саратовского государственного университета.

Рекомендует к публикации

Кафедра физики полупроводников факультета нано- и биомедицинских технологий CARATOBOKNÍN FOCYTLAR Саратовского государственного университета

УДК 004.41/.42 ББК 32.973 - 018.1

ISBN

ОГЛАВЛЕНИЕ

введение	4
ГЛАВА 1. ОСНОВНЫЕ ПОНЯТИЯ БАЗ ДАННЫХ	5
1.1. Базы данных. Основные модели представления данных	5
1.2. Состав и основные функции систем управления базами данных	9
1.3. Проектирование баз данных	11
Вопросы для самоподготовки.	12
ГЛАВА 2. РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ MICROSOFT ACCESS 2007	13
2.1. Таблицы	13
2.2. Формы	26
2.3. Поиск в базе данных	36
2.4. Фильтры в Ms Access	38
2.4. Фильтры в Ms Access2.5. Элементы управления	40
2.6. Отчеты2.7. Связи и фильтры	48
2.7. Связи и фильтры	55
2.8. Подчиненные формы	63
2.9. Запросы	75
2.10. Гиперссылки	86
2.11. Кнопки	89
Список информационных источников	93

Введение

В настоящее время развитие информационных технологий определяется все возрастающей интенсивностью информационных потоков вследствие усиления процессов глобализации мировой экономики и становления информационного всемирного пространства благодаря развитию Интернет.

Наше обозримое будущее, с большой долей вероятности, будет определяться повышением сложности и ускорением темпов изменений. Факторы повышения сложности ведут к переходу на новую экономическую модель: от классической промышленной эпохи (при ее основе на стандартных промышленных процессах) к новой информационной эпохе (здесь определяющим фактором будет все возрастающая сложность процессов и резкое повышение роли знаний). Экономика, основанная на знаниях, поможет наиболее эффективно выстраивать бизнес в современных условиях. Такая экономика с необходимостью требует внедрения новейших информационных технологий.

Наиболее используемыми видами информационных технологий в настоящее время являются:

- работа с едиными базами данных;
- использование сетевых технологий;
- внедрение экспертных систем, что позволяет использовать, вместо экспертов и узких специалистов, специалистов широкого профиля, уменьшая численность работников и снижая вероятность ошибки, возникающей при взаимодействии людей между собой;
- внедрение систем принятия решений, что избавляет менеджеров от принятия тактических решений.

ГЛАВА 1. ОСНОВНЫЕ ПОНЯТИЯ БАЗ ДАННЫХ

1.1. Базы данных. Основные модели представления данных

Одной из наиболее распространенных форм реализации информационных технологий в настоящее время являются базы данных.

База данных (БД) представляет собой совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области.

Другими словами базу данных онжом определить совокупность как взаимосвязанных характеризующихся возможностью использования данных, большого количества приложений, возможностью быстрого получения и модификации необходимой информации, минимальной избыточностью информации, независимостью от прикладных программ, общим управляемым способом поиска.

Также важными понятиями информационных технологий при работе с базами данных являются:

- система управления базами данных (СУБД);
- словарь данных;
- администратор баз данных и др.

Система управления базами данных (СУБД) – это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД пользователями.

Словарь данных предназначен для централизованного хранения информации о структурах данных, взаимосвязях файлов БД друг с другом, типах данных и форматах их представления, кодах защиты, разграничения доступа и т.д.

Администратор БД – это лицо или группа лиц, отвечающих за выработку требований к БД, ее проектирование, создание, использование и сопровождение.

пользователя. Можно сформулировать основные *требования к базам данных* с точки зрения

- БД должна соответствовать актуальным информационным потребностям пользователя.
- БД должна обеспечивать получение требуемых данных за приемлемое время (т.е. отвечать заданным требованиям производительности).
- БД должна легко расширяться при реорганизации и расширении предметной области.

- БД должна легко изменяться при изменении программной и аппаратной среды.
- Помещенные в БД корректные данные должны оставаться корректными.
- Данные до включения в БД должны проверяться на достоверность.
- Доступ к данным, размещаемым в БД, должны иметь только лица с соответствующими полномочиями.

Моделью представления данных или моделью базы данных называют логическую структуру хранимых в базе данных. Структуры данных в существующих моделях БД обладают относительной устойчивостью. Минимальная избыточность и возможность быстрой модификации позволяют поддерживать данные на соответствующем уровне актуальности. Одно из основных свойств БД – независимость данных и использующих их программ (т.е. изменение данных не приводит к изменению программ и наоборот). Структура базы данных предполагает формирование логических записей, их элементов и взаимосвязей между ними. Принято выделять следующие типы взаимосвязей:

- один к одному (1:1) одна запись может быть связана с одной записью,
- один ко многим (1 : N) одна запись взаимосвязана со многими другими,
- **многие ко многим** (M: N) одна запись может входить в отношения со многими другими записями в различных вариантах.

Хранимые в базе данные имеют определенную логическую структуру (модель). В соответствии с применением определенного вида взаимосвязей выделяют следующие важнейшие модели данных:

- иерархическая;
- сетевая;
- реляционная;
- объектно-ориентированная.

Иерархическая модель данных строится по принципу иерархии типов объектов, т.е. один объект является главным, а остальные, находящиеся на низших уровнях иерархии, - подчиненными. Между главным и подчиненным объектами устанавливается взаимосвязь «один ко многим». Данная модель удобна для работы с иерархически упорядоченной информацией и громоздка для информации со сложными логическими связями.

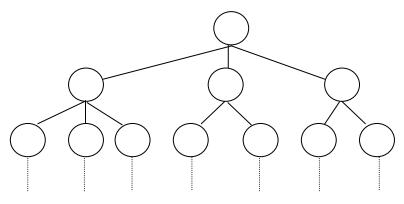


Рис.1.1. Схема иерархической модели данных

Достоинством иерархической модели является достаточно очевидное описание структуры данных на логическом и физическом уровне. *Недостаток* модели – жесткая структура взаимосвязей между элементами данных (потеря информационной гибкости).

В *сетевой модели* данных понятия главного и подчиненных объектов несколько расширены. Любой объект может быть и главным и подчиненным, то есть могут быть реализованы связи «многие ко многим».

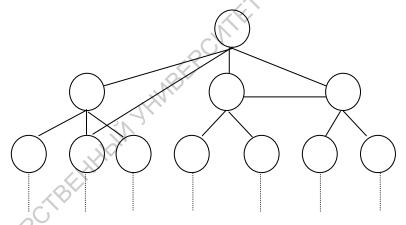


Рис.1.2. Схема сетевой модели данных

Достоинством сетевой модели данных является более высокая информационная гибкость по сравнению с иерархической моделью, эффективность затрат памяти и оперативность. *Недостаток* модели — достаточно сложное математическое описание системы и, как следствие, высокая сложность схемы базы данных, построенной на ее основе.

В *реляционной модели* данных объекты и взаимосвязи между данными представляются с помощью двумерных таблиц (рис. 1.3.). Каждая таблица представляет собой один объект и состоит из строк (записей) и столбцов (полей). Основное *достоинство* реляционной модели – простота, понятность для пользователя и удобство физической реализации на компьютере. Благодаря этому реляционная модель получила

наибольшее распространение в СУБД для персональных компьютеров. *Недостатком* модели является сложность описания иерархических и сетевых связей.

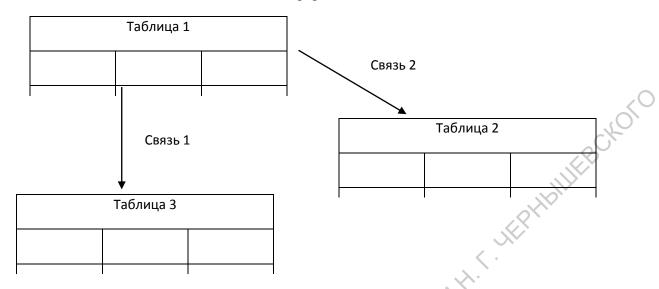


Рис. 1.3. Схема реляционной модели данных

Объектно - ориентированная модель в определенном смысле объединяет в себе две модели данных: реляционную и сетевую, и используется для создания крупных баз данных со сложными структурами данных.

Объектные базы данных строятся на основе объектно-ориентированного программирования, где акцент сделан не на программные структуры, а на объекты. Под объектом понимается достаточно крупный блок функционально взаимосвязанных данных. Типы и структуры данных, из которых состоит объект, могут быть различными у разных объектов и создаваться программистом на основе стандартных типов данных выбранного языка программирования. Объект можно определить как программно-связанный набор методов (функций) и свойств, выполняющих одну функциональную задачу. (Примерами использования объектных баз данных являются программные продукты 1С).

Важнейшими понятиями, связанными с объектом являются свойство, событие и метод.

Свойство — это характеристика, с помощью которой описываются внешний вид и работа объекта.

Событие — это действие, которое связано с объектом. Событие может быть инициировано пользователем, прикладной программой или операционной системой.

Метод – это функция или процедура, управляющая работой объекта при его реакции на событие.

Для выполнения действий над данными в рассматриваемой модели БД применяются логические операции с использованием объектно-ориентированных механизмов инкапсуляции, наследования и полиморфизма.

- *Инкапсуляция* ограничивает область видимости имени свойства пределами того объекта, в котором оно определено. Смысл такого свойства будет определяться тем объектом, в который оно инкапсулировано.
- *Наследование* распространяет область видимости свойства на всех потомков объекта.
- Полиморфизм в объектно-ориентированных языках программирования означает способность одного и того же программного кода работать с разнотипными данными. Другими словами, он означает допустимость в объектах разных типов иметь методы (процедуры или функции) с одинаковыми именами.

Создание объектов – достаточно трудоемкая задача для программистов. В настоящее время объектные модели данных не имеют строгой теоретической основы, что затрудняет их создание и использование. Однако увеличение возможностей ПК, развитие средств мультимедиа и компьютерных сетей предполагает реализацию надежных систем объектных баз данных.

Основным *достоинством* объектно-ориентированной модели данных является возможность отображения информации о сложных связях объектов. Объектно-ориентированная модель данных позволяет идентифицировать отдельную запись базы данных и определять функции их обработки.

Недостатками объектно-ориентированной модели являются высокая понятийная сложность, неудобство обработки данных и низкая скорость выполнения запросов.

1.2. Состав и основные функции систем управления базами данных

Для работы с базами данных используются системы управления базами данных (СУБД).

Современная СУБД включает в свой состав:

- программные средства создания баз данных,
- средства для работы с данными,
- сервисные средства.

С помощью средств создания БД проектировщик (используя язык описания данных) переводит логическую модель БД (то есть некоторое формализованное отображение структуры данных предметной области) в физическую структуру; разрабатывает программы, реализующие основные операции с данными (применяя язык

манипулирования данными). При проектировании используются визуальные средства и программа – отладчик, с помощью которой соединяются и тестируются отдельные блоки программы управления конкретной базы данных.

Средства работы с данными (предназначены для пользователя) позволяют установить удобный интерфейс с пользователем, создать необходимую функциональную конфигурацию экранного представления выводимой и вводимой информации, производить операции с данными, текстовыми и графическими экранными объектами.

Сервисные (дополнительные) средства позволяют при проектировании и использовании БД привлечь другие системы (например, использование программ офиса или сетевых серверных ресурсов).

К основным функциям СУБД с точки зрения пользователя можно отнести:

- Определение данных, т.е. описание таблиц базы данных. Прежде чем заносить данные в таблицы, необходимо определить структуру (поля) этих таблиц: название поля, его тип.
- Обработка данных. Любая СУБД позволяет выполнять четыре простейшие операции с данными:
 - добавить в таблицу одну или несколько записей;
 - удалить из таблицы одну или несколько записей;
 - обновить значение полей в одной или нескольких записях;
 - найти одну или несколько записей, удовлетворяющих заданному условию.
- Управление данными. Под управлением данными обычно понимают защиту данных от несанкционированного доступа, поддержку многопользовательского режима, работу с данными и обеспечение целостности и согласованности данных.

Перечисленные выше функции СУБД используют следующие основные функции более низкого уровня:

- управление данными во внешней памяти;
- управление буферами оперативной памяти;
- управление транзакциями;
- ведение журнала изменений в БД;
- обеспечение целостности и безопасности БД.

СУБД различаются по моделям БД, с которыми они работают. Например, если модель реляционная, то и используемая СУБД реляционная. Наиболее популярными персональными (или настольными) реляционными СУБД (то есть обеспечивающих

создание персональных БД и приложений, работающих с ними) в настоящее время являются: **Paradox**, различные версии **dBASE**, **Ms Access**.

1.3. Проектирование баз данных

Проектирование БД осуществляется путем разработки моделей данных на логическом и физическом уровнях представления:

- концептуальная модель интегрированное представление данных предметной области на логическом уровне, выраженное средствами структур данных выбранной СУБД;
- внутренняя модель отображение концептуальной модели на физическую структуру хранения БД на электронных носителях для выбранной СУБД;
- внешние модели варианты (подмножества) концептуальной модели данных, обеспечивающие санкционированный доступ к БД различных приложений.

Можно определить следующие основные этапы проектирования и создания базы данных.

- 1. Построение модели данных предметной области (на этом этапе документы предметной области отображаются в информационно-логической модели).
- 2. Определение структуры реляционной базы данных (на основании анализа информационно-логической модели реализуется проект структуры базы данных).
- 3. Конструирование таблиц базы данных в выбранной реляционной СУБД (проект структуры БД формализуется в пустые таблицы выбранной реляционной СУБД).
- 4. Создание ехемы данных в выбранной реляционной СУБД (создание схемы данных).
- 5. Ввод данных в таблицы или формы (создание записей).

Рассмотрим разработку *схемы данных* на примере одной из наиболее доступных реляционных СУБД Ms Access, входящих в состав Ms Office. Основные этапы можно сформулировать следующим образом.

- 1. Составление базового списка полей, который может насчитывать десятки и сотни позиций.
- 2. Определение наиболее подходящего типа данных для каждого поля.

- 3. Распределение полей базового списка по базовым таблицам. Критерием необходимости деления таблиц является наличие множественного повтора данных в соседних записях.
- 4. Выбор ключевого поля в каждой из таблиц (т.е. поля, данные в котором являются уникальными). При отсутствии в списке такого поля можно ввести дополнительное поле (типа *Счетиик*).
- 5. Определение связей между таблицами. Связь между двумя таблицами организуется на основе общего поля, которое должно присутствовать как в той, так и в другой таблице, причем в одной из таблиц оно обязательно должно быть ключевым.

Вопросы для самоподготовки

- 1. Сформулируйте основные требования к базам данных.
- 2. В чем достоинства и недостатки иерархической модели данных?
- 3. Каковы основные особенности реляционной модели данных?
- 4. Определите основные понятия, используемые в объектно-ориентированной модели данных.
- 5. Что относится к основным функциям реляционной системы управления данными?
- 6. Определите основные этапы проектирования и создания базы данных.

ГЛАВА 2. РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ MICROSOFT ACCESS 2007

Важнейшим достоинством СУБД **Ms Access** является то, что она имеет простой графический интерфейс, который позволяет пользователю создавать собственные базы данных и разрабатывать приложения различной степени сложности.

Основными объектами **Ms Access** являются *таблицы, формы, запросы, отчеты.* Создавать объекты можно в режимах соответствующих мастеров и конструктора; модифицировать – только в режиме конструктора.

2.1. Таблицы

2.1.1. Создание базы данных

Заранее создадим папку, с произвольным названием (например, **Бутик**), в которой вы впоследствии сохраните файл базы данных.

Базой данных Microsoft Access является файл, который имеет расширение mdb.

Запустить программу Microsoft Access. В появившемся окне нажать кнопку Новая база данных.

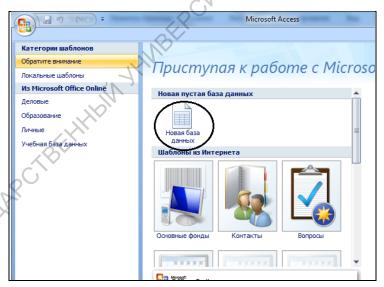
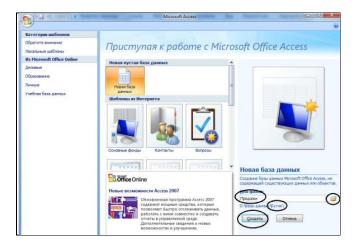


Рис.2.1. Окно программы **Microsoft Access** после запуска

Далее следует ввести имя (например, **Продажи**) и выбрать папку (**Бутик**) для сохранения базы данных.



BIIIE BOKOFO

Рис.2.2. Сохранение новой базы данных

В результате окно базы данных будет выглядеть следующим образом.

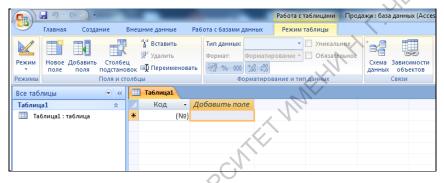


Рис.2.3. Окно базы данных Продажи

2.1.2. Окно базы данных

Особым окном в Microsoft Access является окно базы данных. В левой части окна находится панель объектов: Таблицы, Запросы, Формы, Отчеты и т.д. При создании новой базы данных присутствует только объект Таблица.

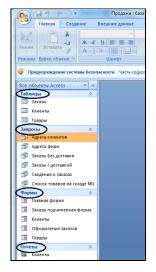


Рис.2.4. Панель объектов

При щелчке мышью по ярлыку отображается список объектов: таблиц, запросов, форм, отчетов.

Объект **Таблица** предназначен для хранения данных в виде записей (строк), состоящих из полей (столбцов).

Объект Форма предназначен для более удобного ввода данных.

Объект Запрос позволяет получить нужные данные из одной или нескольких таблиц.

Объект Отчет предназначен для печати данных.

С каждым объектом из списка можно работать в двух режимах.

Для таблиц, запросов, форм первый режим означает открытие объекта и называется, соответственно, режим **Таблицы**, режим **Формы**, режим **Таблицы** для запроса. Для отчета – это режим предварительного **Просмотра**.

Второй режим – режим **Конструктора**. Этот режим применим ко всем типам объектов и предназначен для создания и изменения объекта.

Созданные объекты можно переименовывать, копировать, удалять.

Чтобы удалить объект из списка, необходимо его выделить и нажать клавишу **Delete**. В появившемся диалоговом окне, запрашивающем подтверждение на удаление, нажать кнопку **OK**.

Чтобы скопировать объект из списка, необходимо его выделить и нажать правую кнопку, выбрать из контекстного меню команду **Копировать**. Затем выполнить команду **Вставить**. В появившемся диалоговом окне ввести имя для скопированного объекта, нажать кнопку **ОК**.

Чтобы переименовать объект, нужно выделить объект, нажать правую кнопку и выбрать из контекстного меню команду **Переименовать**. Затем ввести новое имя и нажать клавишу **Enter**.

2.1.3. Создание таблицы с помощью шаблона

При создании таблицы с помощью **шаблона** предлагается образцы таблиц, предназначенных для использования в личных или деловых целях, каждая из которых содержит соответствующий набор полей. Остается выбрать образцы полей, после чего **Мастер шаблонов** автоматически создает таблицу.

Рассмотрим создание с помощью шаблона таблицы Клиенты.

1. В окне база данных **Продажи** в главном меню выберем команду **Создание**, а затем раскроем список **Шаблоны таблиц** команду **Контакты**.

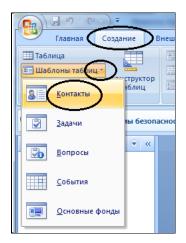


Рис.2.5. Создание таблицы с помощью Мастера

2. Появляется таблица со списком полей.

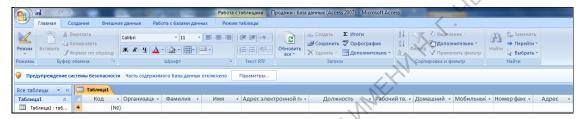


Рис.2.6. Окно Новая таблица

3. Далее нужно определить перечень полей, которые нужно включить в таблицу. Для этого поочередно выделим мышью имя поля в списке образцов полей и нажмем правую кнопку мыши удалить ненужные поля таблицы.

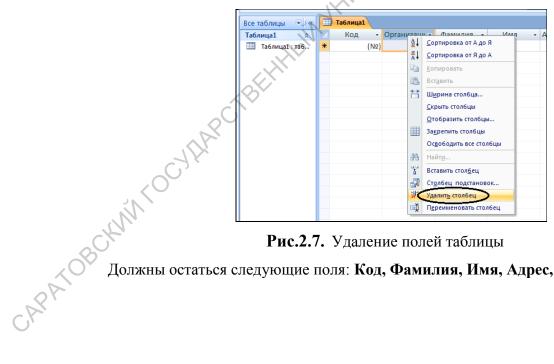


Рис.2.7. Удаление полей таблицы

Должны остаться следующие поля: Код, Фамилия, Имя, Адрес, Рабочий телефон.

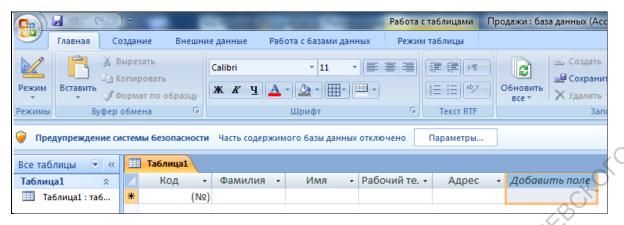


Рис.2.8. Поля таблицы

При необходимости можно переименовать поле **Фамилия** в поле **Фамилия**(**Название**), т.к. в это поле предполагается вводить или фамилию клиента, если клиент физическое лицо, или – название фирмы (юридического лица).

Чтобы переименовать поле, нужно его выделить и выбрать команду **Переименовать столбец**, в появившемся диалоговом окне ввести новое имя.

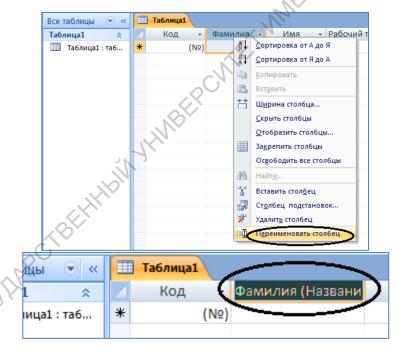


Рис.2.9. Переименование поля

4. Сохраним таблицу под именем **Клиенты**, нажав правую кнопку мыши на имени **Таблица 1**.

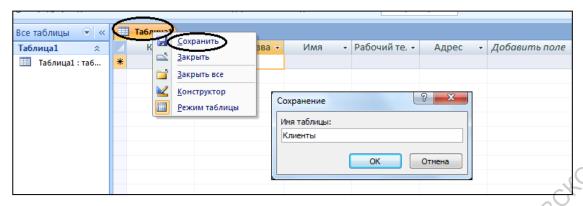


Рис.2.10. Сохранение таблицы

Містоsoft Access самостоятельно определит ключ (т. е. в этом окне не придется ничего вводить). **Первичный ключ** (в последующем изложении будем называть его просто **ключом**) — одно или несколько полей, совокупность значений которых однозначно определяет любую запись таблицы.

В нашем варианте ключевым станет поле Код.

Ответить Нет.

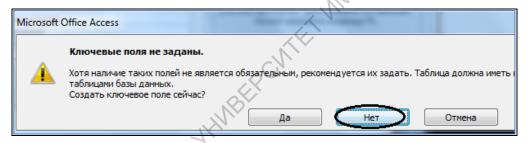


Рис.2.11. Автоматическое определение ключа таблицы

Таблица создана. В строке заголовков таблицы можно найти ее название **Клиенты**. Все имена полей представлены в виде заголовков столбцов таблицы.

2.1.4. Заполнение таблицы

В процессе создания таблицы возникает понятие ключа.

В таблице видно, что поле **Код** - счетчик, т. е. нумеруются записи в порядке ввода. Заполнять поле **Код** не нужно, счетчик срабатывает автоматически, как только заполнена хоть одна ячейка строки таблицы. Заполним поля **Фамилия** (**Название**), **Имя**, **Адрес**. Создадим пять записей (пять строк таблицы). Оставим незаполненным поле **Рабочий** телефон.

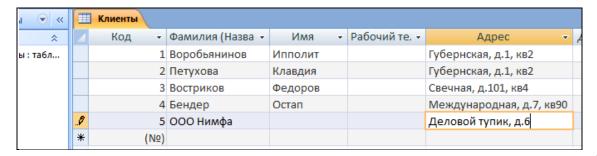


Рис.2.12. Заполненная таблица Клиенты

2.1.5. Изменение ширины столбца таблицы

Если ширина столбцов не соответствует размеру введенных в них записей (для нумерации место слишком много, а названия не просматриваются целиком).

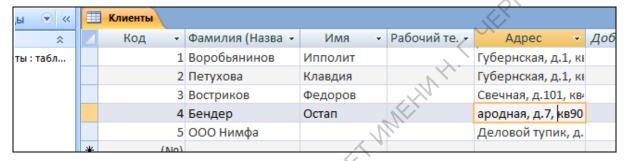


Рис.2.13. Ширина столбцов не соответствует размеру данных

Исправить положение можно:

- если подвести указатель мыши к правой границе столбца в строке заголовков и, удерживая нажатой левую клавишу мыши, переместить границу столбца;
- если подвести указатель мыши к правой границе столбца в строке заголовков и дважды щелкнуть мышью. В этом случае ширина столбца подгоняется под самую длинную запись столбца (учитывая заголовок);

Выполните необходимые действия и сравните результат с рисунком.

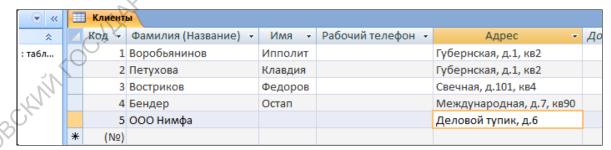


Рис.2.14. Изменение ширины столбцов

В таблице существует возможность переставлять столбцы. Сделать это можно, выделить столбец и, "схватив" столбец за заголовок мышью, перенести его на новое место.

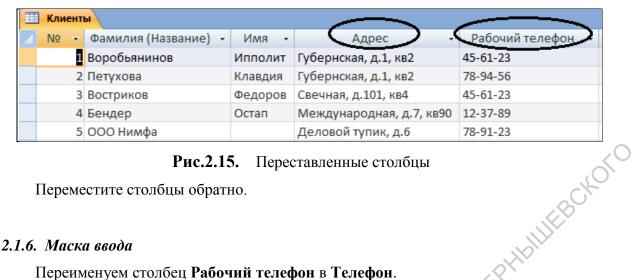


Рис.2.15. Переставленные столбцы

Переместите столбцы обратно.

2.1.6. Маска ввода

Переименуем столбец Рабочий телефон в Телефон.

Для удобства заполнения поля Телефон можно для этого столбца установить Маску ввода.

Маска ввода - это шаблон, позволяющий вводить в поле значения, имеющие одинаковый формат. Маска ввода автоматически изображает в поле постоянные символы. При вводе данных в поле нет необходимости набирать эти постоянные символы, даже если они должны быть включены в значение поля; достаточно заполнить пустые позиции в маске ввода. Кроме того, Microsoft Access не позволит ввести в поле значения, не вписывающиеся в определенную для этого поля маску ввода.

Любая таблица Microsoft Access может быть представлена в двух режимах:

1. в режиме Таблицы, предназначенном для ввода данных, их просмотра и редактирования;

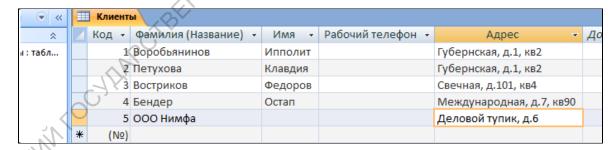


Рис.2.16. Режим Таблипа

в режиме Конструктора, предназначенном для создания и изменения таблицы.

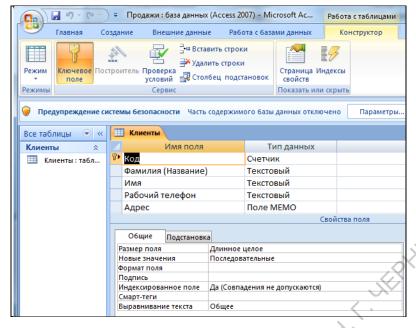


Рис.2.17. Режим Конструктор

Для того чтобы изменить маску ввода для телефона, следует переключиться в режим **Конструктора** таблицы. Это можно сделать, выбрав команду **Конструктор** меню **Вид,**

или при помощи кнопки панели инструментов

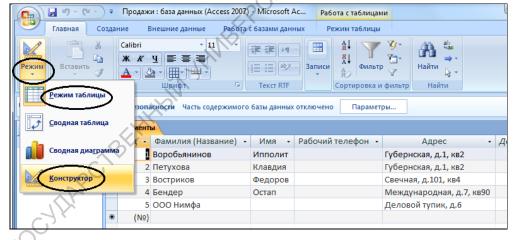


Рис.2.18. Кнопка Конструктор

Окно **Конструктора** состоит из двух частей. Верхняя часть содержит три столбца: **Имя поля, Тип данных, Описание**. В столбце имя поля содержатся имена полей таблицы, в столбце **Тип данных** указывается тип данных для каждого поля. Заполнение первых двух столбцов является обязательным. Столбец **Описание** заполнять не обязательно.

В нижней части располагаются две вкладки свойств полей: Общие и Подстановка.

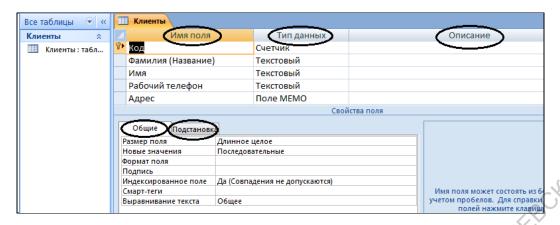
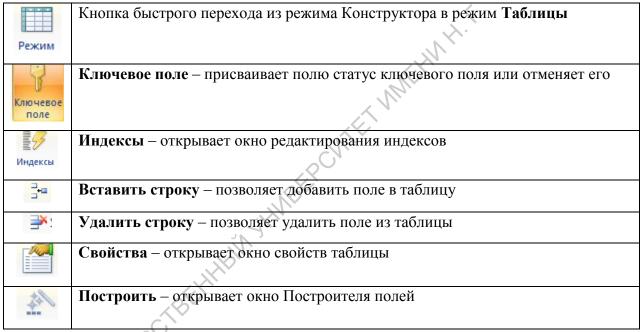


Рис.2.19. Окно Конструктора

В верхней части окна Конструктора находится панель инструментов. Рассмотрим основные кнопки этой панели инструментов.



Окно **Конструктора** содержит таблицу, в первом столбце которой перечислены поля, а во втором - типы данных (форматы).

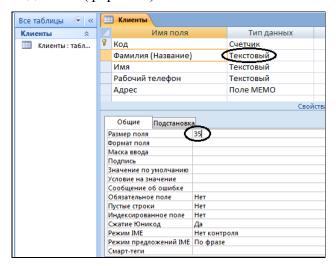


Рис.2.20. Окно Конструктора таблицы Клиенты

B Microsoft Access возможны следующие типы данных:

Текстовый — символьные или числовые данные, не требующие вычислений. Поле данного типа может содержать до 255 символов. На вкладке **Общие** в нижней части окна **Конструктора** находится свойство **Размер поля**. В нем можно указать максимальное количество символов, которые могут быть введены в данное поле. Например, поле **Фамилия** (**Название**) максимально может содержать 35 символов.

Поле МЕМО — предназначено для ввода текстовой информации, по объему превышающей 255 символов. Это поле может содержать до 65535 символов. В дальнейшем в таблице **Товары** имеет смысл полю **Примечание** присвоит тип **МЕМО**.

Числовой — используется для хранения числовых данных, используемых в математических расчетах. Например, числовым должно быть поле **Количество** в таблице **Заказы**.

Дата/Время — тип для предоставления даты и времени. В свойствах поля типа Дата/Время можно выбрать формат предоставления этого типа поля.

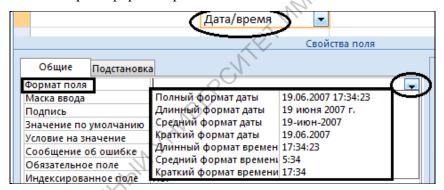


Рис.2.21. Выбор формата представления поля типа Дата/Время

Денежный – предназначен для хранения денежных данных. Поле **Оклад** в таблице **Сотрудники,** поле **Цена** в таблице **Товары** могут быть типа **Денежный**.

Счетчик – содержит уникальный номер, определяемый Microsoft Access для каждой новой записи автоматически путем увеличения предыдущего значения на 1. Редактирование значений этого поле невозможно. Чаще всего ключевые поля имеют тип Счетчик.

Логический — этот тип поля может содержать только два значения Да/Нет. Microsoft Access использует 1 для предоставления значения Да, 0 — для Нет. Поле Включать в прайс-лист таблицы Товары может быть логическим.

Мастер подстановок — выбор этого поля запускает **Мастера подстановок**, создающего поле подстановок. Поле подстановок позволит выбрать нужное значение из фиксированного списка или из поля другой таблицы. Например, поле **Образование** в

таблице Сотрудники может быть фиксированным списком: среднее, среднее-специальное, высшее. Поле Пол в таблице Сотрудники также может быть фиксированным.

Рассмотрим некоторые свойства полей.

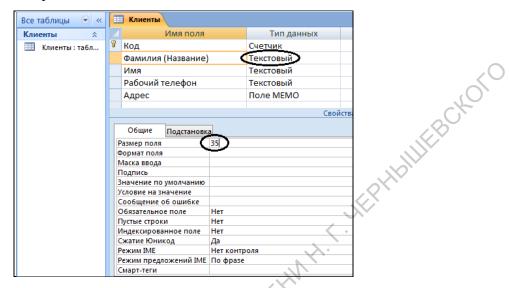


Рис.2.22. Свойства поля Конструктора таблиц

Подпись – с помощью этого свойства можно задать названия полей таблицы. Это название может отличаться от имени поля в режиме **Таблица**. Например, имя поля **ФИО**, а подпись для него **Фамилия**, **Имя**, **Отчество**.

Обязательное поле – определяет, является ли ввод данных в это поле обязательным.

Формат поля – указывает формат ввода данных в режиме Таблица. Для определения формата текстового типа используются специальные символы форматирования. числовых полей Для значение формата выбрать онжом раскрывающегося списка.

Маска ввода - иногда может понадобиться, чтобы при вводе информации некоторые символы всегда присутствовали на экране. Например, номер телефона (845)-14-56-18 содержит скобки для указания кода города и дефисы в самом номере. С помощью Маски ввода можно автоматизировать ввод номеров телефона.

\ - указывает, что следующий символ следует рассматривать в качестве постоянного символа.

")-" вместо многократного ввода \)\- можно заключить любые символы в двойные кавычки.

Размер поля - Указывает максимальное количество вводимых символов в данное поле.

Условие на значение – определяет ограничение на вводимые в это поле символы. При несоответствии вводимых данных указанному условию выдается сообщение об ошибке.

Сообщение об ошибке — это свойство определяет то сообщение, которое будет выдаваться пользователю, если не выполняется условие на значение.

Выделим поле Телефон (достаточно установить текстовый курсор в соответствующую ячейку таблицы). В нижней части окна изменится перечень свойств поля.

Установим **Размер поля** – 30, **Подпись** – Телефон клиента, **Маска ввода** - шаблон **99-99-99**, позволяющий вводить в качестве номера телефона шестизначное число, причем разделительные тире будут вставляться автоматически.

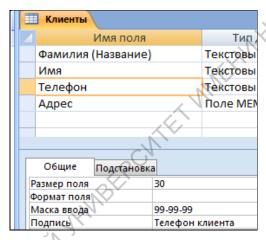


Рис.2.23. Свойство Маска ввода

Перейдем из режима Конструктора в режим формы. Сделать это можно, нажав

кнопку на ленте

На вопрос о сохранении таблицы, отвечаем Да.

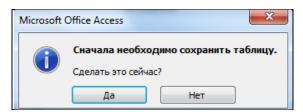


Рис.2.24. Окно сообщения Сохранение таблицы

При появлении сообщения о потери данных ответить Да.

Далее заполним столбец **Телефон**. Нужно обратить внимание на то, чтобы перед началом ввода текстовый курсор находился в самом конце ячейки. В данном случае перемещаться между ячейками удобнее при помощи кнопок управления курсором.

Ш Клиенты						
Код ▾	Фамилия (Название) 🔻	F RMN	Телефон клиента →	Адрес -		
1	Воробьянинов	Ипполит	45-61-23	Губернская, д.1, кв2		
2	Петухова	Клавдия	78-94-56	Губернская, д.1, кв2		
3	Востриков	Федоров	45-61-23	Свечная, д.101, кв4		
4	Бендер	Остап	12-37-89	Международная, д.7, кв90		
5	ООО Нимфа		78-91-23	Деловой тупик, д.6		

Рис.2.25. Таблица Клиенты с номерами телефонов

Задания для тренировки

1. Самостоятельно измените высоту строки (по аналогии с такой же возможностью в электронных таблицах).

Можно ли изменить высоту только одной строки (как это делали для столбца)? Как вы думаете почему?

2. Перейдите в режим **Конструктора**. Самостоятельно измените подпись (в свойствах поля) к полю **Код клиента** на №.

Перейдите в режим **Таблицы** и измените ширину столбца № в соответствии с размером нового заголовка.

Ш Клиенты						
1	N⊈ ▼	Фамилия (Название) 🕶	- RWN	Телефон клиента	¥	Адрес -
	1	Воробьянинов	Ипполит	45-61-23		Губернская, д.1, кв2
	2	Петухова	Клавдия	78-94-56		Губернская, д.1, кв2
	3	Востриков	Федоров	45-61-23		Свечная, д.101, кв4
	4	Бендер	Остап	12-37-89		Международная, д.7, кв90
	5	ООО Нимфа		78-91-23		Деловой тупик, д.6

Рис.2.26. Окончательный вид таблицы

Закройте таблицу, выполнив сохранение.

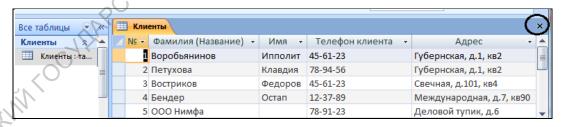


Рис.2.27. Закрытие таблицы

2.2. Формы

2.1.7. Создание формы

Microsoft Access позволяет организовать удобный и понятный интерфейс для работы с данными с помощью форм. **Формами** называются окна, сохраняемые в базе данных в

виде объектов специального вида. Формы являются такими же объектами базы данных, как и таблицы, отчеты и запросы. Формы используются для ввода и отображения данных.

Формы - более удобный способ для просмотра и правки данных в таблице, чем режим **Таблицы**.

Откроем созданную ранее базу данных **Продажи.** Имя файла можно найти в списке недавно использовавшихся файлов.

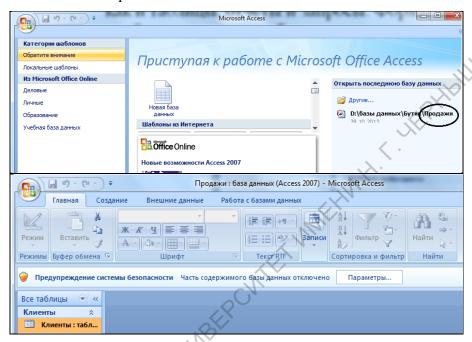


Рис.2.28. Открытие файла и окно базы данных Продажи

В некоторых случаях удобнее вводить данные не в таблицу, а в форму, в которой каждую запись можно представить в отдельном окне.

Для создания формы для таблицы Клиенты выполним следующие действия:

1. в окне база данных Продажи на панели объектов выберем объект Формы и способ Мастер форм.

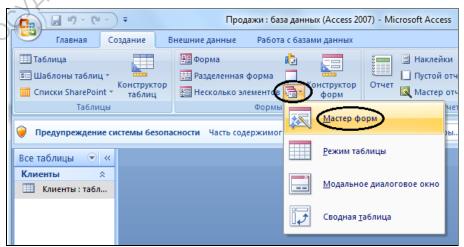


Рис.2.29. Создание формы с помощью Мастера

в окне диалога нужно выбрать для формы все, имеющиеся в таблице поля. Для 2. этого достаточно нажать \geq (кнопка \geq добавляет только выделенные поля);

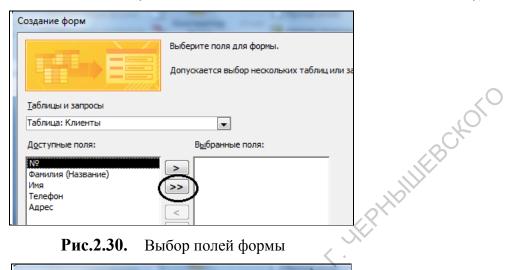


Рис.2.30. Выбор полей формы

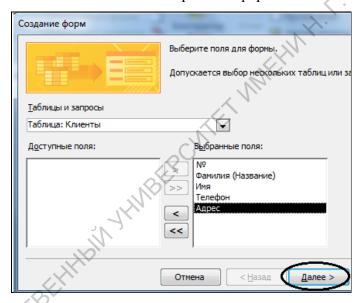


Рис.2.31. Выбранные поля из таблицы Клиенты

- нажать кнопку Далее; 3.
- 4. в окне диалога выбрать команду В один столбец и нажать кнопку Далее; в следующем окне диалога нужно выбрать стиль и нажать кнопку Далее;

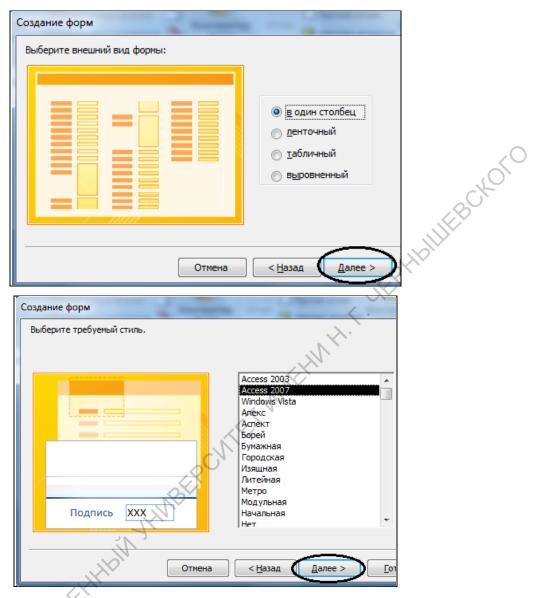


Рис.2.32. Выбор внешнего вида формы и выбор стиля формы

5. нужно ввести название формы или согласиться с предложенным **Клиенты** и нажать кнопку **Готово**.

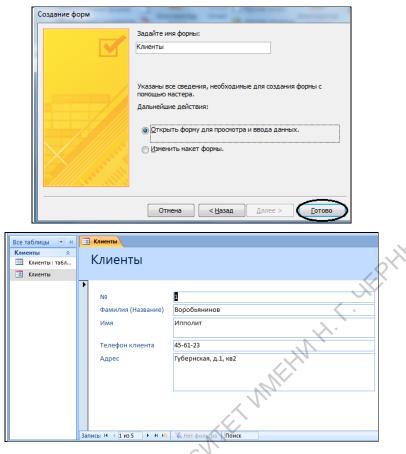


Рис.2.33. Ввод имени формы и резултат создания таблицы

Каждая запись (то, что являлось отдельной строкой таблицы) теперь представляет собой как бы отдельную карточку. Вводить данные удобнее именно в таком режиме.

В строке состояния (в нижней части окна формы) отражено общее число записей и номер записи.



Рис.2.34. Кнопки перехода между записями

Перемещаться между записями можно при помощи кнопок:

- на одну запись вперед,
- на одну запись назад,
- к последней записи.

Те же результаты можно получить, выполнив команду **Перейти** меню **Записи**, выбрав одну из команд подменю:

Первая - переводит курсор из поля в текущей записи в то же поле в первой записи.

Последняя - переводит курсор из поля в текущей записи в то же поле в последней записи.

Следующая - переводит курсор из поля в текущей записи в то же поле в следующей записи.

Предыдущая - переводит курсор из поля в текущей записи в то же поле в предыдущей записи.

Новая 💾 - переводит курсор на пустую запись таблицы, позволяя ввести новую ... тренировки
Пролистайте все имеющиеся записи (просмотрите все карточки).
Перейдите к первой записи.
Перейдите к последней записи
Внеситс запись.

Задание для тренировки

- Внесите исправления в некоторые записи.

2.1.8. Заполнение формы

Необходимо обратиться к самой последней записи и нажать кнопку "Вперед на одну запись". Откроется первая пустая карточка.

Установим курсор в поле Фамилия (Название) и введем нового клиента. Заполним всю карточку. Перемещаться между полями можно при помощи клавиши **Tab** или при помощи клавиш управления курсором. Перейти к новой записи (карточке) можно нажав клавишу Enter.

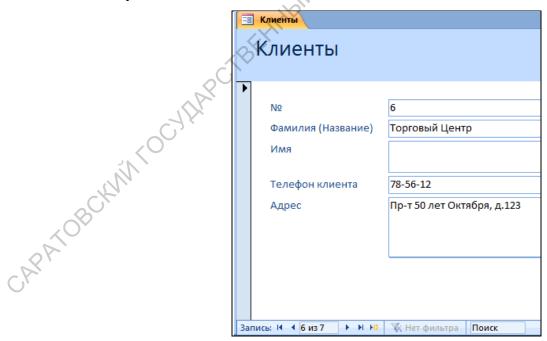


Рис.2.35. Ввод новых записей Торговый центр

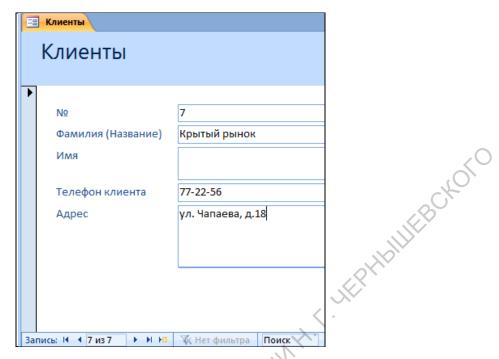


Рис.2.36. Ввод новых записей в таблицу Клиенты

Для проверки отображения записей в исходной таблице нужно:

- 1. выбрать на панели объектов окна базы данных объект Таблица Клиены;
- 2. открыть таблицу Клиенты;



Рис.2.37. Таблица Клиенты

Закроем таблицу Клиенты.

2.1.9. Представление формы в режиме формы и режиме Конструктор

Работа с формами может происходить в трех режимах: в режиме **Формы**, в режиме **Таблицы**, в режиме **Конструктора**. Выбрать режим работы можно с помощью кнопки **Вид** на Ленте.

Представим форму в Режиме таблицы.

Представим форму в режиме Конструктор.

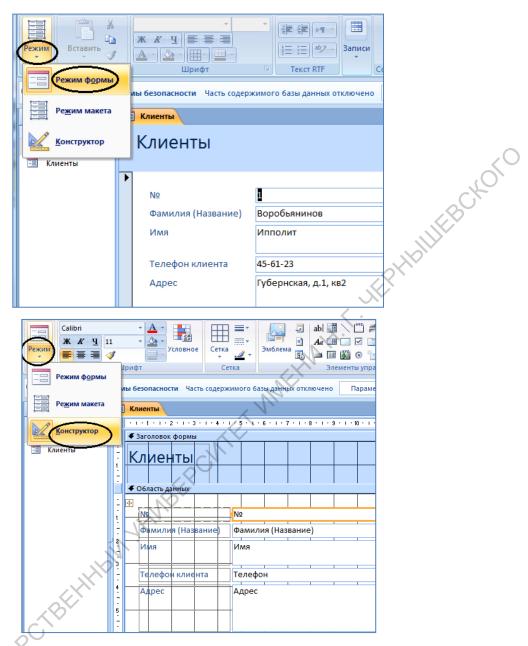


Рис.2.38. Кнопка Вид на Ленте

В режиме формы можно удалять записи, если выделить строку таблицы и нажать клавишу **Delete**.

В верхней части главного окна Microsoft Access имеется панель инструментов **Режима формы**. На ней расположены следующие кнопки:

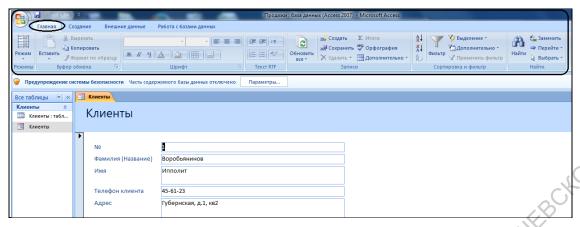


Рис.2.39. Панель инструментов формы

-		
	Режим	Ви д – позволяет быстро переключать режим формы и режим Конструктора
		Сохранить – позволяет сохранить базу данных на диске
	AEC	Орфография – запускает процедуру проверки орфографии в выделенном поле формы
	A↓	Сортировка по возрастанию – сортировка таблицы по возрастанию текущего поля
	A↓	Сортировка по убыванию – сортировка таблицы по убыванию текущего поля
	1	Фильтр по выделенному
	Y 4	Изменить фильтр
	Y	Применить/Отменить фильтр
		Найти – вызывает процедуру поиск и замены
		Новая запись – позволяет добавить запись в таблицу
12	×	Удалить запись — удаляет текущую запись
Для	сортировкі	и записей в таблице необходимо:
- CP	установ	ить курсор в поле, по которому будет проводиться сортировка
САРРАТОВ СТДЛЯ	нажать	одну из кнопок на панели инструментов . По возраста

- установить курсор в поле, по которому будет проводиться сортировка;
- нажать одну из кнопок на панели инструментов _____ По возрастанию или - По убыванию.

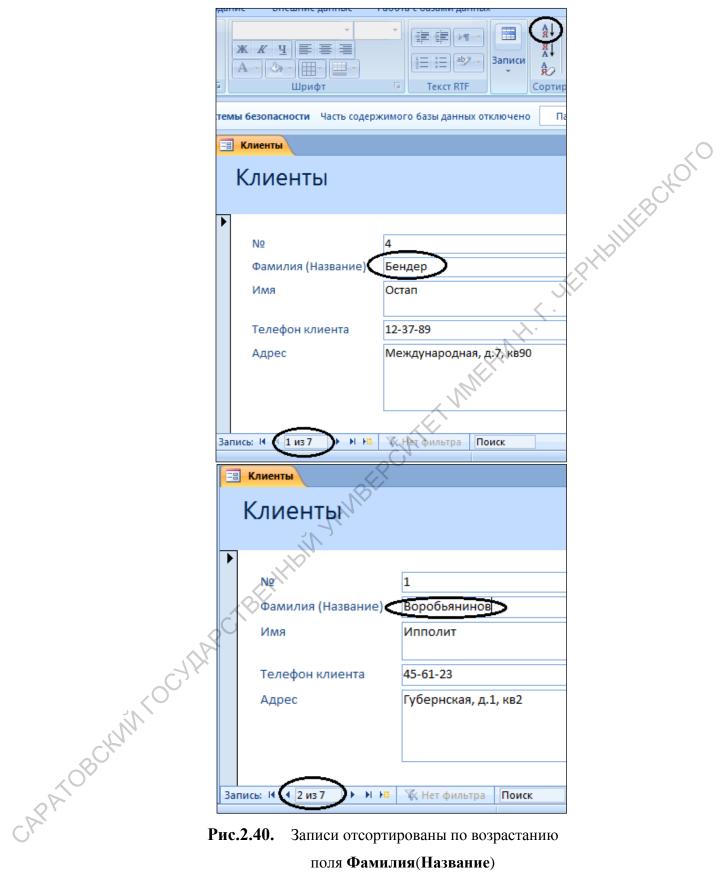


Рис.2.40. Записи отсортированы по возрастанию поля Фамилия(Название)

Задания для тренировки

1. Отсортируйте записи в таблице таким образом, чтобы фамилии располагались по алфавиту.

- 2. Отсортируйте записи таким образом, чтобы имена располагались по алфавиту.
- 3. Отсортируйте записи таким образом, чтобы номера располагались по убыванию.

2.3. Поиск в базе данных

Предположим, в базе данных накопилось довольно много клиентов и понадобилось найти одного из них. Точно известно только имя клиента, остальная информация довольно расплывчата.

Средствами Microsoft Access можно организовать эффективный поиск нужных данных.

- 1. Установить курсор в поле Имя.
- 2. Выполнить команду Найти... панели инструментов.

Если открывшееся окно диалога **Поиск** загораживает форму, обязательно "отодвиньте" его, чтобы можно было прочитать данные формы.

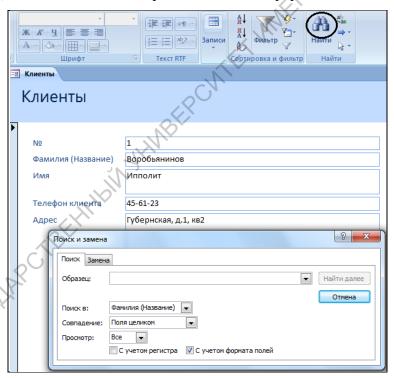


Рис.2.41. Окно диалога Поиск и замена

В раскрывающемся списке Совпадение можно выбрать одну из трех возможностей:

а) **с** любой частью поля - совпадения с образцом ищутся в любой части содержимого поля. Например, для образца "**Ира**" будут найдены "**Кира**" и "**Ираида**".

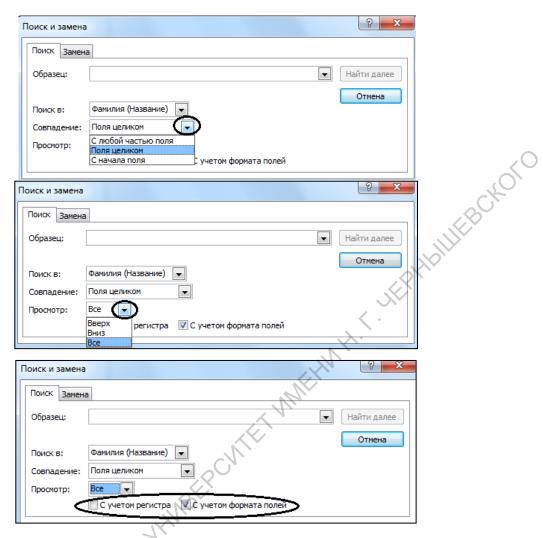


Рис.2.42. Варианты поиска

- b) **поля целиком** будут обнаружены поля, содержимое которых полностью совпадает с образцом.
- с) с начала поля ищутся совпадения с образцом с начала содержимого поля. Например, для образца "Ира" будет найдена "Ираида", но не "Кира". В раскрывающемся списке Просмотр существует возможность определить направление поиска:
- вверх от текущей записи к первой;
- вниз от текущей записи к последней;
- все будут просмотрены все поля, независимо от положения курсора.

При необходимости организовать поиск с учетом регистра символов нужно установить соответствующий флажок. В этом случае, при введенном образце "**Ира**" не будут обнаружены "**ира**" или "**ИРА**".

3. В качестве образца поиска введем **Фамилию**, которую нужно найти. Затем нажать кнопку **Найти далее**.

В форме, расположенной "под" окном **Поиска**, на первом плане будет размещена запись с соответствующей фамилией. Если результат поиска не устраивает, то нажмем кнопку **Найти далее**, и программа найдет следующую запись с такой же фамилией (если она есть). Таким образом, можно просмотреть всех клиентов с выбранной фамилией и найти нужную.

4. При удовлетворительном результате поиска окно диалога Поиск можно закрыть.

Задания для тренировки

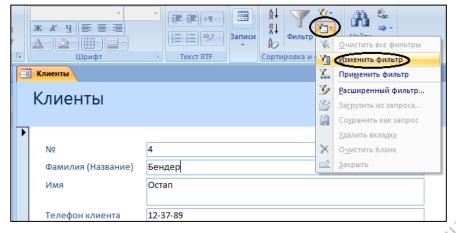
- 1. Найдите клиента, зная только его фамилию.
- 2. Определите, кто вам звонил, если на вашем телефонном определителе номера остался номер телефона звонившего.
- 3. Найдите клиента, зная только первые три цифры его номера телефона.
- 4. Найдите клиента, если вы помните только то, что он живет в каком-то переулке.
- 5. Найдите клиента, если вы помните только номер его квартиры.
- 6. Найдите клиента по первым буквам его фамилии.
- 7. Найдите клиента, если вы помните только то, что ваш номер квартиры входит либо в адрес, либо в номер телефона адресата.

2.4. Фильтры в Ms Access

Способ, рассмотренный в предыдущем упражнении, не всегда самый рациональный. Предположим, что в вашей базе данных несколько людей с заданными именами. Пролистывать карточки не особенно удобно. Проще было бы увидеть список сразу всех адресатов, имеющих нужное имя. Сделать это можно при помощи фильтра.

Фильтр - это набор условий, предназначенный для поиска записей или их сортировки.

Воспользуемся панели инструментов.



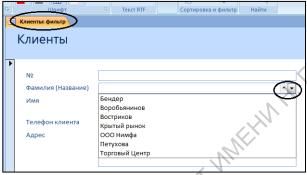


Рис.2.43. Окно фильтра

Для того чтобы выбрать поле из списка необходимо щелкнуть мышью в первой ячейке бланка фильтра и выбрать значение поля в раскрывающемся списке или просто ввести условие отбора для включения в бланк поля, т.е. ввести образец совпадения (имя, которое ищем).

Далее созданный фильтр нужно применить к записям формы. Сделать это можно с помощью одного из следующих действий:

- нажать кнопку Применение фильтра панели инструментов.
- нажать правую кнопку мыши, установив указатель в окне фильтра, и выбрать в контекстном меню команду **Применить фильтр**.

Результатом является список всех людей, имеющих заданное имя. Когда такой список перед глазами намного проще разобраться, кто есть кто.

Если снова необходимо увидеть список всех адресатов, то нужно выполнить команду Удалить фильтр панели инструментов.

Если необходимо снова применить тот же самый фильтр, достаточно выполнить команду Применить фильтр

В выражениях для условий отбора допускается использование операторов шаблона. Приведем некоторые символы шаблона.

Звездочка (*) - заменяет любую группу любых символов; может быть первым или последним символом в шаблоне. Имея условием "Вас*", будут отобраны и "Василиса", и "Василий", и "Васька". Имея условием "*ова", можно отобрать все фамилии, оканчивающиеся на "ова" - "Иванова", "Петрова" и т. д.

Знак вопроса (?) - заменяет любой один символ. Если иметь в качестве условия "М?ша", то будут отобраны и "Миша" и "Маша".

Знак номера (#) - заменяет любую одну цифру. Задав условие "**1**#3", найдете "**103**", "**113**", "**123**"...

Задания для тренировки

- 1. Примените к форме фильтр и закройте ее. Вновь откройте ту же самую форму. Проверьте, сохранился ли отфильтрованный список или показаны все имеющиеся записи и сохранился ли сам фильтр (можно ли применить ранее созданный фильтр).
- 2. Используя фильтр, получите список всех клиентов, у которых номер телефона начинается на три одинаковые цифры (456 или 789).
- 3. Создайте список адресатов, имеющих мужские фамилии оканчивающиеся, например, на **"ов**".
- 4. Создайте список адресатов, у которых имена оканчиваются на "я".
- 5. Задайте фильтр таким образом, чтобы в списке оказались адресаты, имеющие номера телефонов 78-94-56 и 78-91-23.
- 6. Закройте форму Клиенты.

2.5. Элементы управления

Предположим, в форме решено отразить тип адресата (физическое или юридическое лицо). Выбор для каждого небольшой (всего два варианта), поэтому неудобно для каждого адресата вписывать значение пола, проще было бы выбрать одну из двух возможностей. Решим эту задачу с помощью элемента управления.

Элементы управления - это графические объекты, размещенные в форме или отчете и предназначенные для изображения данных, выполнения операций или просто для красоты.

Наиболее распространенная разновидность элементов управления - текстовое поле, которое позволяет размещать текст или цифры. Его можно использовать для ввода новых

данных или редактирования имеющихся. Но для выполнения поставленной задачи значимыми являются флажок или переключатель.

Немного усложним задачу. Предположим, мы хотим, чтобы результат использования переключателя в форме отражался в исходной таблице. Для этого предварительно нужно создать в таблице новое поле.

2.5.1. Создание нового поля в таблице

1. Откройте таблицу **Клиенты** (не путать с формой **Клиенты**) в режиме **Конструктора**.

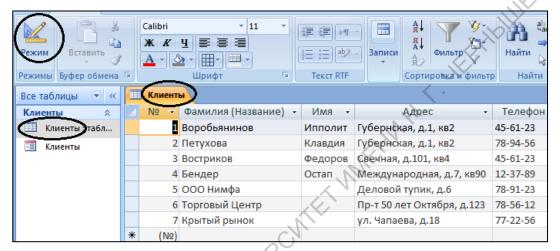


Рис.2.44. Таблица Клиенты

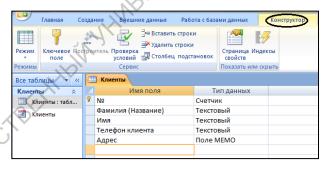


Рис.2.45. Таблица Клиенты в режиме Конструктор

2. Установите курсор в первую пустую ячейку списка полей.

При создании полей таблицы в режиме **Конструктора** следует придерживаться ряда правил:

- имена полей в таблице должны быть уникальными;
- имена полей могут содержать не более 64 символов, включая пробелы;
- имя поля не должно начинаться с пробела;
- имена полей могут содержать любые символы за исключением точки (.), восклицательного знака (!), символа (`) и прямых скобок ([]);
- вместо пробела в именах полей рекомендуется использовать символ подчеркивания или писать все слова слитно, но каждое начинать с заглавной

буквы. Например, Код_отдела, КодОтдела, Наименование_товара, НаименованиеТовара;

- 3. Введите имя поля, например, "Тип".
- 4. Установите курсор в столбце Описание строки Тип и введите текст "1-физическое лицо, 2-юридическое лицо".

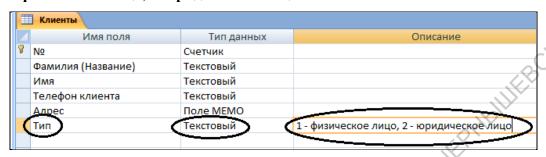


Рис.2.46. Создание нового поля Тип

При установлении переключателя формы в одно из положений (физическое лицо/юридическое лицо), в таблице этот результат будет отражаться цифрами (1/2).

Переключитесь в режим **Таблицы** режим, установите курсор в любую ячейку столбца **Тип.** Обратите внимание на левую часть строки состояния (расположена в нижней части окна) и определите, для чего служит текст, введенный в **Описание**. При запросе на сохранения таблицы ответьте **Да**.

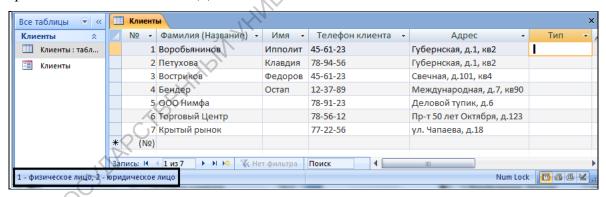


Рис.2.47. Строка состояния

2.5.2. Вставка элемента управления в форму

Представьте форму **Клиенты** в режиме **Конструктора**. Для этого в окне база данных выделите форму **Клиенты** и нажмите кнопку **Конструктор**.

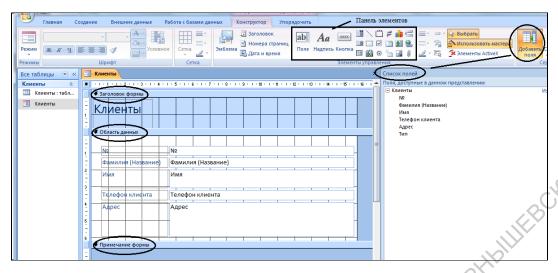


Рис.2.48. Окно Конструктор формы

Список полей базовой таблицы выводится в отдельное окно небольшого размера.

Если список полей отсутствует, вывести его можно нажатием кнопки панели инструментов.

Область данных - служит для размещения основных элементов управления.

Панель элементов позволяет добавлять элементы управления в формы или отчеты.

Для добавления элемента управления следует выбрать соответствующий инструмент.

При переводе в форму указатель приобретает вид крестика с расположенным рядом значком выбранного элемента управления.

Требуется установить указатель в форме в позиции, в которой должен находиться верхний левый угол элемента управления, и нажать кнопку мыши.

Панель элементов содержит элемент управления **Надпись** надпись, предназначенный для изображения произвольного текста (заголовка, сообщения, инструкций и т. п.). Поскольку подписи не связаны с полями базы данных, они являются постоянными, т. е. не меняются при перемещении по записям формы.

Надпись можно изменить или удалить, выделив имеющуюся. Можно добавить новую надпись, выбрав инструмент панели элементов.

Надписи можно перемещать.

Область заголовка, область данных формы можно расширить, если опустить нижнюю границу, "схватив" ее мышью.

Рассмотрим создание переключателя для определения типа клиента.

- 2. Определить место в форме, куда нужно поместить верхний левый угол группы и щелкните мышью. С этого момента начинает свою работу **Мастер по созданию группы** переключателей. Область данных предварительно можно расширить.
- 3. В первом окне диалога ввести значения: физическое лицо и юридическое лицо. Нажать кнопку Далее.

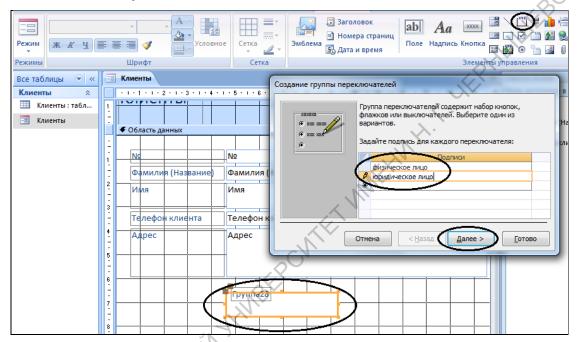


Рис.2.49. Мастер создания группы переключателей

- 4. В следующем окне отказаться от необходимости в выборе значения по умолчанию. Нажать Далее.
- 5. В следующем окне нажать **Далее**, чтобы использовать значения для переключателей, предложенные по умолчанию.

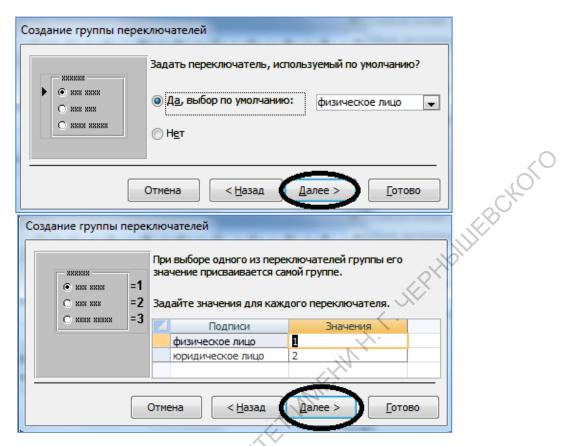
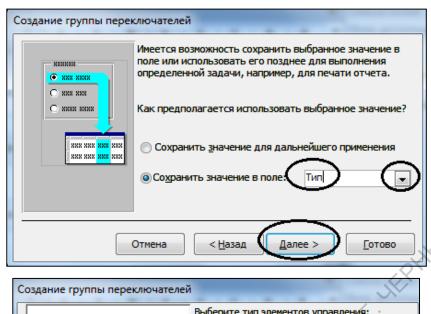


Рис.2.50. Ввод значений группы переключателей и выбор значения переключателя по умолчанию

- 6. В следующем окне щелкнуть **Сохранить значение** и в списке полей выбрать **Тип**. Нажать **Далее**.
- 7. В следующем окне определить вариант оформления группы и выбрать тип элементов управления. Нажать Далее.



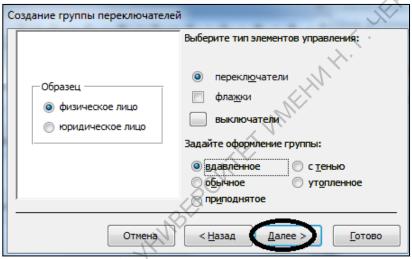


Рис.2.51. Сохранение значений в поле Тип

8. Далее ввести подпись для группы - Тип и нажать кнопку Готово.

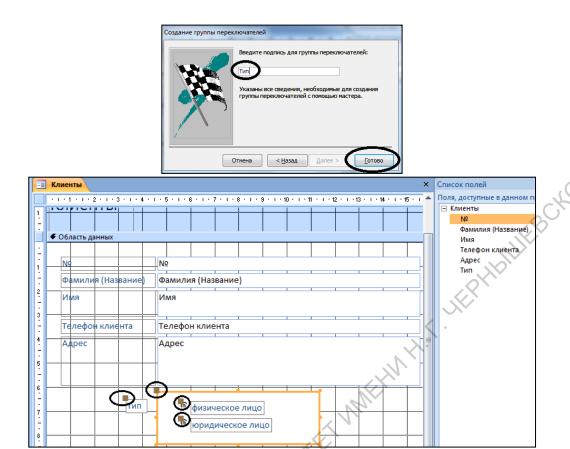


Рис.2.52. Режим Конструктор формы Клиенты

Далее нужно переключиться в режим Формы.

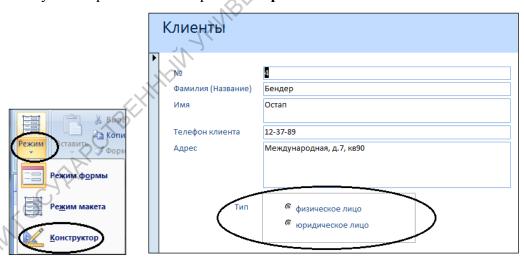
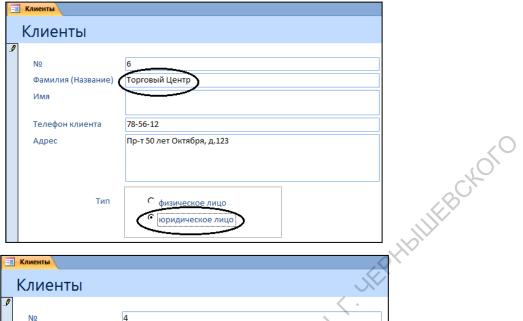


Рис.2.53. Режим формы

2.5.3. Использование нового элемента управления

Пролистать все записи формы и для каждой из них установить переключатель в нужное положение.



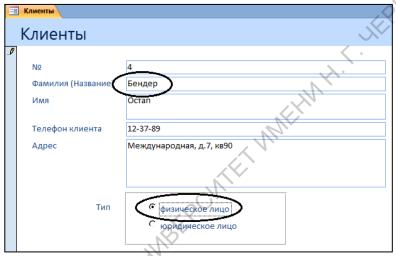


Рис.2.54. Использование группы переключателей **Тип** Закройте форму, сохранив ее.

Откройте таблицу **Клиенты** и проверьте, каким образом **отражены** в ней значения поля **Тип**.

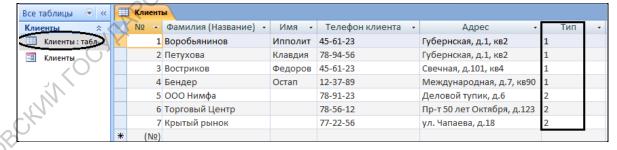


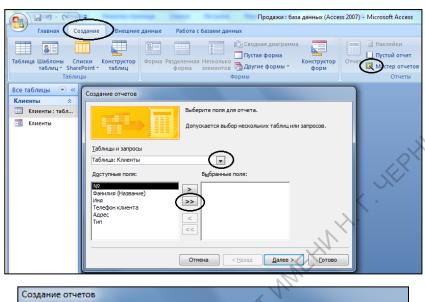
Рис.2.55. Таблица Клиенты

2.6. Отчеты

Одной из основных задач создания и использования баз данных является предоставление пользователям необходимой информации на основе существующих данных. Для этих целей предназначены **отчеты**. Кроме данных, полученных из таблиц, в отчете могут отображаться вычисленные значения и итоговые суммы.

2.6.1. Создание отчета с помощью Мастера

1. В главном меню выбрать команду Создание, затем на Ленте кпопку Мастер отчетов. В окне диалога Создание отчетов выбрать все поля таблицы Клиенты. Нажать кнопку Далее.



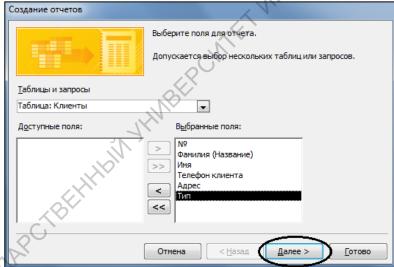
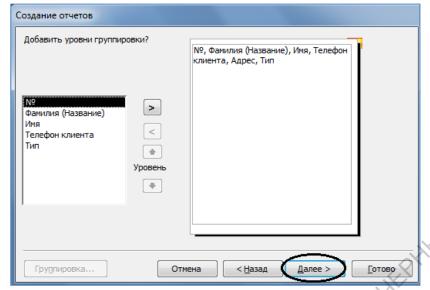


Рис.2.56. Создание отчета с помощь мастера

2. На следующем шаге определяются уровни группировки. Можно пропустить этот шаг щелкнуть кнопку **Далее**.

Для сортировки выбрать поле Фамилия(Название). Нажать Далее.



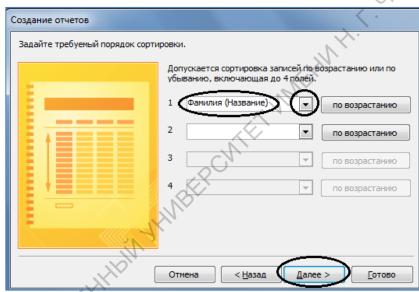


Рис.2.57. Уровни группировки отчета и порядок полей сортировки отчета

3. Выбрать макет отчета В столбец. Нажать Далее.

Определить стиль отчета. Нажать Далее.

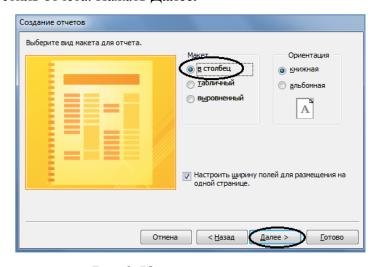


Рис.2.58. Выбор макета

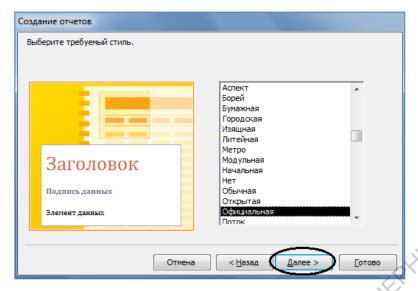


Рис.2.59. Выбор стиля отчета

4. В последнем окне диалога нужно ввести заголовок отчета, который будет отражен в отчете. Здесь же предлагается вывести отчет в режиме **Предварительного просмотра** (просмотреть отчет с данными) или в режиме **Конструктора** (продолжить разработку отчета). Выберем режим просмотра. Нажать кнопку **Готово**.

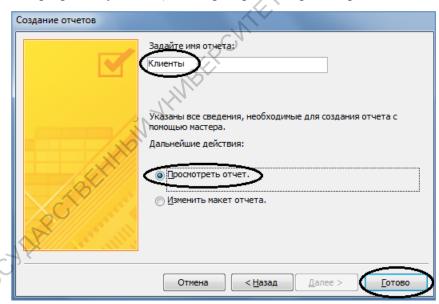


Рис.2.60. Ввод имени отчета

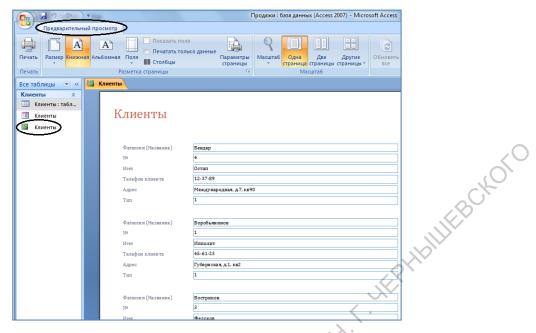


Рис.2.61. Отчет в режиме просмотра

2.6.2. Конструктор отчета

Часто возникает необходимость внести какие-то изменения в отчет. Например, удалить текущую дату, которая выдается в конце отчета. В этом случае необходимо переключиться (аналогично тому, как работали с таблицей и формой) в режим Конструктора отчета.

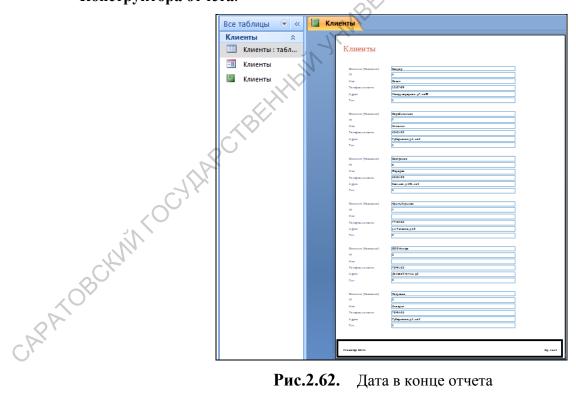
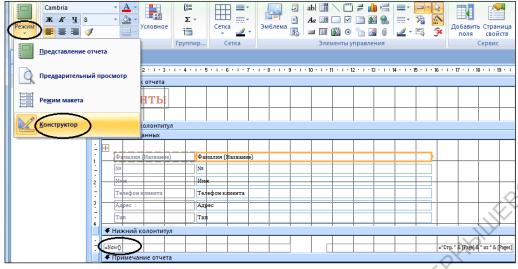


Рис.2.62. Дата в конце отчета



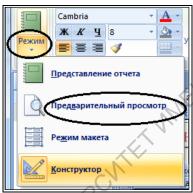


Рис.2.63. Конструктор отчета

Окно **Конструктора отчета** аналогично окну **Конструктора формы.** Поделено оно на четыре области: области верхнего и нижнего колонтитулов, в которых размещаются подписи (заголовок) и поля (дата и номер страниц), область данных и область примечаний.

Для того чтобы удалить текущую дату, нужно выделить соответствующее поле в нижнем колонтитуле и нажать клавишу **Delete**.

Просмотреть отчет можно, выбрав для этого команду Предварительный.

В режиме Конструктора можно, выделив соответствующие элементы выбирать:

- шрифт;
- размер шрифта;
- стиль начертания символов;
- выравнивание.

Для этого используют те же самые кнопки панели инструментов, что и в Microsoft Word, Microsoft Excel и т. д.

2.6.3. Печать отчета

Для печати отчета можно выбрать команду **Печать** меню **Файл** или, находясь в режиме просмотра.

Если выбрана команда **Печать**, то появится окно диалога **Печать**, которое позволяет выбрать принтер, определить страницы для печати и число копий.

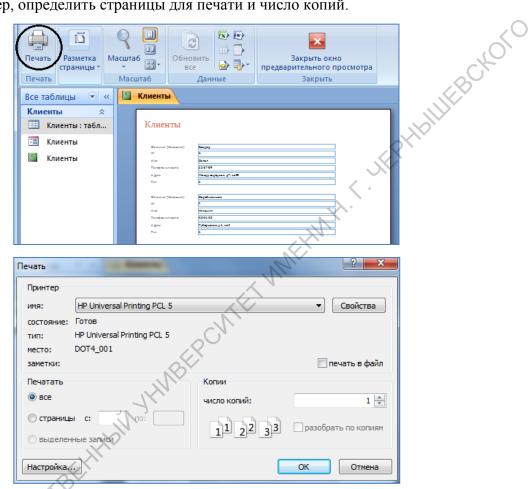


Рис.2.64. Окно диалога Печать

Закройте отчет. Откройте таблицу Клиенты в режиме Конструктора и переименуйте поле **№** в поле **КодКлиента.**

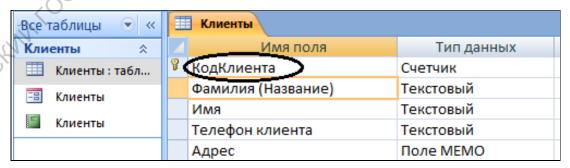


Рис.2.65. Переименование поля

Задания для тренировки

- 1. Для таблицы **Клиенты**, подготовьте отчет, содержащий фамилии, имена и номера телефонов клиентов, выбрав на первом шаге **Мастера отчетов** необходимые поля.
- 2. Отсортируйте записи таблицы **Клиенты** по номерам телефонов, задав такой порядок сортировки в **Мастере отчетов**. Подготовьте отчет таким образом, чтобы сначала в нем шли номера телефонов, а затем имена и фамилии лиц, имеющих эти телефоны. Именно в таком порядке выбирайте поля таблицы на первом шаге **Мастера отчетов** для отчета в процессе работы **Мастера** по созданию отчета.
- 3. Подготовьте отчет для таблицы **Клиенты** таким образом, чтобы сначала в нем шли имена клиентов, затем фамилии и адреса. Именно в таком порядке выбирайте поля таблицы для отчета в процессе работы **Мастера** по созданию отчета.

2.7. Связи и фильтры

1. Создадим еще одну таблицу **Товары**, определив способ создания таблиц с помощью **Конструктора**.

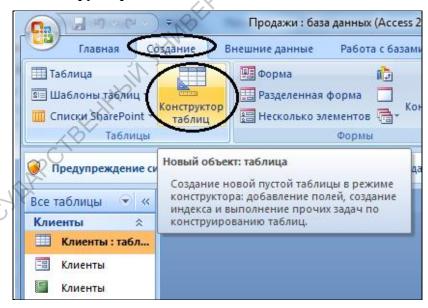


Рис.2.66. Создание таблицы Товары с помощью Конструктора

2. Заполним первый столбец именами полей, создаваемой таблицы. Во втором столбце укажем для каждого поля тип данных. Тип данных выбирается из раскрывающегося списка второго столбца.



Рис.2.67. Поля и типы данных новой таблицы

3. Отметим ключевое поле рассматриваемой таблицы — **КодТовара**. Чтобы поле стало ключевым, нужно установить текстовый курсор в любое место данного

поля, выполнить команду на панели инструментов соответствует кнопка

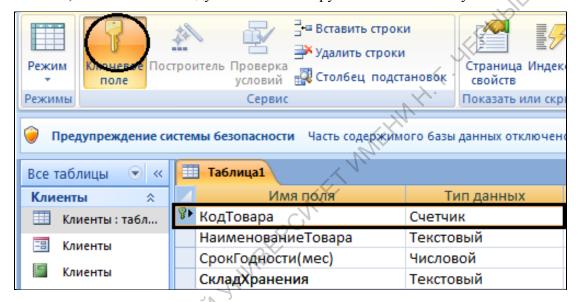
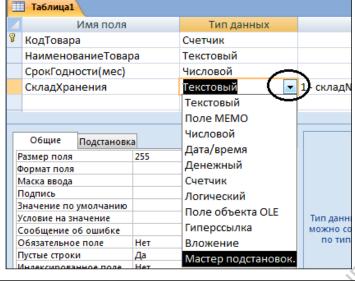


Рис.2.68. Создание ключевого поля

- 4. В поле **СкладХранения** нашей таблицы будем введено одно из трех возможных значений (1,2,3); имеет смысл заполнять это поле, выбирая из фиксированного набора значений. Выберем в этом поле тип данных **Мастер подстановок...**
 - 5. В качестве столбца подстановки можно использовать или данные другой таблицы, или фиксированный набор значений. Выберем в появившемся диалоговом окне первого шага мастера второй вариант и щелкнем кнопку Далее.



HPIIIE BOKOLO

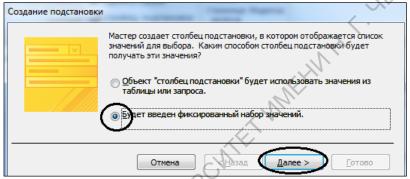
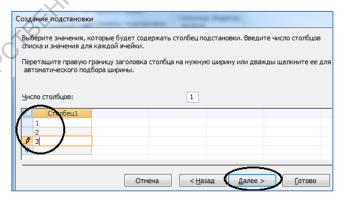


Рис.2.69. Определение типа поля СкладХранения

На следующем шаге введем номера складов. Чтобы перейти в следующую ячейку надо нажать клавишу Вниз на клавиатуре. Нажатие клавиши Enter - соответствует щелчку по кнопке Далее.



CAPATOBCKNINTOCYTLE Рис.2.70. Фиксированный список номеров складов

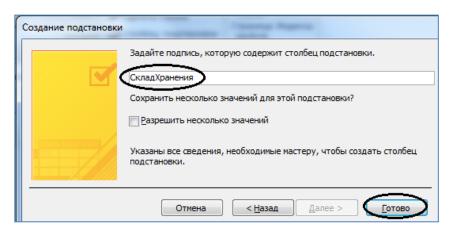


Рис.2.71. Ввод имени подстановки

На последнем шаге определяется подпись в столбце подстановки. Щелкнем кнопку **Готово**.

Изменим свойства столбца подстановки. Проверим, что выделена строка СкладХранения и выберем вкладку Подстановка в свойствах поля.

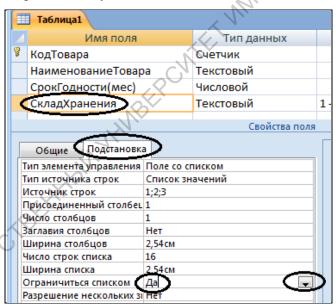


Рис.2.72. Свойства поля СкладХранения

Установим в свойстве Ограничиться списком значение Да.

6. Перейдем в режим **Таблицы** режим, предварительно сохранив таблицу под именем **Товары**.

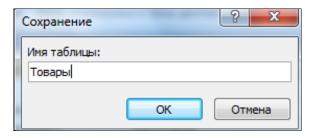


Рис.2.73. Сохранение таблицы в базе данных

Если все предыдущие задания были сделаны верно, должна получиться следующая таблица:

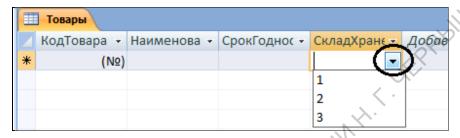


Рис.2.74. Новая таблица Товар

Поле **СкладХранения** можно заполнять с клавиатуры или выбирая значения из раскрывающегося списка. При попытке в это поле ввести не **1**, **2**, **3**, а что-либо другое появится сообщение об ошибке. Ответ – **Нет**, удалите цифру и закройте таблицу.

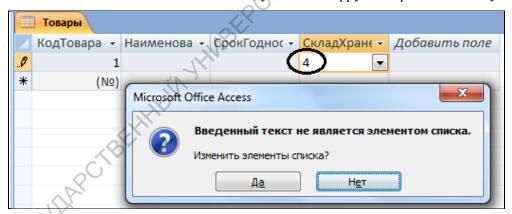
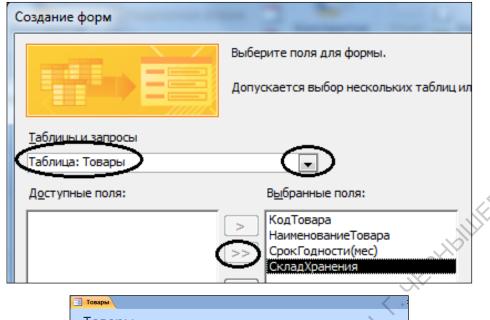


Рис.2.75. Сообщение об ошибке

Задание для тренировки

1. Создайте с помощью мастера форму на основе таблицы **Товары**. При создании формы выберите все поля таблицы.



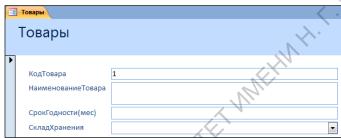
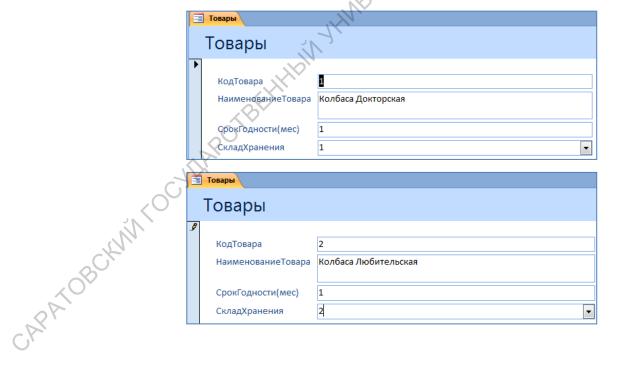


Рис.2.76. Форма на основе таблицы Товары

2. Внесите в форму Товары семь записей.



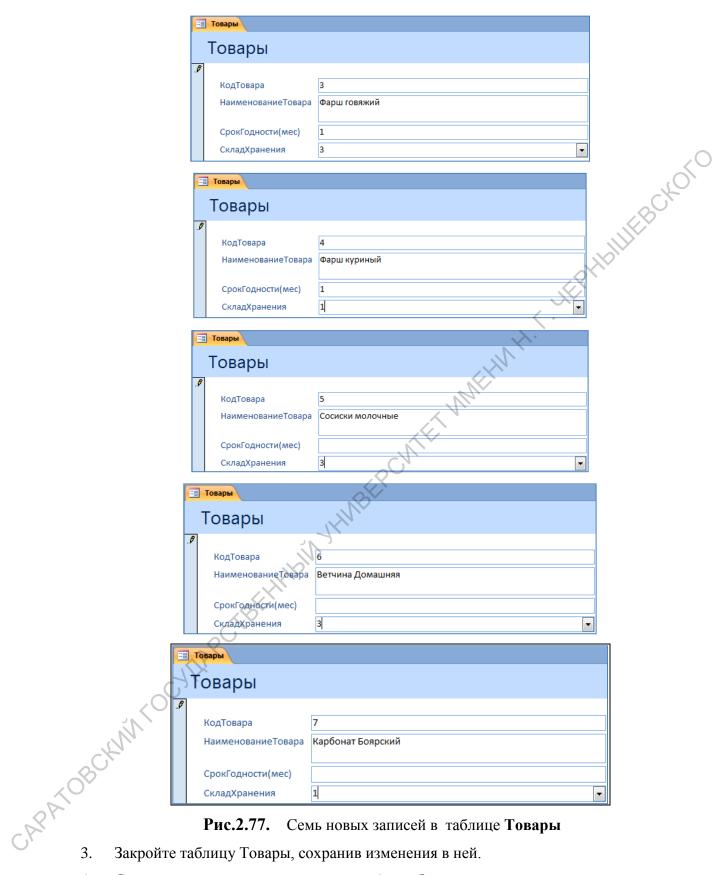


Рис.2.77. Семь новых записей в таблице Товары

- Закройте таблицу Товары, сохранив изменения в ней.
- 4. Создайте с помощью конструктора таблицу Заказы со следующими полями:

Имя поля	Тип данных
КодЗаказа	Счетчик
ДатаЗаказа	Дата/Время
Клиент	Числовой
Товар	Числовой
Цена	Денежный
Количество	Числовой
Доставка	Логический

- 5. Определите поле КодЗаказа как ключевое поле.
- 6. Сохраните таблицу под именем Заказы.

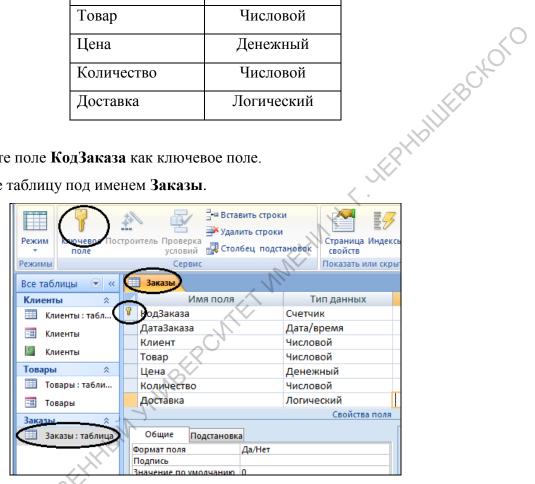


Рис.2.78. Таблица Заказы в режиме Конструктор

7. Для поля ДатаЗаказа установите для свойства Маска ввода - Краткий формат даты. Выбор происходит из раскрывающегося списка в этих свойствах.

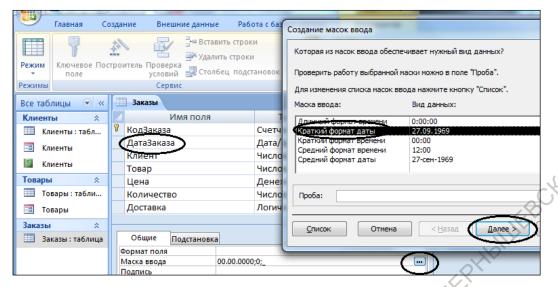


Рис.2.79. Свойства поля Дата заказа

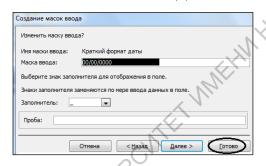


Рис.2.80. Маска ввода даты

8. Закройте таблицу.

2.8. Подчиненные формы

2.8.1. Установление связей между таблицами в окне Схема Данных

1. Откроем схему базы данных.

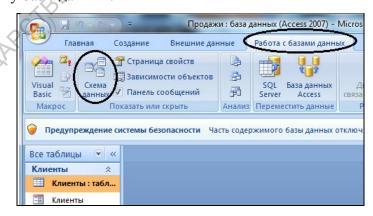


Рис.2.81. Открытие схемы данных

2. В схему, прежде чем устанавливать связи, нужно добавить те таблицы, между которыми эти связи будут установлены.

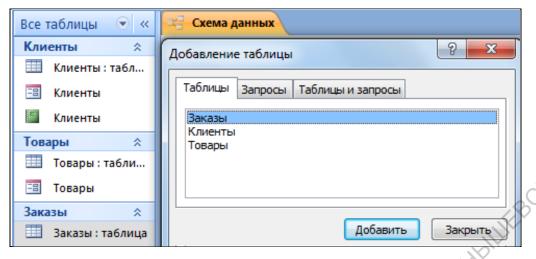


Рис.2.82. Окно диалога для выбора таблиц

- 3. В окне диалога Добавление таблицы поочередно нужно выбрать две таблицы Заказы и Клиенты нажать, для каждой из них, кнопку Добавить. Нажать кнопку Закрыть.
- Для установления связей нужно переместить мышью поле № из схемы таблицы
 Клиенты на место поля Клиент таблицы Заказы. Появится окно диалога Связи.

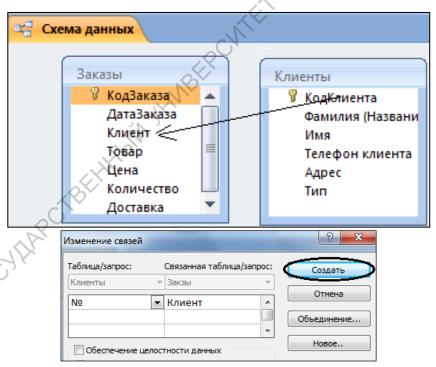


Рис.2.83. Установление связей между таблицами Клиенты и Заказы

5. Нажать кнопку **Создать**. В окне диалога **Схема данных** установленные связи отображаются графически.

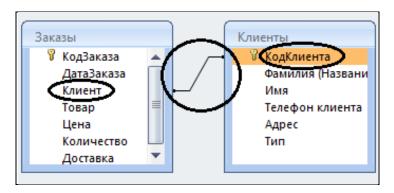


Рис.2.84. Графическое отображение связей

Для редактирования связей достаточно двойного щелчка мыши по линии, изображающей связь.

6. Закрыть окно диалога **Схема данных**, ответив **Да** на вопрос о сохранении макета **Схемы данных**.

2.8.2. Установление связей между таблицами с помощью Мастера подстановок

1. Открыть таблицу Заказы в режиме Конструктора.

CAPATOBOKNÍNFO

- 2. Установить курсор в поле Товар и выбрать Тип данных Мастер подстановок.
- 3. Поле **Товар** будет заполняться с помощью раскрывающегося списка, элементы которого являются строками таблицы **Товары**.

В появившемся диалоговом окне по умолчанию выбран нужный вариант. Щелкнуть кнопку Далее.

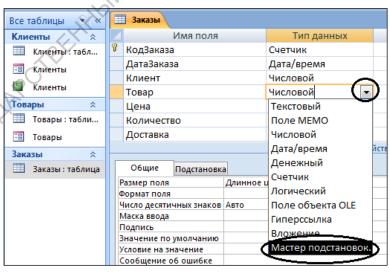


Рис.2.85. Изменение типа данных поля Товар

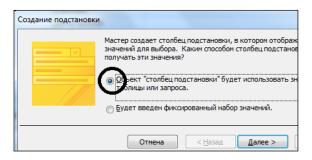


Рис.2.86. Выбор объекта подстановки

- 4. На следующем шаге указываем таблицу источник элементов списка (таблица **Товары**).
- 5. Далее необходимо выбрать поля таблицы **Товары**, с которым устанавливается связь (**Наименование товара**).

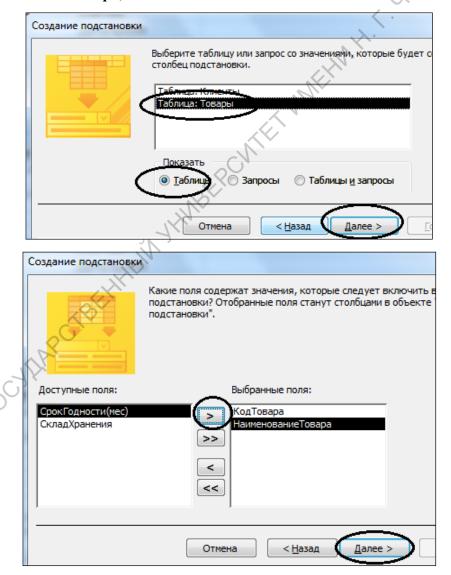


Рис.2.87. Выбор полей для подстановки

6. На следующем шаге выбираются поля для сортировки списка подстановки

7. Затем отображается содержимое выбранного столбца. Здесь же можно по необходимости увеличить ширину столбца.

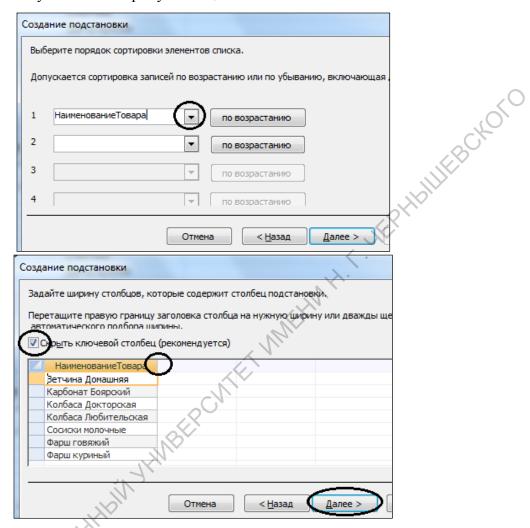
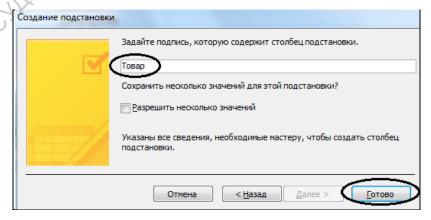


Рис.2.88. Сортировка и содержимое поля связи

8. В следующем окне мастера определяется подпись для столбца подстановки. Можно оставить подпись, предложенную мастером.



CAPATOBCKNINTC

Рис.2.89. Подпись для столбца подстановки

- 9. Щелчок по кнопке Готово дает команду на создание столбца подстановки. Программа выдаст сообщение о необходимости сохранения таблицы, с которым нужно согласиться.
- 10. Перейти в режим просмотра Таблицы и убедиться, что в поле Товар появился раскрывающийся список.

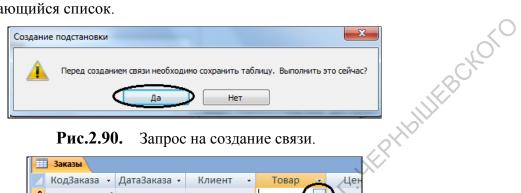


Рис.2.90. Запрос на создание связи.

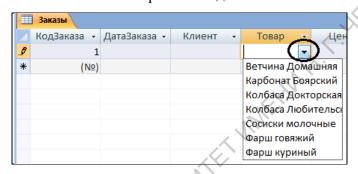


Рис.2.91. Поле Товар содержит раскрывающийся список

11. Закрыть таблицу Заказы и посмотреть Схему Данных. Убедиться, что между таблицами Заказы и Товары появилась линия, обозначающая связь между ними.

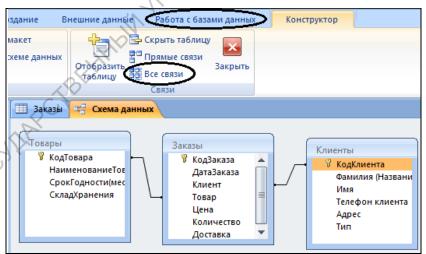


Рис.2.92. Связи базы данных

CAPATOBCKNINTO Если таблица Заказы не видна, необходимо добавить ее, воспользовавшись кнопкой 🕮 Все связи на панели инструментов

Объекта в окне схемы данных можно перетаскивать мышью за заголовки таблиц.

2.8.3. Создание сложной формы

Создадим форму, содержащую данные из двух таблиц.

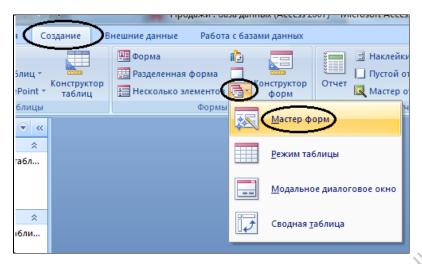


Рис.2.93. Выбор команд для Мастера форм

1. В раскрывшемся окне диалога **Создание форм**, выберем для формы все поля таблицы **Заказы**, кроме поля **Клиент**. Кнопку **Далее** не нажимаем.

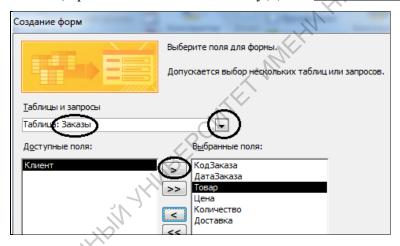


Рис.2.94. Выбор полей для формы из таблица Заказы

2. В раскрывающемся списке Таблицы и запросы выберем таблицу Клиенты. Из этой таблицы для формы возьмем все поля.

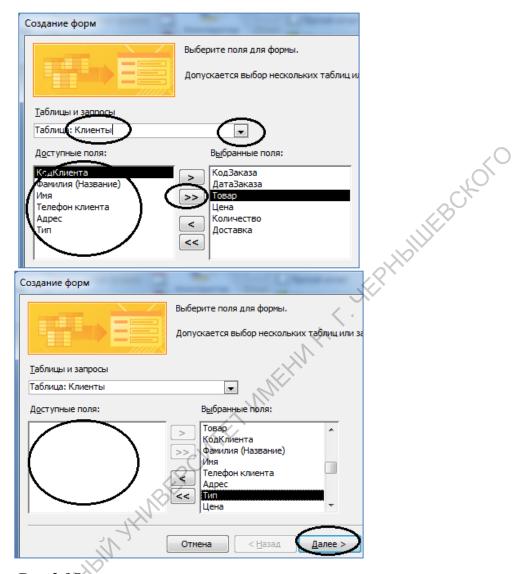


Рис.2.95. Выбор полей из таблицы Клиенты

3. На следующих шагах выберем вид представления по **Клиентам**, внешний вид формы оставляем – **Табличный**. Нажмем кнопку **Далее**.

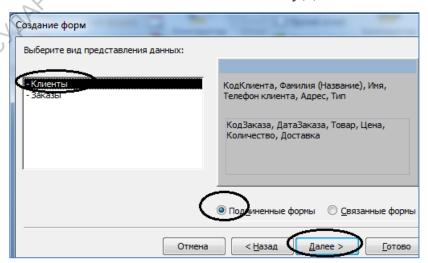


Рис.2.96. Вид представления по Клиентам

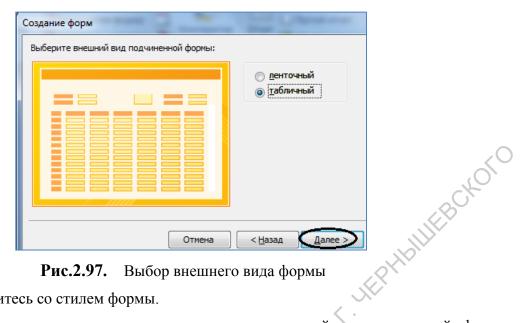


Рис.2.97. Выбор внешнего вида формы

- 4. Определитесь со стилем формы.
- 5. На последнем шаге определяются названия главной и подчиненной форм. Впишем для главной формы – Оформление заказов, для подчиненной оставим вариант, предложенный мастером. Нажмите кнопку Далее.

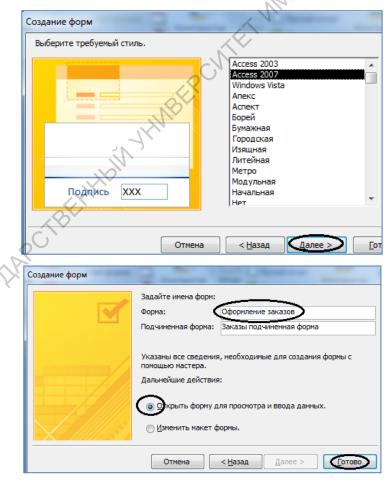


Рис.2.98. Выбор внешнего вида формы, создание имен для главной и подчиненной форм

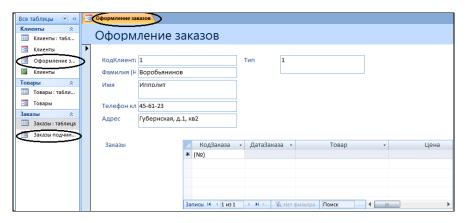


Рис.2.99. Результат создания сложной формы

Сложная форма представляет собой как бы форму в форме. Благодаря этому появляется возможность заполнять разные таблицы одновременно.

Задание для тренировки.

- 1. Откройте форму Оформление заказов в режиме Конструктора.
- 2. Удалите поле с данными Тип из формы Оформление заказов.
- 3. Скопируйте управляющий элемент **Тип** из формы **Клиенты** в форму **Оформление заказов**.

Открыть дополнительно форму **Клиенты**. Выделить прямоугольником все элементы поля **Тип**. Скопировать правой кнопкой мыши. Перейти в **Конфигуратор** формы **Оформление заказов** и **Вставить** или повторить создание управляющих элементов в форме, рассмотренное ранее.

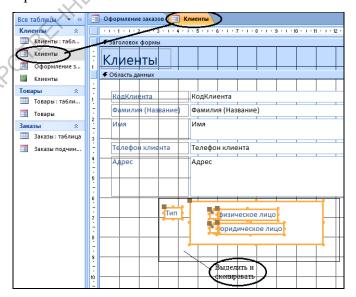


Рис.2.100. Выделение и копирование элемента управления в форме Клиенты

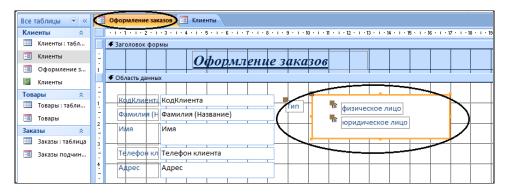


Рис.2.101. Вставка элемента управления в форму Оформление заказов

4. Измените формат элементов в форме по образцу.

Для изменения отдельных элементов в форме необходимо сначала исключить их из макета, иначе изменения будут проблематичны. Для удаления макета выделяем все элементы в области данных (рисуем вокруг них прямоугольник), вызываем меню правой кнопкой мыши, выбираем команду **Макет**, затем **Удалить**.

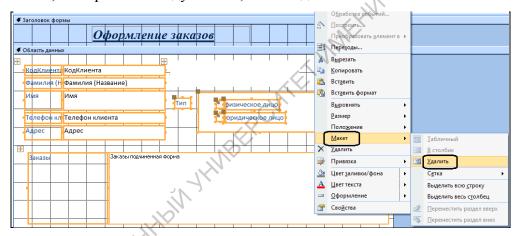


Рис.2.102. Удаление элементов из Макета

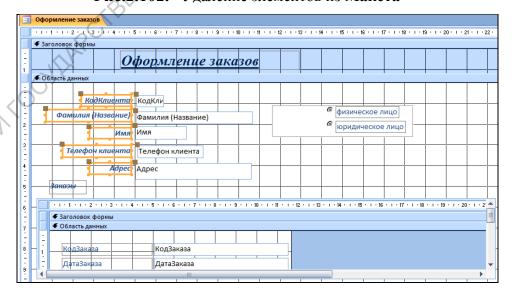


Рис.2.103. Окончательный вид формы в режиме Конструктора

5. Перейдите в режим формы для просмотра результата изменений, сохранив форму, и введите с помощью этой формы по два заказа для каждого клиента.

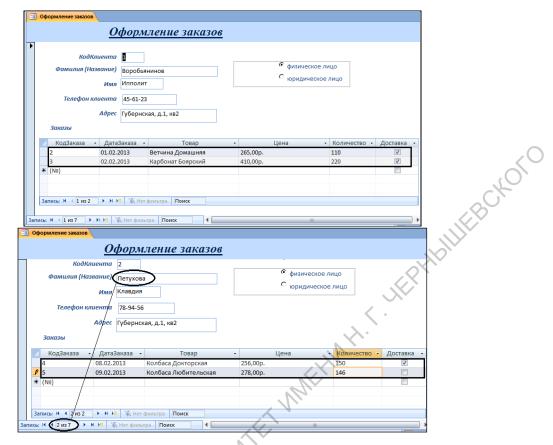


Рис.2.104. Заказы клиентов Воробьянинова и Петуховой

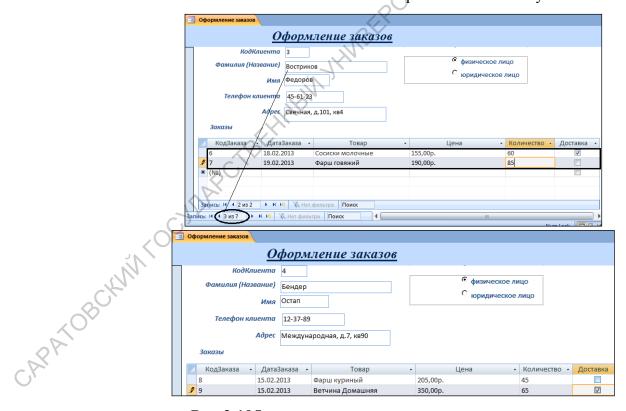


Рис.2.105. Заказы клиентов Вострикова и Бендера

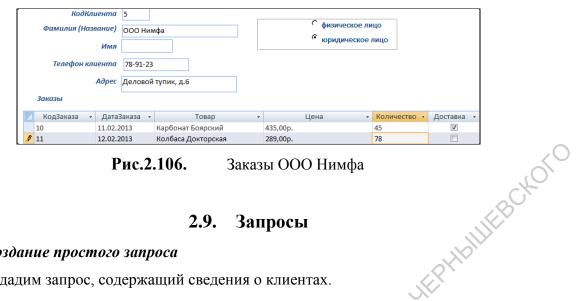


Рис.2.106. Заказы ООО Нимфа

2.9. Запросы

2.9.1. Создание простого запроса

Создадим запрос, содержащий сведения о клиентах.

На панели объектов в окне базы данных нужно выбрать объект Запросы и вариант 1. Создание запроса с помощью мастера.



Рис.2.107. Создание запроса с помощью мастера

2. В появившемся диалоговом окне из списка Таблицы и запросы выбрать таблицу Клиенты. Для запроса выбрать поля Фамилия(Название), Имя, Адрес, Тип. CAPATOBCKNINFOCYT

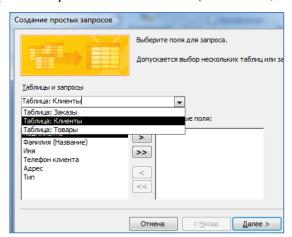


Рис.2.108. Окно диалога Создание простого запроса

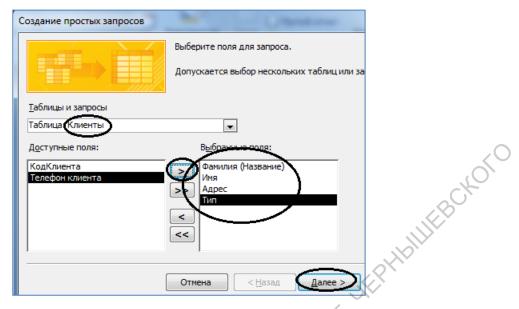
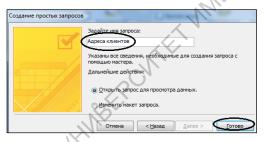


Рис.2.109. Выбор полей запроса

3. На следующем шаге определить название запроса **Адреса клиентов**. Щелкнуть кнопку **Готово**.



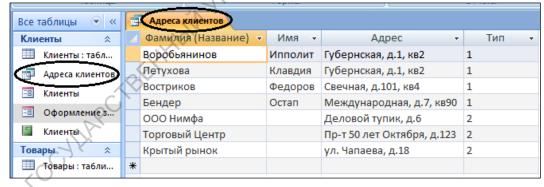


Рис.2.110. Результат запроса

Перейти в режим Конструктора запроса.

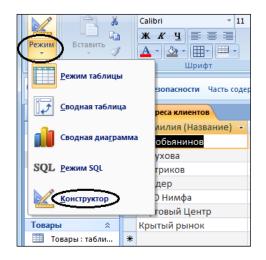


Рис.2.111. Переход в режим Конструктора

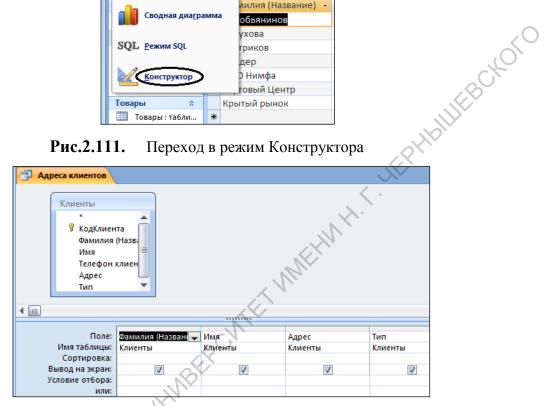


Рис.2.112. Создание запроса

В верхней части окна выводятся таблицы – источники данных, в нижней части для каждого поля запроса можно определить вид сортировки, скрыть или восстановить поле в запросе, наложить условие на выборку.

Отсортируем поле Фамилия (Название) - по возрастанию, наложим условие отбора для поля Tип -1.

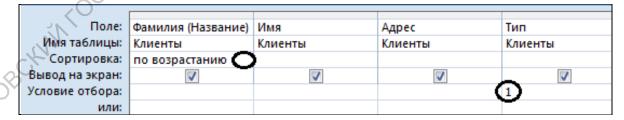


Рис.2.113. Конструктор запроса

Можно переключиться в режим просмотра запроса.

1	Д Адреса клиентов				
	Фамилия (Название) 🔻	VM8 →	Адрес →		Тип
	Бендер	Остап	Международная, д.7, кв90	1	
	Воробьянинов	Ипполит	Губернская, д.1, кв2	1	
	Востриков	Федоров	Свечная, д.101, кв4	1	
	Петухова	Клавдия	Губернская, д.1, кв2	1	

Рис.2.114. Список физических лиц отсортирован в алфавитном порядке

5. Закрыть запрос **Адреса клиентов**, сохранив изменения в структуре запроса. *Задание для тренировки*

- 1. Создайте новый Адреса фирм.
- 2. Измените запрос **Адреса фирм** в режиме **Конструктора** так, что бы он выводил поля **Фамилия (Название), Адреса** только для юридических лиц.

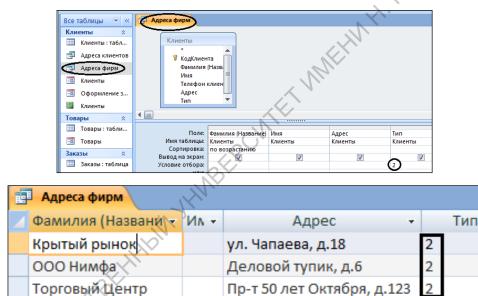


Рис.2.115. Запрос на выборку юридических лиц

2.9.2. Создание вычисляемого поля в запросе

- 1. Создать новый запрос
- 2. Для запроса из таблицы Заказы выбрать поля Код заказа, Дата заказа, Товар, Цена, Количество. Из таблицы Клиенты Фамилия (Название), Тип. Нажмите Далее.

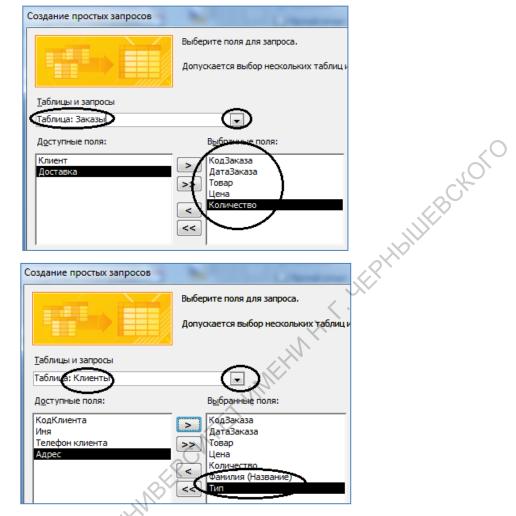


Рис.2.116. Выбор поле запроса из таблиц Заказы и Клиента

3. На следующих шагах создать вид отчета – подробный. Название запроса Сведения о заказах.

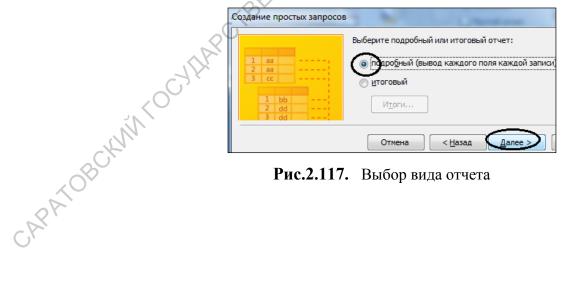


Рис.2.117. Выбор вида отчета

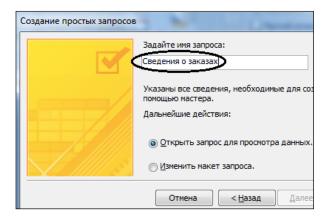


Рис.2.118. Ввод имени запроса

4. В готовом запросе переместим поле **Фамилия (Название**) на второе место (после поля **Код заказа**). Поля переносятся правой кнопкой мыши за заголовки.

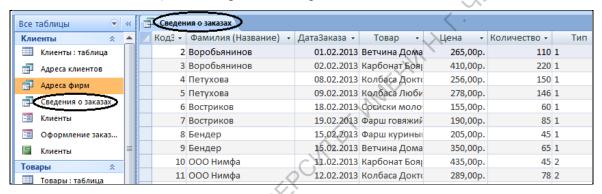


Рис.2.119. Результат запроса Сведения о заказах

5. Добавим в запрос поле **Сумма** заказа. Необходимо перейти в режим **Конструктора** и установить текстовый курсор в первое пустое поле.

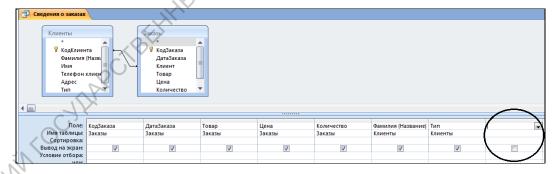


Рис.2.120. Курсор в пустом поле

- 6. Щелкнуть кнопку Построить па панели инструментов на Ленте.
- 7. В диалоговом окне **Построителя выражений** левое поле представляет все источники и инструменты для построения выражения. В середине все поля текущего запроса. Используя кнопку умножить * и поля нашего запроса, постройте выражение: **Цена** * **Количество**. Затем щелкнуть кнопку **ОК**.

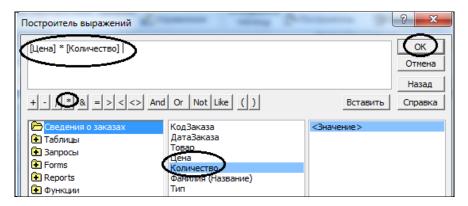


Рис.2.121. Диалоговое окно Построителя выражений

8. Сохранить запрос, перейти в режим просмотра запроса и убедиться, что появилось новое поле Выражение 1.



Рис.2.122. Результат выполнения запроса Сведения о заказах

2.9.3. Создание итогового запроса

Создадим итоговый запрос Среднее значение цены товара.

- Создать новый запрос командой запросов 1.
- 2. Для запроса из таблицы Заказы выбрать поля Товар, Цена. На следующем шаге создать вид отчета – итоговый. Нажать кнопку Итоги. CARATOBCHINIFOCYIL

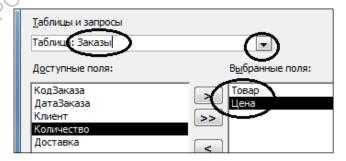


Рис.2.123. Выбор полей для запроса

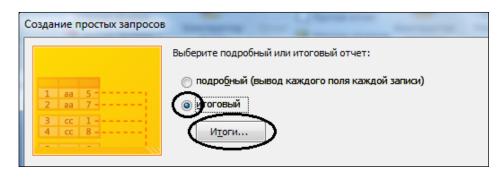


Рис.2.124. Выбор отчета Итоговый

3. В окне диалога **Итоги** включить флажок **Avg** (среднее значение). Нажать кнопку **ОК**. В появившемся окне нажать кнопку **Далее**.

Название запроса Средняя цена товара.

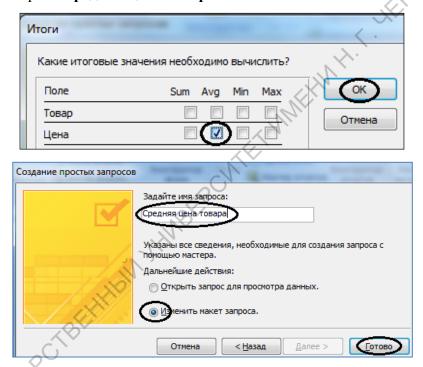


Рис.2.125. Флажок Avg включен

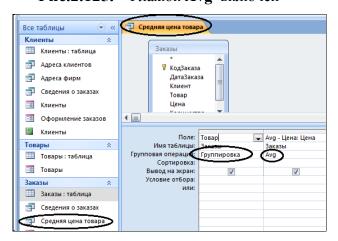


Рис.2.126. Запрос Средняя цена товара

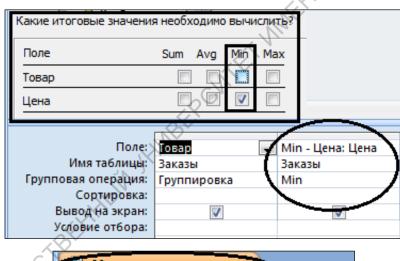


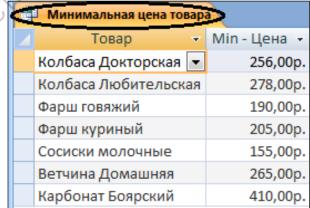
Рис.2.127. Результат выполнения запроса

Средняя цена товара

Задания для тренировки

Аналогично создайте итоговый запрос Минимальная цена товара. 1.





CAPATOBCKMNTOCYTHAP Рис.2.128. Создание итогового запроса Минимальная цена товара

2. Создайте итоговый запрос Максимальная цена товара.

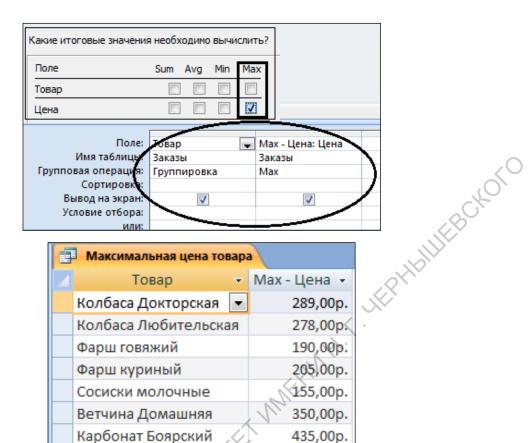


Рис.2.129. Запрос Максимальная цена товара

435,00p.

Запросы можно создавать на основе ранее созданных запросов.

Создадим итоговый запрос Итоги по товарам.

- Создать новый запрос командой запросов 1.
- 2. Для запроса в раскрывающемся списке Таблицы и запросы выберем запрос Сведения о заказах.

Для нового запроса нам достаточно два поля Товар и Выражение1 (цена*количество).

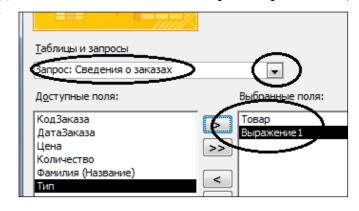


Рис.2.130. Выбор полей запроса из запроса

Сведения о заказах

3. На следующем шаге создать вид отчета – итоговый. Нажать кнопку **Итоги**. В окне диалога **Итоги** включить флажок **Sum**. Название запроса Итоги по товарам

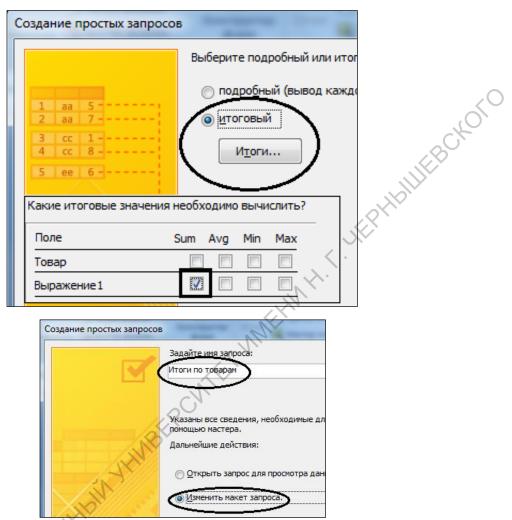


Рис.2.131. Создание запроса Итоги по товарам

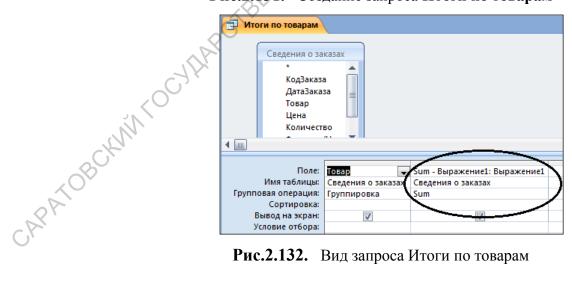


Рис.2.132. Вид запроса Итоги по товарам

Итоги по товарам	
Товар 🔻	Sum - Выра: 🕶
Колбаса Докторская 🔻	60 942,00p.
Колбаса Любительская	40 588,00p.
Фарш говяжий	16 150,00p.
Фарш куриный	9 225,00p.
Сосиски молочные	9 300,00p.
Ветчина Домашняя	51 900,00p.
Карбонат Боярский	109 775,00p.

Рис.2.133. Итоги по товарам

2.10. Гиперссылки

HAIIIEBOKOFO

При вводе заказов в форме **Оформление заказов** может появиться необходимость добавления нового товара в справочник **Товары**. Ввести новый товар можно непосредственно работая с формой, создав гиперссылку на справочник **Товары**.

Гиперссылки дают возможность удобно перемещаться между объектами.

Гиперссылка является указателем с одного объекта на другой. Наиболее часто гиперссылка указывает на Web-страницу, но также может указывать на рисунок, адрес электронной почты. Сама гиперссылка может отображаться как текст или как картинка.

При установке указателя на текст или рисунок, содержащий гиперссылку, указатель примет форму руки, показывающей, что имеется объект, который можно щелкнуть.

Откроем форму Оформление заказов в режиме Конструктора.

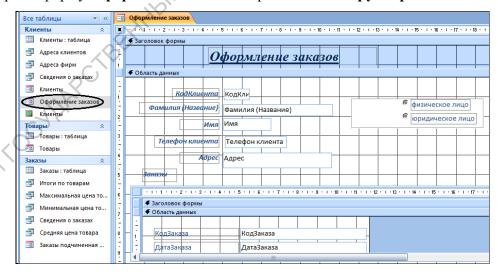


Рис.2.134. Форма Оформление заказов в режиме Конструктор

На Ленте нажать кнопку Вставить гиперссылку.

В диалоговом окне Вставка гиперссылки щелкнуть значок объект в базе данных в области Связать с.

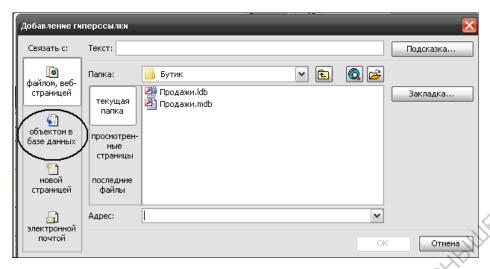


Рис.2.135. Окно диалога Добавление гиперссылки

Выберем из списка объект базы данных, на который должна указывать ссылка. В поле **Текст** введем текст, который должен отображаться в виде гиперссылки. По умолчанию предполагается имя формы **Товары**.

Нажмем кнопку **Подсказка.** Здесь можно ввести текст, который должен появляться при наведении указателя на гиперссылку. Если оставить поле **Текст** подсказки пустым, то в качестве подсказки будут отображаться тип и имя объекта.

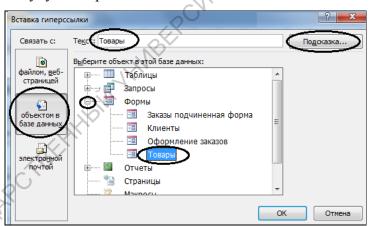


Рис.2.136. Выбор объекта для гиперссылки

Нажмем кнопку ОК. Гиперссылка будет добавлена в надпись.

В области данных появится поле с надписью Товары.

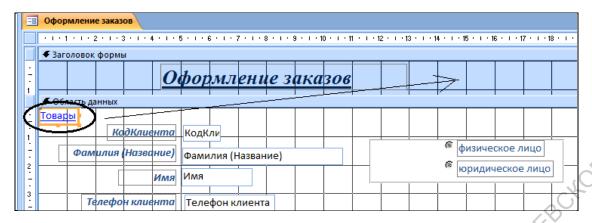


Рис.2.137. Добавлено поле с надписью Товары

Переместим поле с гиперссылкой, например, в заголовок формы, изменим размеры поля и текста.

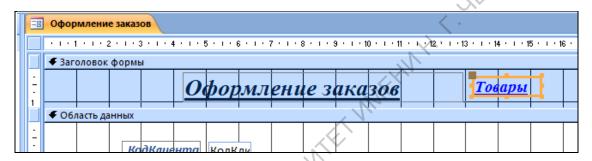


Рис.2.138. Поле с надписью Товары перемещено в заголовок формы

Перейдем в режим формы с помощью кнопки Вид гомом . Нажмем на созданной гиперссылке. Откроется форма Товары. Теперь можно вводить новый товар.

Оформление заказов		
<u>o</u>	формление заказов	<u>Товары</u>
КодКлиента Фамилия (Название) Имя Телефон клиента Адрес	Воробьянинов Ипполит 45-61-23 Губернская, д.1, кв2	© физичес С юридич
□ Оформление заказов Товары	Товары	OHE IIIE

Рис.2.139. Результат добавления гиперссылки на форму

2.11. Кнопки

Кнопки являются удобным средством выполнения действий. Когда пользователь щелкает кнопку, не только выполняется соответствующее действие, но и на экране видно, как кнопка нажимается и отпускается

Например, можно создать кнопку, открывающую другую форму.

На кнопку в форме можно поместить текст или рисунок.

Имеется возможность создать кнопку самостоятельно или с помощью мастера. Мастер ускоряет процесс разработки кнопки, автоматически выполняя все основные действия. Мастер выводит приглашение ввести необходимые сведения и создает кнопку на основе ответов пользователя. С помощью мастера можно создавать более 30 различных типов кнопок.

Создадим новую форму, в которой разместим кнопки, открывающие формы Оформление заказов, справочники Товары, Контрагенты, отчет и запросы. Для этого:

Создать пустую форму в режиме Конструктора.

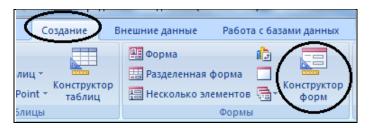


Рис.2.140. Пустая форма в режиме Конструктор

Разместим на ней кнопки. Для этого нужно выбрать в панели элементов на **Ленте** нажать кнопку . Нарисуем прямоугольник в области данных. Стартует мастер создания кнопок.

В списке Категория нужно выбрать Работа с формой и в списке Действия Открыть форму, нажать кнопку Далее.

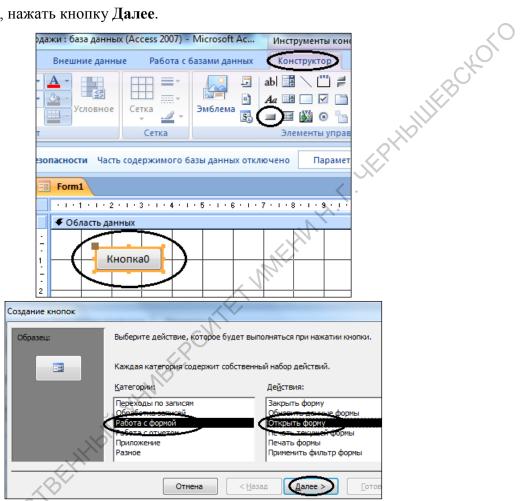


Рис.2.141. Выбор категории и действия для кнопки

На следующем шаге мастера необходимо указать имя формы **Оформление заказов** и нажать кнопку **Далее**.

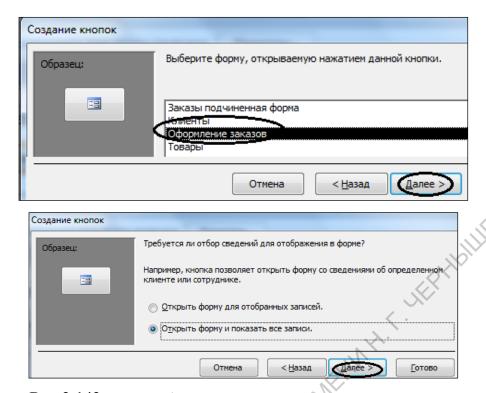


Рис.2.142. Выбор формы, открываемой нажатием кнопки

Следующий шаг позволить сделать надпись на кнопке или выбрать рисунок. Введем в поле **Текст** подпись для кнопки **Оформление заказов**. И в последнем окне мастера создания кнопок нажать **Готово**.

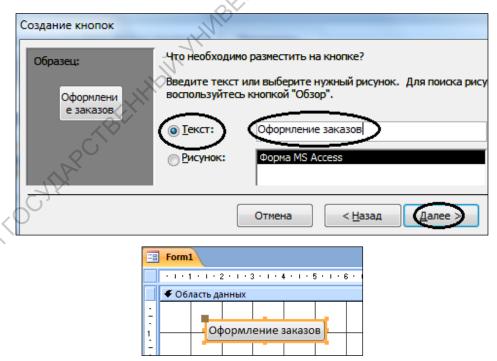


Рис.2.143. Надпись на кнопке и результат добавления кнопки

Аналогично можно создать кнопки для открытия формы **Товары**, формы **Контрагенты**, отчета **Клиенты**, запроса **Заказ с доставкой**, **Заказ без доставки**.

Для создания кнопки, открывающей отчет, нужно выбрать **Категория – Работа с** отчетом, Действия – Просмотр отчета.

Для создания кнопки, открывающей запрос, нужно выбрать **Категория – Разное**, Действия – Выполнить запрос.

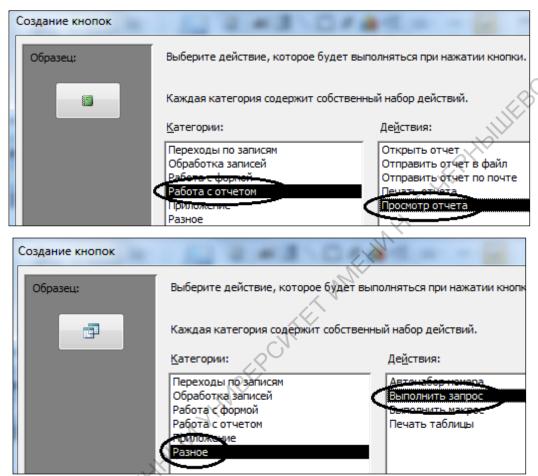


Рис.2.144. Выбор категории и действия для отчета запроса

Создадим еще две кнопки на открытие справочников Товары и Клиенты. Перейдем в режим формы.

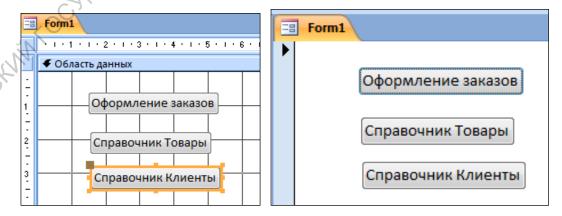


Рис.2.145. Результат добавления кнопок на форму

Данная форма позволит перейти к другим формам.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

- 1. Информационные системы и базы данных. Организация и проектирование [Текст] / В. Ю. Пирогов. Санкт-Петербург: БХВ-Петербург, 2009. 528 с –ISBN978-5-9775-0399-0 Рекомендовано УМО в обл. инновационных междисциплинар. образоват. программ в качестве учеб. пособия (10 экз. НБ).
- 2. Базы данных. Разработка приложений [Текст] : для студента / Л. В. Рудикова. Санкт-Петербург : БХВ-Петербург, 2006. 487, с. ISBN 5-94157-805-9 (40 экз. НБ).
- 3. Информационные технологии в науке и образовании [Электронный ресурс] : Учебное пособие / Елена Леонидовна Федотова, Андрей Александрович Федотов. Москва : Издательский Дом "ФОРУМ" ; Москва : ООО "Научно-издательский центр ИНФРА-М", 2013. 336 с. ISBN 978-5-8199-0434-3
- 4. Информационные системы [Электронный ресурс]: Учебное пособие / Ольга Леонидовна Голицына, Игорь Иванович Попов, Николай Вениаминович Максимов. 2. Москва: Издательство "ФОРУМ"; Москва: ООО "Научно-издательский центр ИНФРА-М", 2014. 448 с. ISBN 978-5-91134-833-5
- 5. Основы использования и проектирования баз данных: учеб. пособие / В.М. Илюшечкин. М.: Юрайт: ИД Юрайт, 2010. 213 с.
- 6. Базы данных: учеб. пособие для студентов вузов / А.В. Кузин, С.В. Левонисова. -4-е изд., стер. М.: Изд. центр «Академия», 2010. -314 с.
- 7. Компьютерные технологии в экономической науке и образовании: учебн. пособ.-Саратов: Изд-во Сарат. ин-та РГТЭУ, 2013.
- 8. Юрий Бекаревич, Нина Пушкина. Самоучитель Access 2007 Издательство: БХВ-Петербург, 2007 г. 724 с.
- 9. Гончаров А.Ю. ACCESS 2007 Самоучитель с примерами. Москва, Издательство «КУДИЦ-ПРЕСС», 2007, 296 с.
- 10. Кушнир А.Н. Access 2007 в кармане. Москва, Издательство "ЭКСМО" 2008 г. 272 с.
- 11. Создание баз данных в СУБД Microsoft Access: учеб. пособие / О.В. Брянцева, Е.А. Новикова. Саратов: Изд-во ГОУ ВПО «Сарат. гос. акад. Права», 2009. 183 с.
- 12. Информационные системы и базы данных. Организация и проектирование / В.Ю. Пирогов. СПб.: БХВ-Петербург, 2009. 528 с.
- 13. Базы данных. Язык SQL: для студента / В.В. Дунаев. 2-е изд., доп. и перераб. СПб.: БХВ-Петербург, 2007. 302 с.
- 14. Microsoft Access 2003: учеб. пособие / В.Д. Михеева, И.А. Харитонова; гл. ред. Е. Кондукова. СПб.: БХВ-Петербург, 2006. 1069 с.
- 15. Проектирование информационных систем: курс лекций: учеб. пособие для студентов вузов, обучающихся по специальностям в обл. информ. технологий / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. М.: Интернет-Ун-т Информ. Технологий, 2005. 298 с.

Учебное издание

Кабанов Владимир Федорович Бурмистров Александр Валерьевич

Основные принципы работы с базами данных

Учебное пособие для студентов Саратовского государственного университета

Оригинал-макет подготовил В.Ф. Кабанов

Компьютерная верстка *В.Ф. Кабанова* Издано в авторской редакции