

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

С.В. Одиночкина

**Разработка баз данных
в Microsoft Access 2010**

Учебное пособие



Санкт-Петербург

2012

УДК 004.655, 004.657, 004.62

С.В. Одиночкина

Разработка баз данных в Microsoft Access 2010 - СПб: НИУ ИТМО, 2012. – 83 с.

В пособии излагаются методические рекомендации к выполнению лабораторных работ по дисциплине «Введение в специальность».

Предназначено для студентов, обучающихся по всем профилям подготовки бакалавров направления: 210700 Инфокоммуникационные технологии и системы связи.

Рекомендовано к печати Ученым советом факультета Инфокоммуникационных технологий, протокол № 4 от 13 декабря 2011г.



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики».

© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2012

© С.В.Одиночкина, 2012.

Оглавление

Введение.....	4
Лабораторная работа №1: Таблицы.....	13
Лабораторная работа №2: Запросы	24
Лабораторная работа №3: Формы	37
Лабораторная работа №4: Отчеты.....	49
Лабораторная работа №5: Макросы.....	59
Лабораторная работа №6: Подготовка приложения к внедрению.....	67
Литература	70
Приложение 1. Описание таблиц.....	71
Приложение 2. Маска ввода.....	75
Приложение 3. Схема данных БД «Мой Сад»	78

Введение

В результате курса, проводимого под руководством преподавателя, студенты познакомятся с:

- технологиями и концепцией реляционных баз данных;
- принципами разработки структуры представления данных в реляционных базах данных;
- подходами к оптимизации работы с информацией средствами сортировки и фильтрации;
- реализацией запросов, форм и отчетов средствами Microsoft Access 2010;
- языком запросов SQL для обработки данных, метаданных, создания объектов базы данных;
- принципами разработки макросов для оптимизации работы с данными в Microsoft Office Access 2010;
- средствами обеспечения безопасности данных приложения, созданного в Microsoft Access 2010.

Цель курса

По окончании данного курса студенты смогут:

- понимать основные свойства базы данных Microsoft Access 2010;
- разбираться в компонентах, теоретических аспектах, терминологии;
- разрабатывать структуру таблиц;
- определять тип, размер и формат данных;
- обеспечивать целостность данных базы;
- импортировать и экспортировать информацию при работе с внешними источниками данных;
- оптимизировать работу с информацией средствами сортировки и фильтрации;
- формулировать различные виды запросов;
- работать с SQL-командой SELECT;
- реализовывать выборки данных из нескольких таблиц;
- создавать и использовать другие объекты базы данных: представления, индексы;
- создавать и модифицировать формы;
- формировать отчеты, используя групповые функции обработки данных;
- разрабатывать макросы;
- применять средства обеспечения безопасности данных приложения.

База данных Microsoft Access – это совокупность данных и объектов (т.е. форм, отчетов и т.п.), относящихся к определенной задаче и представляющих законченную систему. Базу данных Access составляют

таблицы, запросы, формы, отчеты, страницы доступа, макросы и модули. Кроме того, приложение Access содержит некоторые другие объекты, в том числе связи, свойства базы данных и спецификации импорта и экспорта.

Таблица – объект базы данных, в котором данные хранятся в виде записей (строк) и полей (столбцов). Является основным структурным элементом системы управления реляционной базой данных.

Запрос – объект базы данных, позволяющий осуществлять поиск и вывод данных, хранящихся в таблицах, удовлетворяющих заданным условиям (в том числе из нескольких таблиц). С помощью запроса можно модифицировать и удалять записи таблиц, а также выполнять различные вычисления.

Форма – объект базы данных, являющийся элементом пользовательского интерфейса, предназначенный для просмотра, ввода и модификации данных в одной или более таблицах.

Отчет – объект базы данных, предназначенный для анализа и вывода на печать данных, организованных и отформатированных в соответствии с требованиями пользователя.

Макрос – макрокоманда или набор макрокоманд, используемый для автоматизации задач.

Модуль – объект базы данных, который позволяет создавать библиотеки подпрограмм и функций, используемых во всем приложении. Используя коды модулей можно решать такие задачи, как обработка ошибок ввода, объявление и применение переменных, организация циклов и т.п.

Проектирование базы данных (БД) состоит из двух основных фаз: логического и физического моделирования. Во время фазы логического моделирования конструктор собирает требования и разрабатывает модель, не зависящую от конкретной системы управления базами данных (СУБД). Во время фазы физического моделирования конструктор создает модель, оптимизированную для конкретного приложения СУБД; именно эта модель реализуется на практике.

Процесс проектирования БД состоит из следующих этапов:

1. Сбор информации;
2. Идентификация объектов;
3. Моделирование объектов;
4. Идентификация типов информации для каждого объекта;
5. Идентификация отношений;
6. Нормализация;
7. Преобразование к физической модели;
8. Создание базы данных.

Этапы 1-6 образуют фазу логического моделирования. Этапы 7-8 представляют собой фазу физического моделирования.

На первом этапе проектирования базы данных необходимо определить назначение базы данных, режимы ее использования и основные алгоритмы, реализующие реальные бизнес-процессы – т.е. изучить предметную область ее использования с целью создания модели. При этом анализ поставленной задачи должен учитывать требования заказчика к разрабатываемой системе и опыт разработчика.

Идентификации подлежат все сущности, относящиеся к поставленной задаче (в данном случае – данные и объекты базы), а также связи между данными. В ходе идентификации определяются атрибуты (свойства) сущностей, и для этого необходимо принять решения по следующим вопросам:

- Какие значения должны содержаться в поле;
- Сколько места необходимо для хранения значений в поле;
- Какие операции должны производиться со значениями в поле;
- Нужна ли сортировка данных поля;
- Необходимо ли группировать данные.

В *Таблице 1* приведен список предусмотренных в приложении Microsoft Office Access 2010 типов данных, инструкции по их применению и сведения о размере места, необходимого для хранения данных каждого типа. При этом необходимо помнить, что максимальный размер файла базы данных Office Access 2010 составляет 2 гигабайта.

Таблица 1. Типы данных MS Access

Тип данных	Применение	Размер
Текстовый	Используется для буквенно-цифровых символов, включая текст, а также текст и числа, не применяемые в вычислениях.	До 255 символов
Поле МЕМО	Используется для текста, размер которого превышает 255 символов, или для текста, в котором используется RTF-форматирование. Свойство поля МЕМО указывает, поддерживает ли поле форматированный текст.	До 1024 x 1024 x 1024 символов, для хранения которых требуется 2 гигабайта (2 байта на символ). Из них можно отображать до 65 535 символов в одном элементе управления.
Числовой	Применяется для хранения числовых значений (целых или дробных), которые используются в вычислениях (за исключением денежных сумм).	1, 2, 4, 8 или 12 байтов (16 байтов, когда поле используется для кода репликации).
Дата/время	Используется для хранения значений дат и времени.	8 байтов

Денежный	Используется для хранения денежных значений. Используется для предотвращения округлений во время вычислений.	8 байтов
Счетчик	Используется для формирования уникальных значений, которые могут применяться в качестве первичного ключа. Эти значения автоматически вставляются в поле при добавлении записи. Поля с типом данных "Счетчик" могут формироваться добавлением единицы, добавлением заданного значения или с помощью случайных чисел.	4 байта (16 байтов, когда поле используется как код репликации)
Логический	Используется для логических значений: Да/Нет, Истина/Ложь или Вкл/Выкл.	1 бит (0,125 байта)
Поле объекта OLE	Используется для хранения объектов OLE из других программ Microsoft Windows.	До 1 гигабайта
Вложение	Используется для хранения двоичных файлов (файлов, которые нельзя прочесть с помощью текстового редактора), таких как цифровые изображения (фотографии и другие изображения) или файлов, созданных с помощью других приложений Microsoft Office. Используя поле с типом данных "Вложение", можно вложить в одну запись более одного файла.	2 гигабайта для сжатых вложений. Для несжатых вложений приблизительно 700 КБ, в зависимости от степени сжатия вложений.
Гиперссылка	Используется для хранения гиперссылок (в том числе ссылок на объекты приложения Access, которые хранятся в базе данных).	До 1024 x 1024 x 1024 символов, для хранения которых требуется 2 гигабайта (2 байта на символ). Из них можно отображать до 65 535 символов в одном элементе управления.
Мастер подстановок	Используется для запуска мастера подстановок, позволяющего	Если к полю подстановок

	создавать поле, в котором в виде раскрывающегося списка отображаются значения из другой таблицы, запроса или списка значений (фактически мастер подстановок не является типом данных).	присоединена таблица или запрос, то это размер присоединенного столбца. Если к полю подстановок не присоединен другой столбец (т. е. хранится список значений), то это размер текстового поля, используемого для хранения списка.
--	--	---

В большинстве систем управления базами данных, включая Microsoft Access версий до Microsoft Access 2007, в поле можно хранить только одно значение. При работе с Microsoft Office Access 2010 можно создать поле, хранящее несколько значений, например список категорий, к которым относится проблема.

Технически модель поля, одновременно допускающего несколько значений, в Access реализует отношение «многие-ко-многим».

Рассмотреть необходимость создания поля, одновременно допускающего несколько значений, следует в случаях, когда требуется выполнить следующие задачи:

- Хранить выборку из списка вариантов, содержащую одновременно несколько значений (и этот список относительно невелик).
- Экспортировать таблицу Access на узел SharePoint, где будут применяться поля вариантов или подстановок с несколькими значениями, доступные в Windows SharePoint Services.
- Создать связь со списком SharePoint, содержащим поля вариантов или подстановок с несколькими значениями.

Кроме описанных выше ситуаций, поля, одновременно допускающие несколько значений, можно использовать, когда есть уверенность, что не придется позже переносить базу данных на Microsoft SQL Server. Поля, одновременно допускающие несколько значений, из формата Office Access 2010 преобразуются в поля SQL Server в виде полей типа MEMO (ntext), которые содержат значения с разделителями. Поскольку SQL Server не поддерживает типы данных, одновременно допускающие несколько значений и представляющие модель отношения «многие-ко-многим», может потребоваться дополнительная работа по конструированию и преобразованию.

В *Таблице 2* перечислены свойства полей и описано влияние этих свойств на поля в зависимости от их типа данных.

Таблица 2. Свойства полей MS Access

Свойство поля	Применение свойства
Размер поля	Задание максимального размера для данных.
Формат поля	Настройка отображения поля по умолчанию на экране и при печати.
Число десятичных знаков	Задание числа десятичных знаков, используемых при отображении чисел.
Новые значения	Указывает, какие значения (последовательные или случайные) присваиваются полю "Счетчик" при добавлении новой записи.
Маска ввода	Символы редактирования, определяющие способы ввода данных.
Подпись	Текст, отображаемый по умолчанию в качестве подписи в формах, отчетах и запросах.
Значение по умолчанию	Значение, автоматически назначаемое полю при добавлении новой записи.
Условие на значение	Выражение, значение которого должно быть истинно при добавлении или изменении значения поля.
Текст для проверки	Текст, появляющийся при вводе значения, при котором нарушается условие, указанное в поле Условие на значение .
Обязательное поле	Указывает, должно ли это поле содержать значение в каждой записи.
Пустые строки	Разрешает (установкой значения Да) ввод строки нулевой длины ("") в текстовое поле и в поле МЕМО.
Индексация	Посредством создания и использования индекса ускоряет доступ к этому полю для чтения.
Сжатие Юникод	Выполняется сжатие данных, содержащихся в этом поле, если в нем менее 4096 символов (это условие всегда выполняется для текстовых полей). Если в поле содержится более 4096 символов, не выполняется никаких действий.
Режим IME	Управление преобразованием символов в восточноазиатских версиях Windows.
Режим предложений IME	Управление преобразованием предложений в восточноазиатских версиях Windows.
Смарт-теги	Позволяет выбрать смарт-теги, применяемые к полю.
Только добавление	Когда для свойства задано значение Да , ведется журнал значения поля.
Формат текста	Для хранения данных в виде HTML-кода и использования RTF-форматирования следует

	выбрать значение Формат RTF . Чтобы хранить только текст без форматирования, следует выбрать значение Обычный текст .
Выравнивание текста	Выравнивание текста по умолчанию в элементе управления.
Отображать элемент выбора даты	Указывает, должен ли в приложении Access отображаться элемент выбора даты (элемент управления "Календарь"), когда пользователь изменяет значение поля.

Один из способов изучить определенную базу данных — воспользоваться архивариусом (средством документирования базы данных). Архивариус используется для построения отчета, содержащего подробные сведения об объектах в базе данных. Сначала следует выбрать, какие объекты будут подробно рассмотрены в отчете. Когда запускается архивариус, его отчет содержит все данные о выбранных объектах базы данных.

В теории реляционных баз данных таблица представляет собой изначально неупорядоченный набор записей. Единственный способ идентифицировать определённую запись в этой таблице — это указать набор атрибутов, который был бы уникальным для этой записи.

Ключом называется набор атрибутов, однозначно определяющий запись. Существуют следующие виды ключей:

- Первичный ключ – представляет собой одно или несколько полей (столбцов), значения которых однозначно определяют каждую запись в таблице. Первичный ключ не допускает значений Null и всегда должен иметь уникальный индекс. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах. Первичный ключ может быть *естественным* или *искусственным*. Ключ, состоящий из информационных полей таблицы (т. е. полей, содержащих полезную информацию об описываемых объектах) называется естественным ключом. Теоретически, естественный ключ всегда можно сформировать, в этом случае он носит название *«интеллектуальный ключ»*. Искусственный ключ - это дополнительное служебное поле, единственное предназначение которого - служить первичным ключом. Значения этого поля не образуются на основе каких-либо других данных из БД, а генерируются искусственно. Как правило, суррогатный ключ - это просто числовое поле, в которое заносятся значения из возрастающей числовой последовательности.
- Внешний ключ - представляет собой одно или несколько полей (столбцов), содержащих ссылку на поле или поля первичного ключа в другой таблице. Внешний ключ определяет способ связи таблиц.

Ключи также делятся на два класса: простые и составные.

Простой ключ состоит из одного атрибута, составной ключ состоит из нескольких атрибутов. Применение составных ключей усложняет объединение таблиц.

Индекс - средство, ускоряющее поиск и сортировку данных в таблице. Существенное повышение скорости выполнения запросов приносит индексирование полей, расположенных по обе стороны отношения, или создание связи между этими полями, а также индексирование всех полей, используемых для задания условий отбора в запросе. Индекс может быть простым (состоять из одного атрибута) или составным (состоять из нескольких атрибутов). Индекс может быть уникальным или не уникальным.

Реляционные базы данных позволяют объединять информацию, принадлежащую разным сущностям базы данных. Отношения определяются в процессе проектирования базы; для этого следует проанализировать разные таблицы, выявить логические связи, существующие между ними, и охарактеризовать выявленные связи.

После создания таблицы для каждой темы в базе данных нужно предоставить приложению Microsoft Access 2010 средства, с помощью которых можно будет вновь объединять сведения при необходимости. Это делается путем помещения общих полей в связанные таблицы и определения связей между таблицами. После этого можно создавать запросы, формы и отчеты, одновременно отображающие сведения из нескольких таблиц.

Межтабличные связи могут объединять две и более сущности. Как правило, они соответствуют некоторому взаимодействию между сущностями и описывают связь, возникающую между ними. Во время логического проектирования связи между таблицами могут обладать собственными атрибутами. Такое отношение выделяется в отдельную сущность типа связь.

Отношения делятся на три основных типа, в зависимости от количества записей сущности, связанных с записью другой сущности:

Один к одному: каждой записи первой сущности соответствует только одна запись второй сущности, а каждой записи второй сущности - только одна запись первой сущности. Пример - автор, у которого в данный момент имеется лишь одна незавершенная книга.

Один ко многим: каждой записи первой сущности могут соответствовать несколько записей второй сущности, однако каждой записи второй сущности соответствует только одна запись первой сущности. Пример - издательство, выпустившее несколько книг.

Многие ко многим: каждой записи первой сущности могут соответствовать несколько записей второй сущности, а каждой записи второй сущности соответствуют несколько записей первой сущности.

Пример - один автор может написать несколько книг, а у одной книги может быть несколько авторов.

В реляционных базах данных этот тип отношений не реализуем, поэтому создается дополнительная сущность, ассоциирующая данную связь (ассоциация).

Связи между сущностями устанавливаются по равенству значений первичного и внешнего ключей.

В Microsoft Access межтабличные связи можно создать непосредственно с помощью окна «Схема данных» или путем перетаскивания поля из области **Список полей**. В приложении Access межтабличные связи используются для того, чтобы продемонстрировать, как связать таблицы для использования их в объекте базы данных. Существует несколько причин для создания межтаблических связей перед созданием других объектов базы данных (форм, запросов и отчетов):

- Межтабличные связи предоставляют сведения для структуры запросов;
- Межтабличные связи предоставляют сведения для структуры форм и отчетов;
- Межтабличные связи являются основой, с помощью которой можно обеспечить целостность данных.

Целостность данных означает систему правил, используемых в Microsoft Access для поддержания связей между записями в связанных таблицах, а также обеспечивающих защиту от случайного удаления или изменения связанных данных. Установить целостность данных можно, если выполнены следующие условия:

- Связанное поле главной таблицы является первичным ключом или альтернативным ключом (имеет уникальный индекс).
- Связанные поля (первичный ключ главной таблицы и внешний ключ подчиненной таблицы) имеют один тип данных. Здесь существует исключение: поле счетчика может быть связано с числовым полем, свойство которого **Размер поля (FieldSize)** имеет значение **Длинное целое**.
- Обе таблицы принадлежат одной базе данных Microsoft Access. Для установки целостности данных база данных, в которой находятся таблицы, должна быть открыта.

При установке целостности данных, необходимо помнить следующие правила:

- Невозможно ввести в поле внешнего ключа связанной таблицы значение, не содержащееся в ключевом поле главной таблицы.
- Не допускается удаление записи из главной таблицы, если существуют связанные с ней записи в подчиненной таблице.
- Невозможно изменить значение первичного ключа в главной таблице, если существуют записи, связанные с данной записью.

Чтобы задать правила целостности данных для конкретной связи, при ее создании в Microsoft Office Access следует установить флажок

Обеспечение целостности данных. Если данный флажок установлен, то любая попытка выполнить действие, нарушающее одно из перечисленных выше правил, приведет к выводу на экран предупреждения, а само действие будет отменено.

Чтобы преодолеть ограничения на удаление или изменение связанных записей, сохраняя при этом целостность данных, следует установить флажки **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**. Если установлен флажок **Каскадное обновление связанных полей**, то при изменении ключевого поля главной таблицы автоматически изменяются и соответствующие значения связанных записей. Если установлен флажок **Каскадное удаление связанных полей**, то при удалении записи в главной таблице удаляются и все связанные записи в подчиненной таблице.

После создания необходимых таблиц, полей и связей необходимо еще раз просмотреть структуру базы данных и выявить возможные недочеты. Желательно это сделать на данном этапе, пока таблицы не заполнены данными.

В Microsoft Access существует два инструмента, помогающих в усовершенствовании структуры баз данных:

- *Мастер анализа таблиц* может проанализировать структуру таблицы, предложить подходящие новые структуры и связи, а также разделить таблицу на новые связанные таблицы, если это имеет смысл.
- *Анализатор быстрой работы* исследует всю базу данных, дает рекомендации по ее улучшению, а также осуществляет их.

Лабораторная работа №1: Таблицы

Создание базы данных

1. Запустите Microsoft Access из меню **Пуск** или с помощью ярлыка. Отобразится меню создания базы данных Access.
2. В окне раздела «Создать» в разделе **Доступные шаблоны** выберите команду **Новая база данных (Blank Database)**.
3. В области **Новая база данных (Blank Database)** в поле **Имя файла (File Name)** введите имя файла («МойСад») и укажите путь к папке, в которую он должен быть сохранен.
4. Нажмите кнопку **Создать (Create)**.

Будет создана новая база данных и открыта новая таблица в режиме таблицы. При необходимости новую базу данных можно создать с помощью шаблонов, а кроме того, существует возможность загрузки дополнительных шаблонов с веб-узла Microsoft Office Online.

Создание таблиц

При создании базы данных данные сохраняются в таблицах — списках строк (записей) и столбцов (полей), относящихся к конкретной области.

При создании новой базы данных автоматически создается новая таблица. Также таблицу можно создать с помощью добавления новой таблицы в существующую базу данных, импорта или создания ссылки на таблицу из другого источника данных, такого как книга Microsoft Office Excel, текстовый файл или другая база данных.

Создание таблицы в режиме конструктора

Задание: в режиме конструктора создайте таблицы «Заказы», «Доставка», «Поставщики», «Фактический_Адрес_Доставки» (см. Приложение 1).

1. В окне БД выберите вкладку **Создание (Create)**, на ней в группе **Таблицы (Tables) – Конструктор таблиц (Table Design)**.
2. В графу **Имя поля (Field Name)** введите имя формируемого поля – ENTER.
3. В графе **Тип данных (Data Type)** выберите из списка требуемый тип данных.
4. В графу **Описание (Description)** введите текст, поясняющий назначение и использование данного поля таблицы.
5. В секции **Свойства поля (Field Properties)** задайте требуемые свойства поля:

Размер поля (FieldSize);	Условие на значение (Validation
Формат поля (Format);	Rule);
Маску ввода (Input Mask);	Обязательность поля (Required)
	и т.д.

6. Формирование маски ввода (Input Mask) (см. Приложение 2)
Маска ввода определяет способы ввода данных в базу данных.
 - 6.1 Выберите строку **Маска ввода (Input Mask)** в секции **Свойства поля (Field Properties)** и нажмите кнопку **Построить (Input Mask Wizard)** (кнопка с тремя точками).
 - 6.1. Подтвердите сохранение таблицы.
 - 6.2. В открывшемся окне **Мастер создания масок (Input Mask Wizard)** выберите из предлагаемого списка необходимую маску (если она есть в списке).

6.3. Если необходимой маски в списке нет, нажмите в открытом окне **Мастера создания масок** кнопку **Список**. Откроется окно **Настройка масок ввода** (рис. 1.1).

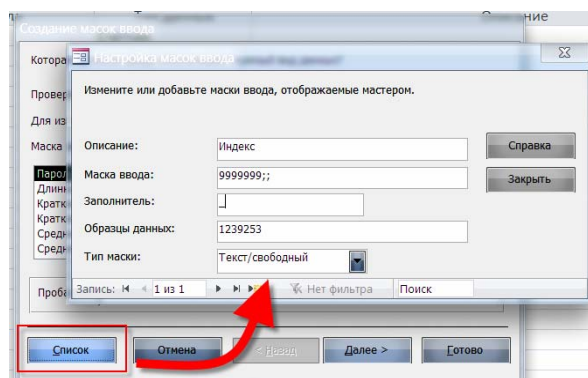


Рис. 1.1 Окна работы с маской ввода.

6.4. В поле **Описание** введите название создаваемой маски ввода; в поле **Маска ввода** введите собственно саму маску (см. Приложение 2); в поле **Заполнитель** выберите заполнитель символов маски (по умолчанию – знак нижнего подчеркивания); введите образец данных в соответствующее поле (в качестве примера для пользователя) и выберите **Тип маски**. Нажмите кнопку **Заккрыть**.

6.5. В окне **Мастера создания масок** появится только что созданная маска – **Далее**. В поле **Проба (Try It)** введите данные для проверки действия маски – **Далее (Next) - Далее (Next) – Готово (Finish)**.

6.6. В случае необходимости маска может быть построена без помощи мастера.

7. Определение первичного ключа

7.1. В режиме **Конструктора (Design)** с помощью указателя мыши выделите строку с описанием поля, которое должно использоваться в качестве ключа.

7.2. Нажмите на панели инструментов кнопку **Ключевое поле (Primary Key)** или нажмите правую кнопку мыши, выбрав соответствующий пункт меню.

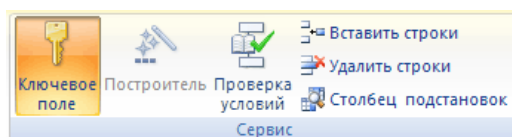


Рис. 1.2. Группа Сервис.

7.3. При определении составного ключа, необходимо выделить сразу все поля, входящие в состав ключа (например, удерживая клавишу Ctrl), и назначить их ключевыми аналогичным образом (п. 7.2).

8. Формирование «Значения по умолчанию» (Default Value)

- 8.1. В таблице «Заказы» для поля «ДатаЗаказа» в секции **Свойства поля** (Field Properties) выберите строку **Значение по умолчанию (Default Value)**.
- 8.2. С помощью **построителя выражений (...)** (Expression Builder) (**Функции (Function) – Встроенные функции (Built-In Function) – Дата/Время (Date/Time)**) постройте выражение Date(). После этого при формировании нового заказа в поле по умолчанию будет отображаться текущая дата.
9. Сохранение структуры таблиц
Существуют несколько вариантов сохранения структуры таблицы:
 - 9.1. Нажать кнопку **Сохранить (Save)** на **Панели быстрого доступа**.
 - 9.2. Кликнуть правой кнопкой мыши на вкладке документа и выбрать в контекстном меню команду **Сохранить (Save As)**.
 - 9.3. Выбрать вкладку **Файл** (представление Backstage), а затем выбрать команду **Сохранить как (Save As)**. Введите имя созданной таблицы, нажмите **ОК**, закройте окно.

Создание таблицы в режиме таблицы

Задание: в режиме таблицы создайте таблицы «Сотрудники», «Транспортная Компания» и «Подробности Заказа» (см. Приложение I).

1. На ленте выберите вкладку **Создание (Create)**, группу инструментов **Таблицы (Tables)**, кнопку **Таблица (Table)**. В результате будет сформирована новая пустая таблица.
2. Щелкните на новом поле - откроется список основных типов полей. Выберите тип создаваемого поля и присвойте ему имя. Для просмотра всех возможных типов полей на вкладке **Поля** в группе **Добавление и удаление** выберите кнопку **Другие поля**.
3. Перейдите в режим **конструктора (Design)**, присвоив имя таблице и сохранив ее.
4. В режиме конструктора установите необходимые свойства полей таблицы.
5. Сохраните структуру таблицы.

Импорт таблицы из внешней базы данных MS Access

Задание: импортируйте в базу данных таблицу «Товары» из внешней базы данных «Товары.accdb».

1. Выберите вкладку **Внешние данные (External Data)**, группу **Импорт и связи**, нажмите кнопку **Access**.

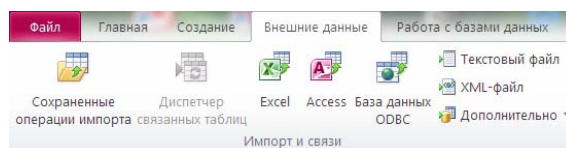


Рис. 1.3. Группа Импорт и связи

2. Укажите источник данных (путь к папке, где хранится БД Товары.accbd). Переключатель должен быть установлен на категории «Импорт таблиц, запросов, отчетов, макросов и модулей в текущую базу данных (Import tables, queries, forms, reports, macros, and modules into a New Table in the Current Database)». Нажмите **ОК**.
3. В открывшемся окне выберите нужную для импорта таблицу (Товары) – **ОК**.
4. Сохраните структуру таблицы, не сохраняя шаги импорта.

Импорт таблицы из MS Office Excel

Задание: импортируйте в базу данных таблицу «Клиенты» из файла «Клиенты.xlsx».

1. Выберите вкладку **Внешние данные (External Data)**, группу **Импорт и связи**, нажмите кнопку **Excel**.
2. Укажите источник данных (путь к папке, где хранится файл Клиенты.xlsx). Переключатель должен быть установлен на категории «Импортировать данные источника в новую таблицу в текущей базе данных (Import the Source Data into a New Table in the Current Database)». Нажмите **ОК**.
3. Выберите объект «Листы (Show Worksheets)» - **Далее (Next)**.
4. Установите флаг «Первая строка содержит заголовки столбцов (First Row Contains Column Headings)» - **Далее (Next)**.
5. Оставьте описание полей без изменений – **Далее (Next)**.
6. Выберите в группе переключателей пункт «**Определить ключ**» – и из выпадающего списка полей таблицы выберите поле «КодКлиента» - **Далее (Next)**.
7. Подтвердите импорт в таблицу «Клиенты» - **Готово (Finish)**.
8. В **режиме конструктора** отредактируйте типы данных полей таблицы (см. Приложение 1).
9. Сохраните структуру таблицы, не сохраняя шаги импорта.

Импорт данных из текстового файла (.txt)

Задание: импортируйте в базу данных таблицу «Категории Товаров» из файла «Категории Товаров.txt».

1. Выберите вкладку **Внешние данные (External Data)**, группу **Импорт и связи**, нажмите кнопку **Текстовый файл (Text File)**.
2. Укажите источник данных (путь к папке, где хранится файл Категории Товаров.txt). Переключатель должен быть установлен на категории «Импортировать данные источника в новую таблицу в текущей базе данных». Нажмите ОК.
3. Переключатель «с разделителями (Delimited) – поля передаются запятыми или табуляцией» - **Далее (Next)**.
4. Переключатель «точка с запятой», установить флаг «Первая строка содержит имена полей (First Row Contains Field Names)» - **Далее (Next)**.
5. Оставьте описание полей без изменений – **Далее (Next)**.
6. Выберите в группе переключателей пункт «**Определить ключ**» и из выпадающего списка полей таблицы выберите «*КодКатегории*» – **Далее (Next)**.
7. Подтвердите импорт в таблицу «*Категории_Товаров*» - **Готово (Finish)**.
8. Сохраните структуру таблицы, не сохраняя шаги импорта.

Заполнение таблиц с помощью импорта данных

Задание: заполните данными все оставшиеся таблицы, импортируя из соответствующих файлов (формата .xlsx или .txt) с одним исключением: **импортируемые данные должен сохраняться в существующие таблицы**, а не новые (переключатель должен быть установлен в пункте «Добавить копию записей в конец таблицы (Append a copy of the records to the table)», далее - указать имя соответствующей таблицы). Помните, что последовательность и названия полей в таблице Access должны соответствовать полям из импортируемых файлов (последовательность и названия полей таблиц – *Приложение 1*).

При импорте данных для каждой таблицы ключ необходимо определить по первому полю в выпадающем списке.

Работа с вычисляемыми полями

В Access 2010 можно использовать тип данных "Вычисляемый" для создания вычисляемого поля в таблице. Это значительно упрощает отображение и использование результатов вычисления в базе данных. Access обновляет вычисляемые поля при редактировании записи, обеспечивая правильность значения в поле.

Задание: создать вычисляемое поле «*ТоваровНаСумму*» в таблице «*Товары*», хранящее информацию о том, на какую сумму имеется каждого наименования товара в данный момент на складе.

1. Откройте таблицу «**Товары**» в режиме конструктора (**Design**). Добавьте поле с именем «**ТоваровНаСумму**» и типом данных «**Вычисляемый**». Откроется окно **Построителя выражений (Expression Builder)**.
2. В **Построителе выражений (Expression Builder)** двойным щелчком мыши выберите поля, участвующие в вычислении значения поля «**ТоваровНаСумму**»: поле «**ЦенаЗаУпаковку**» и поле «**Кол-воНаСкладе**». Знак умножения поставьте вручную или выберите в группе **Элементы выражений** категории **Арифметические**. Нажмите **ОК**.

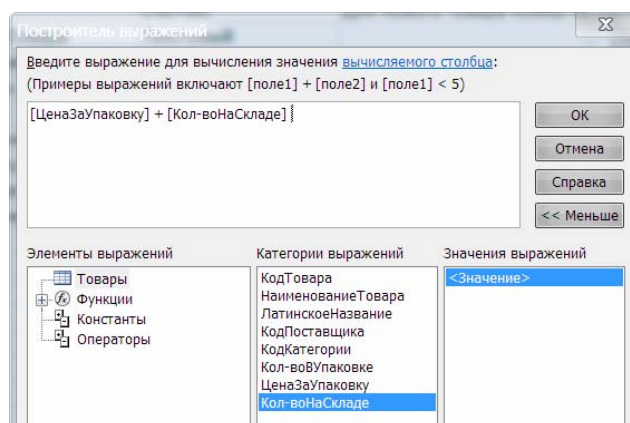



Рис.1.4. Окно построителя выражений.

3. Сохраните изменения, перейдите в режим таблицы и убедитесь в корректности результатов.

Вложение файлов в записи базы данных (поле «Вложение» (Attachment))

Поле **Вложение (Attachment)** предназначено для хранения нескольких файлов в одном поле и предназначено для замены своего предшественника - типа поля OLE. Вложения могут относиться к разным типам.

Задание: добавить поле типа **Вложение** для таблицы «**Сотрудники**» и поместить в это поле имеющиеся дополнительные материалы – резюме каждого сотрудника (находятся в папке «**Резюме**») и имеющиеся фотографии сотрудников (папка «**ФотографииСотрудников**»).

1. Откройте таблицу «**Сотрудники**» в режиме конструктора (**Design**). Добавьте поле с именем «**Приложения**» и типом данных «**Вложение**».
2. Перейдите в режим таблицы – появится только что созданное поле 
3. Для добавления информации к записи нажмите правой кнопкой мыши на соответствующем поле, выберите пункт меню **Управление вложениями** - откроется диалоговое окно **Вложения (Attachment)**.

оно не отображается, на вкладке **Конструктор (Design)** в группе **Связи (Relationships)** нажмите кнопку **Отобразить таблицу (Show Table)**.

4. Выберите одну или несколько таблиц или запросов и затем нажмите кнопку **Добавить(Add)**. По завершении добавления таблиц и запросов в окне «Схема данных» нажмите кнопку **Заккрыть(Close)**.
5. Перетащите поле (как правило, первичный ключ) из одной таблицы в общее поле (внешний ключ) в другой таблице. Чтобы перетащить сразу несколько полей, нажав и удерживая клавишу Ctrl, выберите каждое поле, а затем перетащите их. Будет открыто диалоговое окно **Изменение связей(Edit Relationships)**.

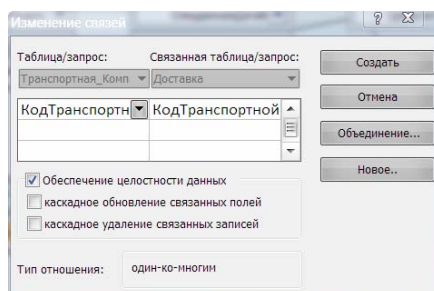


Рис. 1.6. Окно Изменение связей.

6. Убедитесь, что имена общих полей связи отображены правильно. Если имя поля отображено неверно, выберите нужное имя из списка. Чтобы обеспечить целостность данных для этой связи, установите флажок **Обеспечение целостности данных (Enforce Referential Integrity)**. Чтобы преодолеть ограничения на удаление или изменение связанных записей, сохраняя при этом целостность данных, следует установить флажки **Каскадное обновление связанных полей (Cascade Update Related Fields)** и **Каскадное удаление связанных полей (Cascade Delete Related Records)**. Если установлен флажок **Каскадное обновление связанных полей (Enforce Referential Integrity)**, то при изменении ключевого поля главной таблицы автоматически изменяются и соответствующие значения связанных записей. Если установлен флажок **Каскадное удаление связанных полей (Cascade Delete Related Records)**, то при удалении записи в главной таблице удаляются и все связанные записи в подчиненной таблице.
7. Нажмите кнопку **Создать (Create)**.
8. Для удаления или изменения связи поместите курсор на линию связи, а затем щелкните линию для выбора. Выделенная линия связи станет жирной. Когда линия связи будет выбрана, нажмите правую кнопку мыши и выберите необходимый пункт меню.
9. Сохраните структуру схемы данных и закройте ее.

Использование мастера подстановок и создание полей с несколькими значениями

Задание: работая с таблицей «Заказы» (поле «КодСотрудника») с помощью мастера подстановок создать поле с несколькими значениями.

1. Для возможности внесения изменения в типы полей сначала необходимо открыть **Схему данных (Relationships)** и удалить все связи той таблицы, которую необходимо модифицировать.
2. Откройте таблицу «Заказы» в режиме **конструктора (Design)**. Рядом с именем поля «КодСотрудника» выберите из выпадающего меню тип данных **Мастер подстановок (Lookup Wizard)**.
3. Используйте значение из таблицы. В **мастере подстановок** установите переключатель **Объект «Столбец подстановки» будет использовать значения из таблицы или запроса (I want the lookup column to look up the values in a table or query)** и нажмите кнопку **Далее (Next)**.
4. Выберите в списке таблицу «Сотрудники» и нажмите кнопку **Далее (Next)**.
5. В списке **Доступные поля (Available Fields)** выберите поля «Фамилия», «Должность».
6. Щелкните значок «больше» (>), чтобы переместить выбранные поля в список **Выбранные поля (Selected Fields)** – **Далее (Next)**.
7. Можно выделить одно из полей, чтобы сортировать по нему элементы для подстановки, и нажмите кнопку **Далее (Next)**. Нажмите кнопку **По возрастанию**, чтобы переключиться в режим сортировки по убыванию. Надпись на кнопке изменится на **По убыванию**. Нажмите кнопку еще раз, чтобы переключиться обратно. Нажмите кнопку **Далее (Next)**.
8. При необходимости отрегулируйте ширину столбцов в поле подстановок и нажмите кнопку **Далее (Next)**.
9. В группе **Сохранить несколько значений для этой подстановки?** установите флажок **Разрешить несколько значений (Allow Multiple Values)** (этот флажок должен быть установлен, чтобы разрешить хранение одновременно нескольких значений) – **Готово (Finish)**.
10. Сохраните изменения.
11. Перейдите в **режим таблицы (Datasheet)** и измените любую запись в поле **Код сотрудника** таблицы «Заказы», кликнув по соответствующему полю с данными. Обратите внимание, что при работе с полем появляется выпадающий список сотрудников (фамилия и должность), в котором вы можете выбрать одного или нескольких сотрудников для обработки заказа, установив маркер напротив их фамилий в списке.
12. Перейдите в режим **конструктора (Design)** и убедитесь, что тип данных поля «КодСотрудника» остался числовой (это важно при связи таблиц).
13. Сохраните изменения в таблице.

14. Откройте **Схему данных (Relationships)** и восстановите связи таблицы **«Заказы»**. Обратите внимание, что при отображении таблицы **«Заказы»** на схеме произошли изменения.
15. Сохраните схему данных.
16. Аналогичным образом с помощью мастера подстановок отредактируйте в таблице **«Товары»** поле **«КодКатегории»** с одним исключением: поле должно иметь только одно значение (т.е. несколько значений в поле быть не может). Для подстановки необходимо взять из таблицы **«Категории_Товаров»** поле **«Название Категории»**. После проделанной работы убедитесь, что при редактировании данных в поле таблицы появляется выпадающий список, не допускающий выбора нескольких значений. Также убедитесь (в режиме конструктора(**Design**)), что тип поля отсталая числовой.
17. Сохраните изменения в таблице, закройте таблицу, восстановите связи в **Схеме данных (Relationships)**, сохраните **Схему данных** (изменений в отображении таблицы **«Товары»** на схеме данных быть не должно, т.к. нет возможности выбора нескольких значений в поле).

Просмотр сведений об объектах в базе данных с помощью архивариуса (средства документирования базы данных)

Один из способов изучить определенную базу данных — воспользоваться архивариусом. Архивариус используется для построения отчета, содержащего подробные сведения об объектах в базе данных. Архивариус также является инструментом для документирования базы данных. Сначала следует выбрать, какие объекты будут подробно рассмотрены в отчете. Когда запускается архивариус, его отчет содержит все данные о выбранных объектах базы данных.

1. На вкладке **Работа с базами данных (Database Tools)** в группе **Анализ (Analysis)** щелкните **Архивариус (Средство документирования базы данных) (Database Documenter)**.
2. В диалоговом окне **Архивариус (Database Documenter)** щелкните вкладку, на которой представлен тот тип объекта базы данных, который нужно изучить. Чтобы создать отчет по всем объектам в базе данных, щелкните вкладку **Все объекты (All Object Types)**.
3. Выберите один или несколько объектов из перечисленных на вкладке. Чтобы выбрать все объекты на вкладке, щелкните **Выделить все (Select All)**.
4. Щелкните **ОК**.
5. Изучите отчеты, закройте окно с отчетами.

Сохранение и закрытие базы данных

1. Сохраните все изменения.
2. Завершите работу с Microsoft Access.

Лабораторная работа №2: Запросы

Запрос представляет собой обращение к данным для получения информации и выполнения действий с данными. Запрос можно использовать для просмотра, изменения и анализа данных, для выполнения расчетов, объединения данных из разных таблиц или добавления, изменения или удаления данных в таблице. Запросы можно также использовать для включения данных в форму или отчет. В хорошо структурированной базе данных сведения, которые требуется представить с использованием формы или отчета, зачастую хранятся в разных таблицах. С помощью запроса можно собрать необходимые данные перед проектированием формы или отчета.

В Microsoft Access есть несколько типов запросов:

1. **Запросы на выборку** - наиболее часто используемый тип запроса. Запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы, запись в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.
2. **Запросы с параметрами** - это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные, например условие для возвращения записей или значение, которое требуется вставить в поле. Можно разработать запрос, выводящий приглашение на ввод нескольких единиц данных, например двух дат. Затем Microsoft Access может вернуть все записи, относящиеся к интервалу времени между этими датами. Запросы с параметрами также удобно использовать **в качестве основы для форм, отчетов и страниц доступа к данным.**
3. **Перекрестные запросы** - запросы, которые используются для расчетов и представления данных в структуре, облегчающей их анализ. Перекрестный запрос подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк.
4. **Запросы на изменение** - запрос, который за одну операцию изменяет или перемещает несколько записей. Существует четыре типа запросов на изменение.
 - 4.1. **На удаление записи** - удаляет группу записей из одной или

нескольких таблиц. С помощью запроса на удаление можно удалять только всю запись, а не отдельные поля внутри нее.



- 4.2. **На обновление записи** - вносит общие изменения в группу записей одной или нескольких таблиц. Запрос на обновление записей позволяет изменять данные в существующих таблицах.
- 4.3. **На добавление записей** - добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц.
- 4.4. **На создание таблицы** - создает новую таблицу на основе всех или части данных из одной или нескольких таблиц. Запрос на создание таблицы полезен при создании таблицы для экспорта в другие базы данных Microsoft Access или при создании архивной таблицы, содержащей старые записи.
- 4.5. **На объединение** – позволяет объединить данных из двух таблиц с аналогичными структурами.

Запросы можно создавать с помощью **Мастера запросов (Query Wizard)** или **Конструктора запросов (Query Design)**.

Однотабличные запросы-выборки

Создание запроса, выполняющего сортировку (просмотр подмножества данных в таблице)

Задание: необходимо просмотреть список товаров и цен на эти товары.


1. Откройте базу данных «Мой Сад».
2. На вкладке **Создание (Create)** в группе **Запросы (Queries)** щелкните **Конструктор запросов (Query Design)**.
3. В диалоговом окне **Добавление таблицы (Show Table)** на вкладке **Таблицы (Tables)** дважды щелкните «**Товары**».
4. Закройте диалоговое окно **Добавление таблицы (Show Table)**.
5. В таблице «**Товары**» дважды кликните кнопку мыши на полях «**НаименованиеТовара**» и «**ЦенаЗаУпаковку**» (также добавить поле в бланк запроса можно перетаскив поле из таблицы, удерживая нажатой кнопку мыши).
6. В строке **Сортировка (Sort)** бланка запроса для поля «**ЦенаЗаУпаковку**» выберите **По возрастанию (Ascending)** (для того, чтобы товары в запросе выводились по возрастанию цены на них).
7. Проверьте результат, запустив запрос кнопкой **Выполнить (Run)**  на панели инструментов.
8. Сохраните запрос под именем «**Товары_И_Цены**» (кликнув правой кнопкой мыши на вкладке с названием запроса и выбрав пункт меню **Сохранить** или нажав кнопку **Сохранить** на ленте .

Запрос, выполняющий отбор значений в определенном диапазоне


1. Откройте запрос «*Товары_И_Цены*» в режиме конструктора (Design).
2. Для поля «*НаименованиеТовара*» установите значение строки **Сортировка (Sort) - По возрастанию (Ascending)**.
3. Установите курсор в строке **Условия отбора (Criteria)** поля «*НаименованиеТовара*» и вызовите **Построитель выражений (Expression Builder)** одним из способов: Ctrl+F2 или нажатием кнопки **Построителя** на панели инструментов в группе **Настройка запроса (Query Setup)**.



Построитель выражений (Expression Builder) предоставляет удобный доступ к именам полей и элементов управления, используемых в базе данных, а также к множеству других встроенных функций, применяемых во время написания выражений.

4. Откройте папку **Операторы (Operators) – Сравнения** - выберите оператор *Between* и дважды кликните по нему мышкой.
5. Отредактируйте выражение: **Between “К” And “М”** и нажмите **ОК**. Обратите внимание, что раскладка должна быть на языке данных в таблице.
6. Выполните запрос  и просмотрите результаты. Обратите внимание на то, что наименования, начинающиеся на «М» в набор записей не вошли.
7. Закройте запрос, не сохраняя его.

Создание запроса, выполняющего отбор значений по шаблону



1. Откройте запрос «*Товары_И_Цены*» в режиме конструктора (Design).
2. Введите в строку **Условия отбора (Criteria)** для столбца «*НаименованиеТовара*» выражение: ***ая**.
3. Выполните запрос  и просмотрите результаты.
4. Закройте запрос, не сохраняя его.

Создание запросов, выполняющих отбор по нескольким условиям

Задание: необходимо найти все заказы определенного клиента (Джеймса Вилсона, код - ВИЛДЖ) за текущий год.

1. Создайте запрос на базе таблицы «*Заказы*», выбирая поля: «*КодЗаказа*», «*КодКлиента*», «*ДатаЗаказа*».
2. Введите для поля «*КодКлиента*» условие отбора «**ВИЛДЖ**».
3. Введите с помощью **Построителя выражений (Expression Builder)** для поля «*ДатаЗаказа*» следующее выражение: **Year([ДатаЗаказа]) =**

Year(Date()) - 1. Данное выражение позволит отобразить записи из таблицы за прошедший год.

4. Выполните запрос  и посмотрите результаты.
5. Добавьте в таблицу «Заказы» еще один заказ от данного клиента, измените условие запроса таким образом, чтобы выводились записи не за прошедший, а за текущий год: **Year([ДатаЗаказа]) = Year(Date())**. Сохраните запрос под именем «Заказы_Период» и выполните его еще раз . Убедитесь в корректности его выполнения.

Задание: аналогичным образом создайте запрос на основании таблицы «Товары» для выборки всех товаров категории «Борьба с вредителями» дороже 20 рублей. В запрос должны входить три поля из таблицы – «НаименованиеТовара», «КодКатегории» (условия отбора (**Criteria**) – число **18**, которое является кодом категории товаров «Борьба с вредителями») и «ЦенаЗаУпаковку» (сортировка по возрастанию, условия отбора (**Criteria**)>20). Проверьте выполнение запроса. Поменяйте условия отбора (**Criteria**): **Between 10 AND 30**. Выполните запрос. Сохраните запрос под именем «Товары_Вредители».

Многотабличные запросы-выборки


Таблицы в хорошо структурированной базе данных связаны друг с другом логическими отношениями. Эти отношения устанавливаются на основе общих полей, содержащихся в этих таблицах. Когда требуется просмотреть данные из связанных таблиц, используется запрос на выборку.

Задание: необходимо просмотреть общую информацию обо всех сделанных заказах (что входит в заказ, когда и кем сделан, кто ответственный за выполнение заказа).

1. Перейдите на вкладку **Создание (Create)** и нажмите на кнопку **Мастер запросов (Query Wizard)**.
2. Выберите **Простой запрос (Simple Query Wizard)** – ОК.
3. Выберите следующие поля для запроса (в указанном порядке):

Таблица «Товары»	Поле «НаименованиеТовара»
Таблица «Подробности Заказа»	Поле «Количество»
Таблица «Клиенты»	Поля «Фамилия», «Имя»
Таблица «Заказы»	Поля «КодКлиента», «ДатаЗаказа», «ТребуемаяДатаВыполнения»
Таблица «Сотрудники»	Поля «Фамилия», «Имя», «Должность».

нажмите кнопку **Далее (Next)**.

4. Выберите тип отчета **Подробный (Detailed)** – Далее (Next).
5. Введите имя запроса «**Список_Заказов_По_Датам**» - **Готово (Finish)**. Сформируется новый запрос на основании выбранных полей их нескольких таблиц.
6. Откройте данный запрос в режиме **конструктора(Design)**. Переместите поле «**ДатаЗаказа**» так, чтобы оно стало первым – т.е. вставьте его перед полем «**НаименованиеТовара**» (выделите поле и перетащите, удерживая нажатой кнопку мыши).
7. Для поля «**ДатаЗаказа**» установите значение **Сортировки (Sort) - По возрастанию (Ascending)**.
8. Выполните запрос . Обратите внимание, что заказы в запросе вывелись согласно дате их совершения – от первого к последнему.
9. Сохраните запрос.
10. Установите условие отбора для поля «**ДатаЗаказа**»:

Between Date() and Date()-19

Выполните запрос. Запрос должен вывести записи о заказах за последние 20 дней (включая текущую дату).


11. Поменяйте условие отбора для того же поля:

>#01.01.2007# And <#01.01.2010#


Выполните запрос. Запрос должен вывести все записи о заказах, сделанных в период с 2007 по 01.01.2010 г. Обратите внимание, что записи за 2007 год в запрос входят, а за 2010 – не входят.

12. Закройте запрос, не сохраняя изменений (т.е. без условий отбора).

Задание: необходимо вывести список клиентов, которые не сделали ни одного заказа.

1. Перейдите на вкладку **Создание (Create)**, в группе **Запросы (Query)** выберите **Мастер запросов (Query Wizard)**.
2. Выберите тип запроса – **Записи без подчиненных (Find Unmatched Query Wizard)** – Далее (Next).
3. Выберите главную таблицу, для записей которой необходимо выявить отсутствие соответствующих записей в подчиненной таблице, – таблицу «**Клиенты**» - Далее (Next).
4. Выберите подчиненную таблицу – таблицу «**Заказы**» - Далее (Next).
5. Общее поле для обеих таблиц – **КодКлиента** – Далее (Next).
6. В результирующем запросе необходимо вывести поля «**Фамилия**», «**Имя**» - Далее (Next).
7. Имя запроса – «**Клиенты_Без_Заказов**».
8. Выполните запрос . Проверьте правильность его выполнения.


Задание: необходимо просмотреть список товаров, поставляемых из штата Нью-Йорк (создание многотабличного запроса-выборки с условием).

1. Перейдите на вкладку **Создание (Create)** и нажмите на кнопку **Конструктор запросов(Query Design)**.
2. Добавьте таблицы «**Поставщики**» и «**Товары**», закройте окно добавления таблиц.
3. Из таблицы «**Поставщики**» в бланк запроса добавьте поля «**Организация**», «**Штат**».
4. Из таблицы «**Товары**» в бланк запроса добавьте поле «**НаименованиеТовара**».
5. В бланке запроса для поля «**Штат**» установите условие отбора: **Нью-Йорк** и выполните запрос .
6. Сохраните запрос под названием «**Поставки_Нью-Йорк**».

Создание параметрических запросов


При выполнении запроса с параметрами у пользователя запрашиваются значения полей, которые затем используются для создания условий для запроса.

Задание: необходимо создать запрос, выводящий список товаров, поставляемых из заданного пользователем штата (для выполнения этого задания необходимо модифицировать созданный ранее запрос «**Поставки_Нью-Йорк**»).

1. В окне базы данных откройте запрос «**Поставки_Нью-Йорк**».
2. Перейдите в окно конструирования и в ячейке **Условие отбора (Criteria)** столбца «**Штат**» вместо заданного ранее условия «Нью-Йорк» введите текст: **[Введите название штата:]**.
3. Выполните запрос . В поле появившегося окна введите название штата (Техас, Айова и т.п.).



В динамическом наборе, отражающем результат выполнения запроса, будут присутствовать только те поставщики, которые располагаются в запрашиваемом штате.

4. Вернитесь в окно **конструктора(Design)** и модифицируйте условие следующим образом: **Like [Введите начало названия штата:]&***
5. Выполните запрос . В появившемся окне введите одну или несколько первых букв названия штата и нажмите ОК.



В результате выполнения запроса будет выводиться список только тех поставщиков, название штата которых удовлетворяет заданному условию.

6. Сохраните запрос под именем «*Поставщики_Из_Заданного_Штата*».

Задание: на основе таблицы «*Заказы*» необходимо создать запрос, выводящий список всех заказов, выполненных сотрудниками пофамильно (поля «*КодСотрудника*», «*ДатаЗаказа*»), во временной период, указанный пользователем (т.е. пользователь задает дату начала и конца выборки). Запрос сохраните под именем «*Диапазон_Дат*». Условие для поля «*ДатаЗаказа*» (создается с помощью **Построителя выражений (Expression Builder)**):

Between [Введите дату начала списка в формате DD/ММ/YYYY]
And [Введите дату конца списка DD/ММ/YYYY]

Создание запроса с вычисляемым полем

Задание: необходимо вычислить сумму заказа определенного товара.

1. Создайте запрос (мастером запросов (**Query Wizard**) или конструктором запросов (**Query Design**)), включая в него поля из таблиц в следующем порядке:

Таблица « <i>Заказы</i> »	Поля « <i>КодЗаказа</i> », « <i>КодКлиента</i> »
Таблица « <i>Товары</i> »	Поля « <i>НаименованиеТовара</i> », « <i>ЦенаЗаУпаковку</i> »
Таблица « <i>Подробности Заказа</i> »	Поля « <i>Количество</i> », « <i>Скидка</i> »

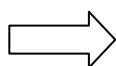
2. Присвойте имя запросу: «*Стоимость_Товара_В_Заказе*».

3. Выполните запрос , убедитесь в правильности его исполнения. Вернитесь в окно конструктора (**Design**).


4. Скройте поле «*КодКлиента*» (необходимо убрать флаг из поля **Вывод на экран (Show)**).

5. В пустую ячейку строки **Поле (Field)** введите следующий текст (для удобства ввода перейдите в окно **Область ввода (Zoom)**, нажав Shift+F2):

**Стоимость: ([ЦенаЗаУпаковку]*[Количество])-
([ЦенаЗаУпаковку]*[Количество]*[Скидка])**



Здесь «*Стоимость*» - имя нового вычисляемого поля, а текст поле двоеточия задает выражение для его вычисления.

6. Выполните запрос  и проанализируйте результаты его выполнения.

Итоги по всей таблице

Задание: в запросе «*Стоимость_Товара_В_Заказе*» подсчитать сумму всех проданных товаров.

1. Откройте запрос «*Стоимость_Товара_В_Заказе*» и оставьте результаты открытыми в **режиме таблицы (Datasheet)**.
2. На вкладке **Главная (Home)** в группе **Записи (Records)** щелкните **Итоги (Totals)**. В нижней части таблицы появится новая строка со словом **Итог** в первом столбце.
3. Щелкните ячейку в поле «**Стоимость**» с именем **Итог**. Обратите внимание на стрелку, которая появилась в ячейке.
4. Щелкните стрелку для просмотра доступных статистических функций.
5. Выберите **Сумма (Sum)**. В поле будет отображаться сумма значений в столбце.
6. Сохраните запрос.

Задание: создать запрос, выводящий информацию о покупках каждого клиента компании.

1. С помощью **Мастера запросов (Query Wizard)** создайте простой запрос на основании таблицы «*Клиенты*» (поля «*Фамилия*», «*Имя*») и запроса «*Стоимость_Товара_В_Заказе*» (поля «*КодЗаказа*», «*НаименованиеТовара*», «*Количество*», «*Скидка*», «*Стоимость*»).
2. Выберите **Итоговый отчет**, нажмите кнопку **Итоги** и выберите вычисляемое значение «**Стоимость**» - необходимо вычислить **сумму (Sum)**.
3. Сохраните запрос под именем «*Заказы_Клиентов*» и выполните его.

Перекрестные запросы

Если сводные данные требуется реорганизовать, чтобы сделать их более удобочитаемыми и понятными, целесообразно использовать перекрестный запрос.

Перекрестный запрос вычисляет сумму, среднее или иную статистическую функцию, а затем группирует результаты в два набора значений, располагая один из них вдоль боковой стороны таблицы, а другой — вдоль ее верхнего края.

При создании перекрестного запроса необходимо указать, в каких полях должны содержаться заголовки строк, в каких — заголовки столбцов, а в каких — значения, по которым вычисляются сводные данные. При задании заголовков столбцов и суммируемых значений

можно использовать только одно поле. При задании заголовков строк можно использовать до трех полей.

Задание: создать запрос, выводящий информацию о стоимости покупок каждого клиента.

1. На вкладке **Создание (Create)** выберите **Мастер запросов(Query Wizard)**, тип запроса – **перекрестный (Crosstab Query Wizard)**.
2. Новый запрос создайте на основании запроса «**Заказы_Клиентов**»: для заголовка строк (Row Heading) – поле «**Фамилия**», для названий столбцов (Column Heading) – поле «**НаименованиеТовара**»; на пересечении строк и столбцов необходимо показать **сумму стоимости** (функция – **сумма(Sum)**).
3. Задайте имя запроса: «**Клиенты_Суммы_Товары**».
4. Выполните запрос. В результате выполнения будет предоставлена информация об общей сумме покупок каждого клиента, кроме того, будет предоставлена информация о сумме покупок клиента по каждой позиции товара.
5. В **режиме таблицы (Datasheet)** используйте функцию **Итоги** на вкладке **Главная (Home)** для подсчета общей суммы продаж товаров, а также сумме продаж по каждой отдельной позиции товара (т.е. установив значение **Сумма (Sum)** для каждого поля в последней записи запроса **Итог**).
6. Сохраните запрос.

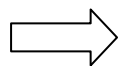
Задание: необходимо получить информацию о работе каждого сотрудника – количестве обрабатываемых заказов и полученной за заказы сумме в текущем году.

1. С помощью **Мастера запросов(Query Wizard)** создайте простой запрос на основании таблиц «**Сотрудники**» (поля «**Фамилия**», «**Должность**»), «**Заказы**» (поля «**КодЗаказа**», «**ДатаЗаказа**») и запроса «**Стоимость_Товара_В_Заказе**» (поле «**Стоимость**»).
2. Сохраните запрос под именем «**Работа_Сотрудников**». Выполните запрос.
3. В режиме **конструктора(Design)** для поля «**ДатаЗаказа**» установите **Сортировку (Sort) - По возрастанию (Ascending)**.
4. Для поля «**Фамилия**» введите Условие отбора (Criteria) [**Введите фамилию сотрудника:**].
5. Выполните запрос (для разных сотрудников – Андерсон, Эмони, Каплан и т.п.). Убедитесь в корректности его выполнения.
6. В **режиме таблицы (Datasheet)** используйте функцию **Итоги (Totals)**: для поля «**КодЗаказа**» выберите **Количество значений** (что будет подсчитывать количество заказов для выбранного сотрудника), для поля

«Стоимость товара в заказе» - **Сумма (Sum)** (будет подсчитана общая сумма заказов в обработке выбранного сотрудника).

7. Для поля «ДатаЗаказа» введите условия отбора (**Criteria**):

Year([ДатаЗаказа]) = Year(Now()).



данная конструкция позволяет в качестве условия выбора установить выбор записей только за текущий год

8. Выполните запрос для разных сотрудников. Обратите внимание, что данные функции **Итоги (Totals)** должны отображаться для каждого сотрудника.


Запросы на изменение

Запрос на создание таблицы

Запрос на создание таблицы получает данные из одной или нескольких таблиц, а затем помещает набор результатов в новую таблицу. Эта новая таблица может располагаться в базе данных, открытой в данный момент, или создается в другой базе данных.

Обычно запрос на создание таблицы создается, когда необходимо скопировать данные или поместить их в архив.

Запрос на создание таблицы состоит из двух этапов: создания запроса на выборку и преобразование этого запроса в новую таблицу.

1. Создайте новый запрос на основании таблиц «**Транспортная_Компания**» (поле «**Название компании**»), «**Доставка**» (поля «**ДатаДоставки**» и «**Тариф**»), «**Подробности_Заказа**» (поле «**КодЗаказа**»).
2. Для полей запроса «**Название**» и «**ДатаДоставки**» установите **Сортировку (Sort) - по возрастанию (Ascending)**.
3. Выполните запрос, убедитесь в корректности результатов.
4. На вкладке **Конструктор (Design)** в группе **Тип запроса (Query Type)** выберите команду **Создание таблицы** . Откроется диалоговое окно **Создание таблицы**. введите имя для новой таблицы «**Доставка_Транспорт**» (таблица должна быть помещена в текущую базу данных).
5. Выполните запрос, создав новую таблицу. Сохраните запрос под именем «**Запрос_Доставка_Транспорт**».
6. В результате выполнения запроса появилась новая таблица с записями в ней.

Запрос на добавление записи

Запрос на добавление используется, когда требуется добавить к имеющейся таблице новые строки данных.


Задание: с помощью запроса добавить запись о новом поставщике в таблицу «*Поставщики*».

1. Перейдите на вкладку **Создание (Create)** и нажмите на кнопку **Мастер запросов (Query Wizard)**.
2. Выберите **Простой запрос (Simple Query Wizard)** – в качестве источника выберите таблицу «*Поставщики*» - добавьте в запрос все поля таблицы (кнопка >> добавляет сразу все поля) – имя запроса «*Данные_О_Поставщиках*» - ОК.
3. В запросе внесите информацию о новом поставщике (произвольную), заполнив все поля запроса. После заполнения полей запроса данные автоматически должны передаться в таблицу «*Поставщики*».
4. Нажмите кнопку **Обновить все**.
5. Откройте таблицу «*Поставщики*» и убедитесь, что в ней появилась новая запись.
6. В запросе «*Данные_О_Поставщиках*» измените название организации для последней введенной записи и выполните запрос.
7. Откройте таблицу «*Поставщики*» и убедитесь, что название организации в последней записи также изменилось.
8. Сохраните изменения.

Запрос на обновление записи

С помощью запроса на обновление можно добавлять, изменять или удалять данные в одной или нескольких записях.

Задание: тариф доставки всех транспортных компаний на все заказы увеличился в два раза. Необходимо ввести новый тариф (работая с таблицей «*Доставка_Транспорт*»).

1. На базе таблицы «*Доставка_Транспорт*» создайте конструктором новый запрос на обновление , включив в него поле «*Тариф*».
2. В ячейку **Обновление (Update To)** для поля «*Тариф*» введите следующее выражение: **[Тариф]*2**.
3. Выполните запрос, подтвердив обновление записей.
4. Сохраните запрос под именем «*Удвоение_Тарифа*».
5. Закройте запрос и посмотрите измененную таблицу.

6. Выполните запрос «*Удвоение_Тарифа*» еще раз, подтвердив все операции. Проверьте, что в таблице «*Доставка_Транспорт*» тарифы повысились еще раз.

Запрос на удаление записи

Задание: удалить из запроса все транспортные компании, тариф которых превышает 1 000 руб.

1. Создайте новый запрос, внося в него все поля таблицы «*Доставка_Транспорт*».
2. В режиме конструктора в группе **Тип запроса (Query Type)** выберите кнопку **Удаление (Delete)**.
3. В ячейку **Условие отбора (Criteria)** для поля «*Тариф*» введите выражение: **>1000**.
4. Выполните запрос; убедитесь, что в представленных данных нет тарифов больше 1 000 руб.
5. Сохраните запрос под именем «*Условие_Тарифа*».

Запросы в режиме SQL

Для извлечения данных из базы данных используется язык SQL (Structured Query Language — язык структурированных запросов). SQL — это язык программирования, который очень напоминает английский, но предназначен для программ управления базами данных. Каждый выполняемый запрос на самом деле основан на SQL.

Простой запрос на выборку

Задание: из таблицы «*Транспортная_Компания*» выбрать все транспортные компании, кроме «Кенгуру».

1. Создайте запрос на выборку на основании таблицы «*Транспортная_Компания*» (поле «*Название компании*»).
2. Выполните запрос.
3. Перейдите в **режим SQL («Вид(View)» - SQL)**. Измените код таким образом:

```
SELECT [Транспортная_Компания].[НазваниеКомпании]  
FROM [Транспортная_Компания]  
WHERE [Транспортная_Компания].[НазваниеКомпании]<>'Кенгуру';
```

4. Выполните запрос, убедитесь, что он выполняется корректно.
5. Вернитесь в режим SQL. Программно измените запрос таким образом, чтобы он выводил не только название компании, но и адрес (имения полей «Улица», «Город», «Штат»). Выполните запрос.
6. В режиме SQL измените запрос таким образом, чтобы выводились только транспортные компании, расположенные в городе Денвер. Выполните запрос.
7. Сохраните запрос под именем «*SQL_Выборка*».
8. Сделайте сортировку выбранных записей по названию компании в алфавитном порядке:
ORDER BY НазваниеКомпании;
9. Выполните запрос. Сохраните изменения.


Задание: удалить таблицу «*Доставка_Транспорт*»

1. В конструкторе запросов выберите режим SQL.
2. Введите следующий код:
DROP TABLE [Доставка_Транспорт];
3. Выполните запрос, убедитесь, что выбранная таблица удалена.
4. Сохраните запрос под именем «*Удаление_Таблицы*».

Запрос на объединение

Данный вид запросов используется для объединения идентично структурированных данных.

Задание: создать запрос, возвращающий список всех контактов компании (клиенты, сотрудники) с номерами телефонов, отсортированный в алфавитном порядке.

1. В конструкторе запросов выберите тип запроса – объединение  (автоматически работа будет происходить в режиме SQL).
2. Введите следующий код:

```
SELECT Фамилия, Имя, НомерТелефона FROM Клиенты UNION
SELECT Фамилия, Имя, Телефон FROM Сотрудники;
```

3. Для объединения двух полей «*Фамилия*» и «*Имя*» в одно – «*ФамилияИмя*» измените код следующим образом:

```
SELECT Фамилия&" "&Имя AS ФамилияИмя, НомерТелефона FROM
Клиенты
UNION
SELECT Фамилия&" "&Имя AS ФамилияИмя, Телефон FROM Сотрудники;
```

4. Сохраните запрос под именем «*Список_Телефонов*».

ФамилияИмя	НомерТелефона
Акерс Ким	(604) 555-0193
Албук Стив	(604) 555-0192
Андерсон Эмми	(206) 555-0105
Аскерман Пилар	(206) 555-0194

Рис. 2.1. Результат выполнения запроса «*Список_Телефонов*».

Сохранение и закрытие базы данных

1. Сохраните все изменения.
2. Завершите работу с Microsoft Access.

Лабораторная работа №3: Формы

Форма — это объект базы данных, который можно использовать для ввода, изменения или отображения данных из таблицы или запроса. Формы применяются для управления доступом к данным, например для определения того, какие поля или строки данных должны отображаться. Эффективная форма ускоряет работу с базой данных, так как пользователям не требуется искать нужные сведения. Привлекательная форма делает работу с базой данных не только эффективной, но и более приятной. Кроме того, формы могут предотвращать ввод неверных данных. Виды форм:

1. Форма для ввода и модификации данных:
 - 1.1. Форма с одним элементом;
 - 1.2. Разделенная форма;
 - 1.3. Форма для нескольких элементов (ленточная форма);
 - 1.4. Составная форма (главная и подчиненная, с отношением «один-к-многим»).
2. Сводная таблица (сводная диаграмма).
3. Форма навигации.
4. Пользовательское окно диалога.

Форму можно создавать с помощью множества средств, находящихся на вкладке **Создание (Create)** в группе **Формы (Forms)** (в зависимости от типа и сложности создаваемой формы). Ниже будут рассмотрены различные варианты создания.

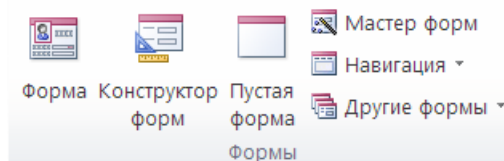


Рис. 3.1. Группа *Формы*

Формы для ввода и модификации данных

Форма с одним элементом

При работе с данным видом формы одновременно выводится только одна запись.

Задание: на основе таблицы «Доставка» создать форму «Доставка», содержащую соответствующую информацию.

1. В окне **Области переходов (Navigation Pane)** выберите таблицу «Доставка».
2. На вкладке **Создание(Create)** в группе **Формы (Forms)** выберите элемент **Форма (Form)**.
3. В **режиме макета (Layout View)** откроется вновь созданная на базе указанной таблицы форма. Перейдите в **режим формы (Form View)** и проверьте корректность отображаемой информации. Обратите внимание, что на форме также отображается связанная с таблицей «Доставка» информация о заказах.
4. Сохраните форму под именем «Доставка».
5. Настройка свойств формы:
 - 5.1. В **режиме конструктора (Design)** откройте **Окно свойств формы (Property Sheet)**, щелкнув на кнопке **Страница свойств (Property Sheet)** в группе **Сервис (Tools)** на вкладке **Конструктор (Design)**. Откроется **окно свойств (Property Sheet)**.
 - 5.2. В **окне свойств(Property Sheet)** для полей **Надписи** щелкните на вкладке **Макет (Format)**, найдите и задайте значения (на ваше усмотрение) следующим свойствам:
 - ✓ Шрифт (Font Name);
 - ✓ Размер шрифта (Font Size);
 - ✓ Насыщенность (Font Weight);
 - ✓ изучите другие доступные свойства.

(чтобы выбрать элемент управления или область формы, для которой задаются свойства, необходимо щелкнуть по этому элементу или области и таким образом выбрать его или выбрать необходимый элемент из выпадающего меню Окна свойств).

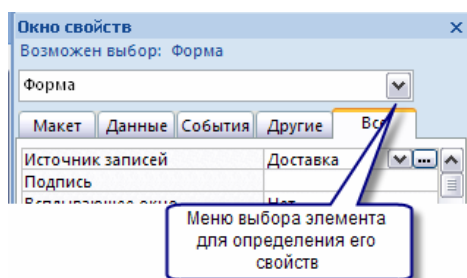


Рис. 3.2. Окно свойств формы.

- 5.3. Откорректируйте поле **Подпись (Caption)** для размещенных на форме элементов (задайте подписи элементам).
- 5.4. Щелкните в любом месте **Области данных (Details)** формы и установите:
 - ✓ цвет фона (Back Color);
 - ✓ оформление (Special Effect);
 - ✓ изучите другие доступные свойства.
6. Для выравнивания элементов формы в **режиме конструктора** перейдите на вкладку **Упорядочить** в группу **Размер и порядок**, кнопка **Выровнять**.

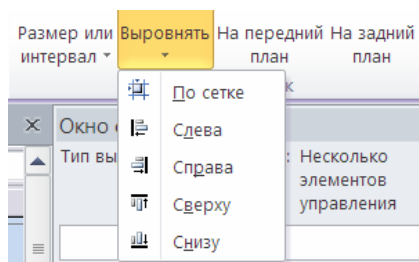


Рис. 3.3. Группа *Размер и порядок*.

- 6.1. Выделите на форме названия полей и выровняйте по левому краю, а также по нижнему или верхнему.
- 6.2. Выделите на форме сами поля и выровняйте по правому краю, а также по нижнему или верхнему.

Задание: на основе таблицы «*Транспортная Компания*» создать форму «*Транспорт*» с помощью **Мастера форм (Form Wizard)**.

1. На вкладке **Создание (Create)** в группе **Формы (Forms)** выберите элемент **Мастер форм (Form Wizard)**.
2. В окне **Мастера форм (Form Wizard)** выберите все поля таблицы «*Транспортная Компания*» (>>) – **Далее (Next)** – **В один столбец (Stacked)** – **Далее (Next)** – **Имя «Транспорт» - Готово (Finish)**.
3. Перейдите в режим макета, в группе **Темы (Themes)** выберите понравившийся стиль оформления.
4. Изучите сформированную форму. Перейдя к последней записи – добавьте произвольную новую, заполнив все поля. Откройте таблицу «*Транспортная Компания*» и убедитесь, что внесенная через форму запись появилась в таблице.

Задание: на основе таблицы «*Фактический Адрес Доставки*» создать форму «*Фактический Адрес*» с помощью **пустой формы (Blank Form)**.

1. На вкладке **Создание (Create)** в группе **Формы (Forms)** выберите элемент **Пустая Форма (Blank Form)**.

2. Справа откроется окно **Списка полей (Field List)**, в котором необходимо выбрать и перенести на форму все поля из таблицы «**Фактический Адрес Доставки**» (двойным нажатием на название поля или перетаскиванием поля мышкой). Если **Список полей (Field List)** не открыт – откройте его, нажав на кнопку **Добавить существующие поля (Add Existing Fields)** в группе **Сервис (Tools)**. В результате, автоматически сформируется простейшая форма.
3. Сохраните форму под именем «**Фактический Адрес**».
4. В **режиме конструктора (Design)** поработайте со стилем формы.
5. Сохраните форму.

Задание: на основе таблицы «**Сотрудники**» с помощью **Конструктора форм (Form Design)** создать форму «**Сотрудники**», предоставляющую в компактном виде полную информацию о персонале организации.

The screenshot shows a form titled "Сотрудники" (Employees) with a table of employee data and a photo. The table has columns for "Код сотрудника" (Employee ID) and "Имя" (Name). The data row shows "1" for the ID and "Катрин Тернер" (Katrin Turner) for the name. Other fields include "Фамилия" (Surname) as "Тернер", "Должность" (Position) as "Владелец" (Owner), "ДатаНайма" (Hire Date) as "01.05.1992", and "ДатаРождения" (Date of Birth) as "08.12.1968". A photo of a woman is displayed on the right side of the form.

Код сотрудника	Имя
1	Катрин Тернер

Фамилия: Тернер
Должность: Владелец
ДатаНайма: 01.05.1992
ДатаРождения: 08.12.1968

Рис.3.4. Форма «Сотрудники».

1. На вкладке **Создание (Create)** в группе **Формы (Forms)** выберите элемент **Конструктор форм (Form Design)**.
2. В окне **Список полей (Field List)** выберите и перенесите на форму следующие поля таблицы «**Сотрудники**»: «**Код**», «**Имя**», «**Фамилия**», «**Должность**», «**ДатаНайма**», «**ДатаРождения**». Расположите их друг под другом.
3. Из группы **Элементы управления (Controls)** на вкладке **Конструктор (Design)** выберите элемент **Вкладка** и активизируйте его щелчком мыши.

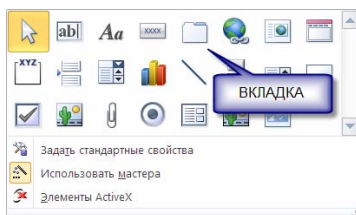


Рис. 3.5. Группа Элементы Управления

4. Поместите вкладку на поле, щелкнув на том месте поля, где вы хотите разместить вкладку.

5. На первую вкладку из таблицы «*Сотрудники*» поместите следующие поля: «*Адрес*», «*Город*», «*Штат*», «*Индекс*», «*Телефон*».
6. В группе **Сервис (Tools)** нажмите кнопку **Страница свойств (Property Sheet)**, в открывшемся **Окне свойств (Property Sheet)** выберите для элемента Вкладка (с соответствующим номером) вкладку **Макет (Format)** и в поле **Подпись (Caption)** напишите «Контактная информация».
7. Перейдите на вторую вкладку, откройте **Список полей (Field List)** и перенесите на вкладку поле «*Заметки*» из таблицы «*Сотрудники*». Растяните поле содержания заметки на все поле вкладки, присвойте вкладке **Подпись (Caption)** «*Заметки*» (см. п. 6).
8. Перейдите в **режим формы (Form View)** и посмотрите на результат.
9. Вернитесь в **режим конструктора (Form Design)** и отредактируйте внешний вид формы (расположение элементов и т.п.).
10. Создайте заголовок формы, выбрав в **режиме конструктора (Design)** в группе **Колонтитулы** элемент управления **Заголовок**.
11. В **режиме конструктора (Design)** добавьте на форму третью вкладку, выбрав элемент управления **Вставить вкладку**. Присвойте ей **Подпись (Caption)** «*Приложения*».
12. На вкладку «*Приложения*» перетащите из таблицы «*Сотрудники*» поле «*Приложения*» (всю связанную группу элементов). Отредактируйте внешний вид вкладки.
13. Перейдите в **режим формы (Form View)**. На вкладке «*Приложения*» должны отображаться фото и/или ярлык текстового документа, содержащего резюме сотрудника.
14. Если приложение было не одно – для того, чтобы открыть все приложения необходимо кликнуть мышкой на фото или ярлыке приложения. Появится меню навигации по приложениям:

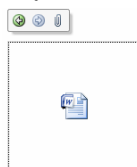


Рис. 3.6. Ярлык поля «Приложение» и меню навигации.

Для последовательного перемещения по приложениям необходимо нажать стрелку, для открытия списка всех приложений – изображение скрепки.

15. Изучите получившуюся форму и сохраните ее под именем «*Сотрудники*».
16. Перейдите в **режим макета (Layout View)**. Выделите на вкладке **Заметки** поле **Заметки**. Откройте **Страницу свойств (Property Sheet)** (для данного поля), перейдите на вкладку **Данные (Data)**. Изучите возможности параметров **Доступ (Access)** и **Блокировка (Block)**, применяя их на **поле Заметки**. Обратите внимание на возможность запрета изменения данных в поле формы.
17. Сохраните изменения.

Разделенная форма

Данный вид формы позволяет одновременно отображать данные в двух представлениях — в **режиме формы (Form View)** и в режиме таблицы.

Задание: на основе таблицы «Товары» создайте форму «Товары» таким образом, чтобы была возможность работать с данными как через форму, так и напрямую с таблицей.

Код това	НаименованиеТовара	ЛатинскоеНазвание	Код поставщика
1	Ликорис чешуйчатый	Lycoris squamigera	8
2	Осеннецветущий крокус	Colchicum	15
3	Ящик для компоста		1

Рис. 3.7. Форма «Товары».

1. В **Области переходов (Navigation Pane)** выберите таблицу «Товары».
2. На вкладке **Создание (Create)** в группе **Формы (Forms)** выберите элемент **Разделенная форма (Split Form)**.
3. Проверьте корректность отображения данных на форме.
4. Сохраните форму под именем «Товары».
5. Добавьте новые записи – сначала одну через форму, затем одну напрямую в таблицу. Убедитесь, что записи появляются в обоих представлениях.
6. В **режиме конструктора (Design)** добавьте на форму элемент управления **Поле** (см. рис.3.8). Укажите **Надпись поля «На сумму»**. Откройте **Окно свойств (Property Sheet)** собственно поля, вкладку **Данные (Data)**, свойство **Данные** и введите следующее выражение (воспользуйтесь **Построителем выражений**):
$$=[\text{ЦенаЗаУпаковку}]*[\text{Кол-воНаСкладе}]$$

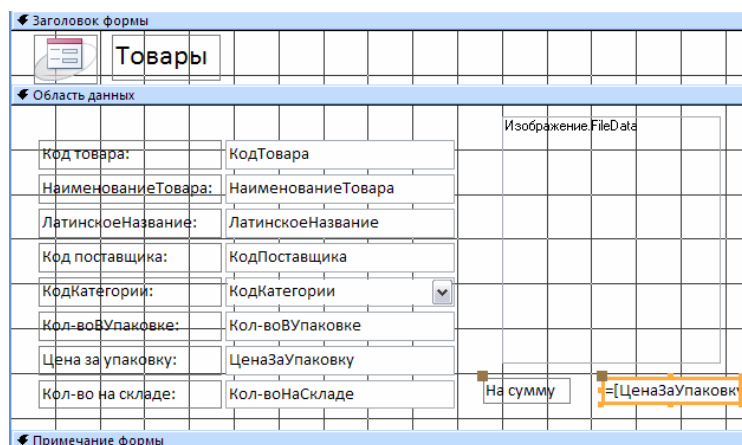


Рис. 3.8. Форма «Товары», режим Конструктора.

- Отредактируйте формат поля. Удалите из нижней части формы дублирующую информацию вычисляемое поле «ТоваровНаСумму». Убедитесь, что поле удалено только из формы – а не из таблицы-источника формы. Сохраните изменения.

Ленточная форма

Представляет собой форму для нескольких элементов и позволяет отображать сведения более чем об одной записи одновременно.

Задание: на основе запроса «*Список_Заказов_По_Датам*» создать форму «*Список_Заказов_По_Датам*».

- В окне **Области переходов (Navigation Pane)** выделите запрос «*Список_Заказов_По_Датам*».
- На вкладке **Создание (Create)** в группе **Формы (Forms)** выберите элемент **Несколько элементов (Multiple Items)**.
- В **режиме макета (Layout View)** появится созданная форма. Отредактируйте ее внешний вид.
- Перейдите в **режим формы (Form View)**, проверьте корректность отображаемых данных.
- Сохраните форму под именем «*Список_Заказов_По_Датам*».

Составная форма (главная и подчиненная)

Необходима для просмотра в одной форме данных из нескольких таблиц со связью «Один-ко-многим».

Задание: на основе таблиц «*Категории Товаров*» и «*Товары*» разработать форму, отображающую категории товаров. Каждой отображающейся категории должен соответствовать список товаров данной категории.

1. На вкладке **Создание (Create)** в группе **Формы (Forms)** выберите элемент **Мастер форм (Form Wizard)**.
2. В **Мастере форм (Form Wizard)** из таблицы «*Категории Товаров*» выберите все поля, из таблицы «*Товары*» - все, кроме поля «*Код категории*», нажмите **Далее (Next)**.
3. Вид представления данных – подчиненные формы.
4. Внешний вид подчиненной формы – ленточный.
5. Задайте имена для форм: для главной - «*Категории Товары*», для подчиненной - «*Товары Подчиненная Форма*», нажмите **Готово (Finish)**.
6. Удалите из подчиненной формы изображение, отредактируйте внешний вид форм.
7. Проверьте корректность отображения данных в форме – одной категории должны соответствовать несколько товаров.

Задание: на основе таблицы «*Клиенты*» создать форму, отображающую полную информацию о клиентах, а также заказах, сделанных каждым клиентом (используя формы «*Список Заказов По Датам*» и «*Фактический Адрес*» в качестве подчиненных).

Дата заказа	Наименование товара	ДатаВыполнения	Количество	Фамилия
02.02.2001	Тюльпаны	07.02.2001	1	Вилсон
02.02.2001	Лилия	07.02.2001	6	Вилсон
02.02.2001	Костяная мука	07.02.2001	1	Вилсон
03.02.2008	Защита от животных		3	Вилсон
07.02.2009	Наконечник для полива		1	Вилсон
07.02.2009	Костяная мука		1	Вилсон
07.02.2009	Шланг 25'		1	Вилсон
07.02.2009	Почвенная смесь		1	Вилсон

Рис. 3.9. Форма «Клиенты».

1. На вкладке **Создание (Create)** в группе **Формы (Forms)** выберите элемент **Конструктор форм (Form Design)**.
2. В окне **Список полей (Field List)** выберите и перенесите на форму все поля таблицы «*Клиенты*».
3. Добавьте на форму элемент **Вкладка** с целью добавления на вкладки сведений из подчиненных таблиц.
4. На вкладке **Конструктор (Design)** в группе **Элементы управления (Controls)** активируйте элемент управления **Подчиненная форма/отчет** и щелкните на поле первой вкладки. Откроется **Мастер подчиненных форм**.

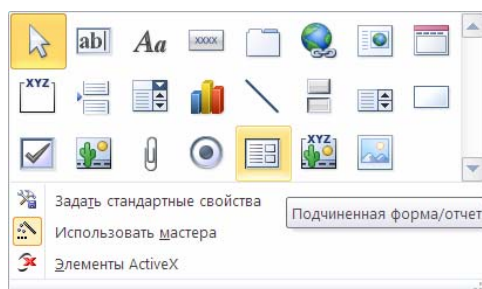


Рис. 3.10. Группа Элементы Управления

5. Для вставки подчиненной формы в **Мастере подчиненных форм** выберите форму «**Список_Заказов_По_Датам**» - **Далее (Next) – Выбор из списка – Далее (Next) – Предложенное** автоматически имя подчиненной формы – **Готово (Finish)**.
6. Аналогичным образом на вторую вкладку поместите форму «**Фактический_Адрес**».
7. Отредактируйте внешний вид всех форм.
8. Откорректируйте положение и размер окон таблиц и вкладок, проверьте корректность отображения формы в **режиме формы (Form View)** – для каждого клиента на основанной форме должны отображаться все его заказы и все его фактические адреса на вкладках.
9. Определите названия вкладок.
10. Оформление группы переключателей “Credit Card”
 - 10.1. Удалите с основной формы поле **Кредитная карта**.
 - 10.2. В группе **Элементы управления (Controls)** активизируйте кнопку **Использование мастера (Control Wizard)** и выберите элемент управления **Группа переключателей (Option Group)**.

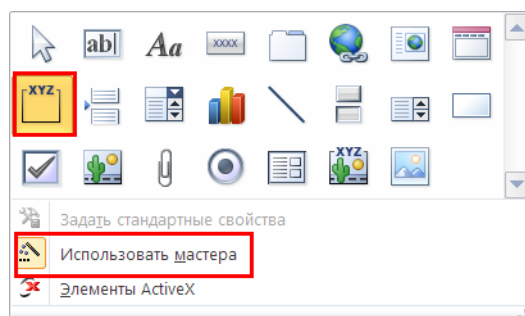


Рис. 3.11. Группа Элементы Управления

- 10.3. Разместите группу переключателей на основной форме. В появившемся окне **Создание группы переключателей (Option Group Wizard)** введите названия типов кредитных карт (*MasterCard, Visa, American Express*) – **Далее (Next) – Значения по умолчанию «нет» – Далее (Next) – Далее (Next) – Сохранить значение** в поле «**КредитнаяКарта**» - **Далее (Next) – Выберите оформление – Название группы «Credit Card» - Готово (Finish)**.
11. Переключитесь в **режим формы (Form View)** и посмотрите результат.

12. В режиме конструктора (Design) или макета (Layout View) сформируйте заголовок формы - «Клиенты».
13. Отредактируйте внешний вид формы.
14. Сохраните форму под именем «Клиенты».

Задание: аналогичным образом в режиме конструктора (Design) создайте форму «Поставщики», которая отображает информацию о каждом поставщике и о поставляемых им товарах.

Главная форма должна быть разработана на основе таблицы «Поставщики», подчиненная на основании существующей формы «Товары_Подчиненная_Форма». Отредактируйте внешний вид формы.

Информация о поставщиках

Код поставщика: 1

Организация: Садовый инвентарь Факс: 555-5555
 Фамилия: Ричардсон Адрес: 45 Валовая ул.
 Имя: Девид Город: Лэнсинг
 Должность: Заведующий отделом закупок Штат: Мичиган
 РабочийТелефон: (202) 555-0120 Индекс: 84736
 МобильныйТелефон: (123) 456-4567

Поставляемые товары:

Код	Наименование товара	Латинское	Код	КодКатегории	Кол-во в
3	Ящик для компоста		1	Инструменты	6 литров
23	Вращающийся опрыскиватель		1	Инструменты	1 шт.
82	Сеялка		1	Инструменты	1 шт.
83	Электрокосилка		1	Инструменты	1 шт.
92	грабли		1	Инструменты	1 шт.
96	15 литровая автоцистерна для		1	Инструменты	1 шт.
102	Тачка		1	Инструменты	1 шт.
103	Шланг 50'		1	Инструменты	1 шт.
104	Шланг 25'		1	Инструменты	1 шт.
108	Навозоразбрасыватель		1	Инструменты	1 шт.
109	Садовые ножницы		1	Инструменты	1 шт.

Рис.3.12 Форма «Поставщики».

Сводная таблица / сводная диаграмма

Сводная таблица (Pivot Table) и сводная диаграмма (Pivot Chart) дают возможность выполнять комплексный анализ данных (по аналогии с MS Excel).

Задание: необходимо проанализировать продажи по каждой категории товаров.

1. Создайте запрос «Анализ_Продаж_Категорий», включив в него следующие поля:

Название таблицы	Имя поля
«Категории Товаров»	«НазваниеКатегории»
«Товары»	«НаименованиеТовара», «КодКатегории»
«Подробности Заказа»	«Количество»

2. Выполните запрос, убедитесь в правильности его выполнения.
3. Перейдите на вкладку **Создание (Create)** и выберите тип формы **Сводная таблица (Pivot Table)**. Откроется рабочее окно, на котором необходимо разместить поля из запроса.
4. Для отображения списка полей (Pivot Table Field List) запроса кликните мышкой на пустом поле или выберите кнопку **Список полей (Field List)** в группе **Показать или скрыть (Show/Hide)** на вкладке **Конструктор (Design)**.
5. Добавьте поля строк создаваемой сводной таблицы: «*НазваниеКатегории*», «*НаименованиеТовара*»; поле «*Количество*» добавьте в поле итогов для деталей.



Рис. 3.13. Окно размещения элементов сводной таблицы.

6. Сохраните форму (сводную таблицу) под именем «*Анализ_Продаж_Категорий*».
7. Выделите поле «*Количество*» и на вкладке **Конструктор (Design)** в группе **Сервис (Tools)** выберите **Автовычисления – Сумма (Sum)**. Эта функция позволит просматривать количество проданных единиц товаров по каждой категории без указания подробностей.

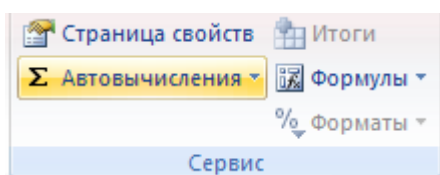


Рис. 3.14. Группа Сервис

8. В результате сводная таблица должна выглядеть следующим образом:

НазваниеКатегории	НаименованиеТовара	Количество
Болотные растения	Кувшинка белая	1
	Кувшинка белая	1
	Кувшинка белая	1
	Кувшинка белая	2
	Понтедерия сердцевидная	1
	Белоцветик щитовидный	1
	Элодея	1
	Сумма "Количество"	8
Бонсай Оборудование	Можжевельник чешуйчатый	1
	Набор инструментов для бонсаи	1

Рис.3.15. Сводная таблица «*Анализ_Продаж_Категорий*»

9. На основании запроса «*Анализ_Продаж_Категорий*» необходимо создать сводную диаграмму. Для этого на вкладке создание необходимо выбрать кнопку **Сводная диаграмма (Pivot Chart)**.
10. Для открытия окна **Список полей данных (Pivot Chart Field List)** необходимо щелкнуть на пустом поле области диаграммы или нажать кнопку **Список полей (Field List)** в группе **Показать или скрыть (Show/Hide)** на вкладке **Конструктор (Design)**.
11. В область поля категории перетащите поле «*НазваниеКатегории*», в область полей данных – «*Количество*». Откорректируйте внешний вид диаграммы, измените тип по Вашему усмотрению.

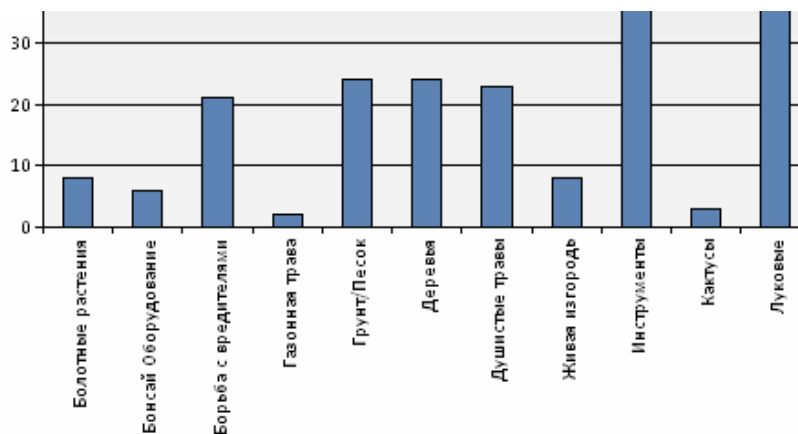


Рис. 3.16. Сводная диаграмма.

12. Сохраните форму (диаграмму) под именем «*Анализ_Продаж_Диаграмма*».

Окно диалоговой формы

Окна диалоговой формы, как правило, используются для сбора информации, предоставляемой пользователем. Они всегда модальные – т.е. пользователь не может перейти к выполнению других этапов работы с приложением, пока не закончит обработку формы. Разрабатываются, чаще всего, с использованием средств программирования.

Задание: создать простейшую диалоговую форму.

1. На вкладке **Создание (Create)** в группе **Формы (Forms)** в меню **Другие формы (More Forms)** выберите **Модальное диалоговое окно (Modal Dialog)**.
2. На вкладке **Конструктор (Design)** в группе **Элементы управления (Controls)** выберите элемент **Подпись (Caption)** и разместите его на форме. Введите произвольный текст вопроса (например, «Вы уверены, что хотите завершить работу с приложением?»).

3. Перейдите в **режим формы (Form View)** и попробуйте открыть любые объекты (вкладки, таблицы и т.п.). Обратите внимание на то, что пока вы не закончите работу с окном, вы не сможете продолжить работу.
4. Выберите вариант ответа и закройте окно, не сохраняя форму.


Сохранение и закрытие базы данных

1. Сохраните все изменения.
2. Завершите работу с Microsoft Access.

Лабораторная работа №4: Отчеты

Создание отчета с помощью средства «Отчет»

Задание: на основе таблицы «*Поставщики*» создайте простой отчет, содержащий информацию о поставщиках.

1. В области переходов выберите таблицу «*Поставщики*».
2. На вкладке **Создание (Create)** в группе **Отчеты (Reports)** щелкните **Отчет (Report)**.  Приложение Access автоматически сформирует отчет и отобразит его в **режиме макета (Layout)**.
3. Просмотрите этот отчет в режимах **конструктора (Design)**, **предварительного просмотра (Print Preview)**, **представления отчета**.
4. Измените свойства желаемых объектов (например, шрифт, цвет, расположение элементов и т.п.)
5. Сохраните отчет под именем «*Поставщики*» и закройте его.

Создание отчета с помощью Мастера отчетов

Табличный отчет

Задание: на основе таблицы «*Сотрудники*» создайте отчет, содержащий информацию о сотрудниках.

1. Предварительно создайте запрос на базе таблицы «*Сотрудники*», включите в него все поля, кроме поля «*Приложение*» и связанных с ним полей, отсортируйте поле «*Фамилия*» по возрастанию. Сохраните под именем «*Сотрудники_Запрос*»
2. На вкладке **Создание (Create)** в группе **Отчеты(Reports)** щелкните **Мастер отчетов (Report Wizard)**.
3. Выберите запрос «*Сотрудники_Запрос*» и все его поля – **Далее (Next)** – **Далее(Next)** - **Далее(Next)**.

4. Выберите макет **табличный**, ориентацию **альбомную (Landscape)** - **Далее(Next)**.
5. Присвойте имя отчету: **«Сотрудники» - Готово (Finish)**.
6. Сформированный отчет будет показан в окне предварительного просмотра.
7. В **режиме конструктора(Design)** отредактируйте расположение данных в отчете таким образом, чтобы оно соответствовало ширине страницы. Поменяйте местами поля **«Фамилия»** и **«Имя»**.
8. Сохраните отчет.

Отчет в один столбец

Задание: создайте отчет на основании таблицы **«Транспортная_Компания»**.

1. На вкладке **Создание(Create)** в группе **Отчеты(Reports)** щелкните **Мастер отчетов(Report Wizard)**.
2. Выберите таблицу **«Транспортная_Компания»** и все ее поля – **Далее(Next) – Далее(Next) – Далее(Next)**.
3. Выберите макет **в столбец**, ориентация **книжную – Далее(Next)**.
4. Присвойте имя отчету: **«Транспортная_Компания» - Готово(Finish)**.
5. Если необходимо – отредактируйте результаты в **режиме конструктора(Design)**.
6. Сохраните отчет.

Почтовая наклейка

Задание: создать наклейки для почтовых отправлений клиентам (на базе таблицы **«Клиенты»**, поля **«Имя»**, **«Фамилия»**, **«Адрес»**, **«Город»**, **«Штат»**, **«ПочтовыйИндекс»**, **«Страна»**).

1. На вкладке **Создание(Create)** в группе **Отчеты(Reports)** щелкните **Наклейки (Labels)**.
2. С помощью **мастера наклеек (Label Wizard)**, последовательно отвечая на вопросы, создайте отчет.
3. Название отчета **«Наклейки_Клиенты»**.
4. В случае необходимости отредактируйте отчет.
5. Посмотрите отчет во всех режимах.

Группировка данных и вычисление итогов

Задание: на основе запроса **«Заказы_Клиентов»** создайте отчет, выводящий информацию обо всех заказах каждого клиента и общей сумме заказов клиента.

1. На вкладке **Создание (Create)** в группе **Отчеты (Reports)** щелкните **Мастер отчетов(Report Wizard)**.
2. Выберите запрос «**Заказы_Клиентов**» и все поля запроса, кроме поля «**Имя**» – **Далее(Next)**.
3. Добавьте уровни группировки – «**Фамилия**» - **Дале(Next)е**.
4. Нажмите кнопку **Итоги (Summary Options)** и задайте вычисление итоговых значения по сумме (Sum) (Sum - стоимость), **показать данные и итоги – ОК – Далее(Next)**.
5. Выберите любой макет и **альбомную ориентацию (Landscape)** – **Далее(Next)**.
6. Имя отчета «**Заказы_Клиентов**».
7. В **режиме конструктора(Design)** или в **режиме макета (Layout View)** измените формат поля «**Sum-([Sum-Стоимость])**» на денежный (открыв **Окно свойств (Property sheet)**, вкладку **Макет (Format)**, установив **формат поля**).
8. Отредактируйте отчет в случае необходимости.
9. Сохраните отчет.

Создание отчетов без помощи Мастера

Создание простого отчета

1. Выберите сформированный отчет «**Поставщики**», изучите его в **режиме конструктора(Design)**, закройте отчет.
2. На вкладке **Создание (Create)** в группе **Отчеты (Reports)** выберите **Конструктор отчетов (Reports Design)**.
3. В **режиме конструктора(Design)**, начиная с пустого поля, используя стандартные элементы управления и их свойства, сформируйте отчет, аналогичный изученному отчету «**Поставщики**», вручную.

Создание сложного отчета

Задание: создать отчет, выводящий информацию о том, какая позиция товара какими клиентами была куплена (отчет аналогичен отчету «**Заказы_Клиентов**» с тем отличием, что группировка должна происходить по полю «**НаименованиеТовара**»).

1. На вкладке **Создание (Create)** в группе **Отчеты(Reports)** выберите **Мастер отчетов(Report Wizard)**.
2. Добавьте все поля из запроса «**Заказы_Клиентов**», на основании которого формируется отчет.
3. Оформите отчет, отвечая на вопросы **Мастера** (группировки нет).
4. Имя отчета «**Товары_По_Клиентам**».

5. Ознакомьтесь с полученным отчетом и перейдите в режим макета(Layout).
6. В режиме макета(Layout) отредактируйте отчет. Удалите поля – «Имя», «КодЗаказа», «Скидка»; перенесите поле «НаименованиеТовара» – в начало (см. рис.4.1).
7. Измените заголовок отчета на «Товары клиентов».
8. В режиме макета(Layout) пределите группировку (Grouping) (вкладка Конструктор (Design) группа Группировка и итоги (Grouping&Totals), кнопка Группировка (Group)) по Наименованию товара, отсортируйте по Наименованию товара по возрастанию («начиная с А»).

Товары клиентов

НаименованиеТовара	Фамилия	Количество	Sum - Стоимость
"Бдительная сова"(против насекомых)	Вилламс	2	11,78р.
		2	11,78р.
"Мартин Хаус" (против насекомых)	Клайтон	1	70,00р.
	Смит	1	67,90р.
"Улий" (против насекомых)		2	137,90р.
	Никонен	1	13,58р.
"Эвегрин", удобрения		1	13,58р.
	Сак	1	4,37р.

Группировка, сортировка и итоги

Группировка **НаименованиеТовара** ▾ начиная с А ▾ , по всему значению ▾ , с итогами: Sum - Стоимость ▾ , с заголовком **НаименованиеТовара** , с разделом заголовка ▾ , с разделом примечания ▾ , не удерживать группу на одной странице ▾ , Основные параметры ◀

[+] Добавить группировку [A] Добавить сортировку

Рис. 4.1. Отчет «Товары По Клиентам», режим Макета

9. В разделе Группировка, сортировка и итоги (Group, Sort, And Total) в разделе группировка **Наименование товара (Group) «Больше» (More)** указать следующие параметры:
 - 9.1. **Итоги** по полю «Sum-стоимость», **тип** – сумма, **флаг** – *показать в примечании группы*, **флаг** - *показать общий итог*.
 - 9.2. **Итоги** по полю «Количество», **тип** – сумма, **флаг** – *показать в примечании группы*.
 - 9.3. **Заголовок** - «Наименование товара».
 - 9.4. *С разделом заголовка (With a header section)*.
 - 9.5. *С разделом примечания (With a footer section)*.
10. В режиме конструктора(Design) установите для полей «Sum – стоимость» денежный формат.
11. В режиме конструктора (Design) в примечание группы добавьте элемент управления **Подпись (Caption)** («ИТОГО:»), см. рис. 4.2.

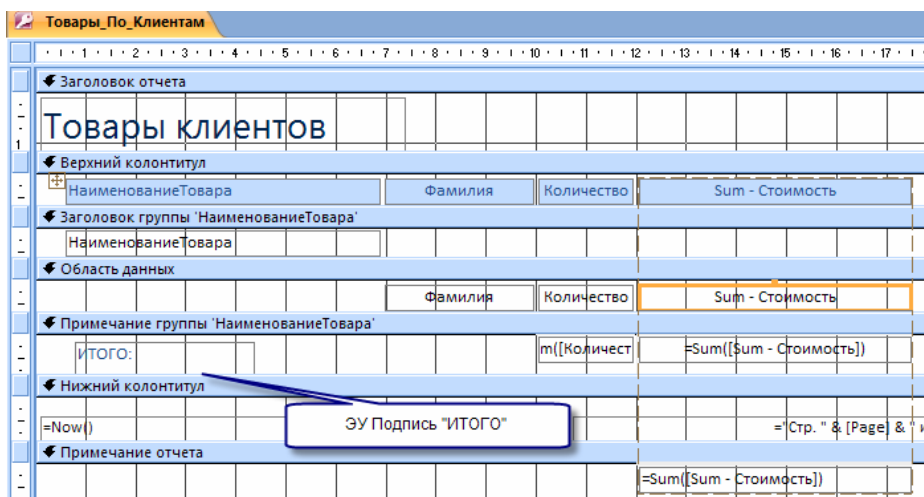


Рис. 4.2. Отчет «Товары_По_Клиентам», режим Конструктора.

12.Посмотрите результаты работы отчета. Он должен выглядеть аналогично рис. 4.3.

13.Сохраните отчет «Товары_По_Клиентам».

Товары клиентов

Наименование товара	Количество	Фамилия	Sum - Стоимость
"Бдительная сова"(против насекомых)	2	Вилламс	11,78р.
ИТОГО:	2		11,78р.
"Мартин Хаус" (против насекомых)	1	Клайтон	70,00р.
	1	Смит	67,90р.
ИТОГО:	2		137,90р.
"Улей" (против насекомых)	1	Никонен	13,58р.
ИТОГО:	1		13,58р.

Рис. 4.3. Отчет «Товары_По_Клиентам».

Создание ленточного отчета

Задание: создать отчет «Список_Товаров», предоставляющий информацию о товарах и ценах на товары.

1. На вкладке **Создание (Create)** в группе **Отчеты (Reports)** выберите **Конструктор отчетов (Reports Wizard)**.
2. В окне **Список полей (Field List)** выберите таблицу «Товары», поля «НаименованиеТовара», «ЛатинскоеНазвание», «ЦенаЗаУпаковку».
3. Разместите их в области данных друг под другом.
4. Выделите все элементы управления, расположенные в области данных отчета.
5. В режиме **конструктора (Design)** на вкладке **Упорядочить (Arrange)** в группе **Таблица (Table)** выберите макет расположения элементов

управления – **В виде таблицы (Tabular)**. При этом подписи полей должны переместиться в область верхнего колонтитула отчета (рис. 4.4).

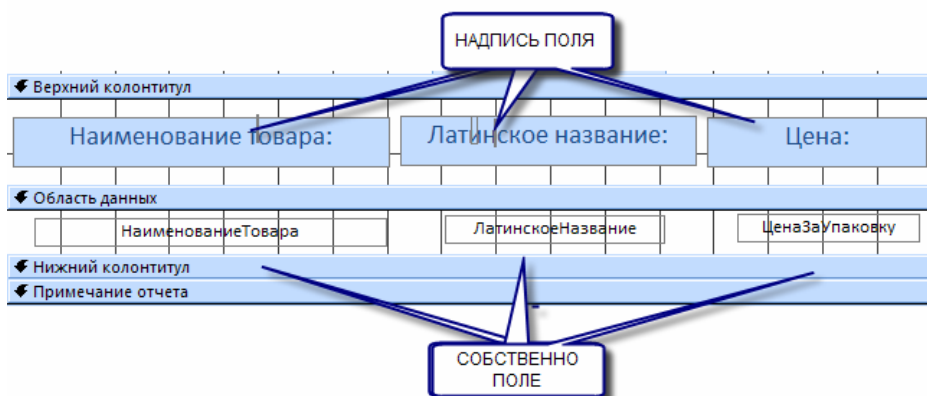


Рис. 4.4. Отчет «Список_Товаров», режим Конструктора.

6. С помощью **Окна свойств (Property Sheet)** откорректируйте вид формы (шрифты, цвет, размер элементов и т.п.).
7. В **режиме макета(Layout View)** добавьте сортировку от минимального к максимальному для поля «ЦенаЗаУпаковку».
8. В **режиме макета (Layout)** добавьте эмблему в отчет (файл *Эмблема.bmp*), щелкнув на элемент управления **Эмблема** в группе **Элементы управления** на вкладке **Конструктор (Design)**.
9. Просмотрите результаты отчета. Они должны выглядеть аналогично рис.4.5.

Наименование товара:	Латинское название:	Цена:
Садовые цветы		0,00р.
Глиняный цветочный горшок 6"		1,75р.
Лаванда	Lavender	2,25р.
Ориган	Monarda didyma	2,75р.
Держатель для шланга		2,95р.

Рис. 4.5. Отчет «Список_Товаров».

10. Сохраните отчет под именем «Список_Товаров».

Задание: создать телефонную книгу, содержащую все контакты компании.

1. На основании запроса «Список_Телефонов» **Мастером отчетов(Reports Wizard)** создайте отчет с именем «Телефонная_Книга» (группировки нет, сортировки нет).
2. В **режиме конструктора (Design)** добавьте **группировку** (вкладка **Конструктор (Design)**, группа **Группировка и итоги (Grouping&Totals)**, кнопка **Группировка (Grouping)**). Появится окно **Группировка**,

сортировка и итоги (Group, Sort And Total). В данном окне добавьте группировку по полю «*ФамилияИмя*»; нажмите **Больше(More)** и добавьте группировку **По первому знаку**.

3. В поле заголовка группы *ФамилияИмя* добавьте элемент управления **Поле (Field)**. Удалите подпись поля.
4. С помощью **Построителя выражений (Expression Builder)** (*Ctrl+F2* или *Окно свойств поля, вкладка Данные, поле Данные*) в созданное поле введите выражение:

=UCase (Left (Nz[*ФамилияИмя*]); 1))

Имя поля «*ФамилияИмя*» подставляется из создаваемого отчета «*Телефонная_Книга*».

Телефонная Книга	
← Верхний колонтитул	
ФамилияИмя	НомерТелефона
← Заголовок группы 'ФамилияИмя'	
=UCase(Left(Nz([ФамилияИмя]);1))	
← Область данных	
ФамилияИмя	НомерТелефона
← Нижний колонтитул	
=Now()	
← Примечание отчета	

Рис.4.6. Отчет «Телефонная_Книга» в режиме Конструктора

5. Изучите отчет в режиме отчета. Откорректируйте его внешний вид.

Телефонная Книга	
Фамилия Имя	Номер Телефона
A	
Албук Стив	(604) 555-0192
Андерсон Эмми	(206) 555-0105
Аскерман Пилар	(206) 555-0194
Аткинсон Тереза	(425) 555-0191
Акерс Ким	(604) 555-0193
B	
Белвин Амин	(425) 555-0190
Бенсон Макс	(360) 555-0189

Рис. 4.7. Отчет «Телефонная_Книга».

Отчет, получающий параметры через форму

Задание: создать форму, динамически формирующую отчет по товарам, относящимся к выбранной в форме категории (рис. 4.8-4.11).

Выберите интересующую категорию товаров:

Цветы

Сформировать отчет

Рис. 4.8. Форма для динамического формирования отчета

Категории товаров

Название категории	Код товара	Наименование Товара	Латинское название	Код поставщика	Количество в упаковке	Цена за упаковку	Количество на складе
Цветы							
	127	Жимолость	Loncera s.	17	1 шт.	5,50р.	5
	126	Маргаритка	Leucanthamum	17	1 шт.	3,50р.	9
	121	Вороний глаз	Paris quadrifolia L	17	1 шт.	3,25р.	23
	81	Гортензия метельчатая	H. paniculata	17	1 шт.	40,00р.	5
	80	Ландыш майский	Convallaria majalis	17	3 шт. в пачке	33,00р.	4

19 марта 2010 г.

Стр. 1 из 1

Рис. 4.9. Отчет, динамически сформированный на основании полученных из формы данных.

1. Сформируйте простой запрос, включающий следующие поля:

Название таблицы	Название поля
Категории Товаров	«НазваниеКатегории»
Товары	«КодТовара», «НаименованиеТовара», «ЛатинскоеНазвание», «КодПоставщика», «Кол-воВУпаковке», «ЦенаЗаУпаковку», «Кол-воНаСкладе», «ТоваровНаСумму»

2. Сохраните запрос под именем «Категории_Товары».
3. На основе только что созданного запроса «Категории_Товары» сформируйте отчет, включающий все поля запроса с группировкой по **Названию категории**.
4. Сохраните отчет под именем «Категории_Товары».
5. Создайте несвязную форму для задания условий отбора данных в отчет.
 - 5.1. Поместите на форму (с помощью мастера элементов управления (Control Wizard)) элемент управления **Поле со списком (Combo Box)**, указав, что подстановка значений будет осуществляться из таблицы «Категории_Товаров», поле «НазваниеКатегории», **сортировка** - по названию категории.
 - 5.2. В **Окне свойств (Property Sheet) Поля со списком** на вкладке **Данные (Data)** укажите, что номер присоединенного столбца таблицы – 2.

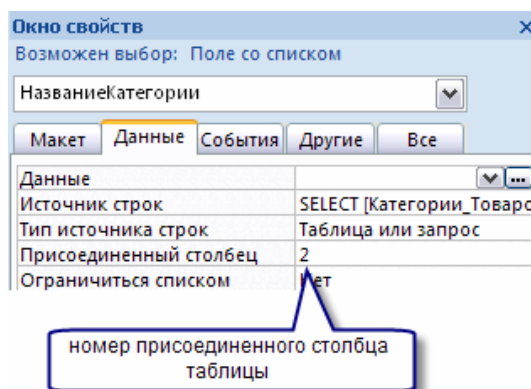


Рис. 4.10. Окно свойств поля со списком

5.3. Присвойте имя **Полю со списком** – *НазваниеКатегории*.

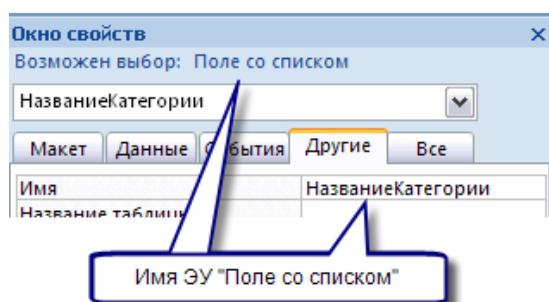


Рис. 4.11. Окно свойств поля со списком

5.4. Сохраните форму под именем **«Выбор_Категории»**.

6. Откройте запрос **«Категории_Товары»** в режиме конструктора (Design) и для поля **«НазваниеКатегории»** укажите условие отбора (Criteria):

[Forms]![Выбор_Категории]![НазваниеКатегории]

Данное условие позволяет формировать в запросе список товаров, относящихся к категории, которая выбрана в форме **«Выбор_Категории»** (по полю **«НазваниеКатегории»**). Затем сформированный список передается в отчет **«Категории_Товары»**, созданный на основании данного запроса.

7. Сохраните измененный запрос.

8. Для открытия динамически сформированного отчета откройте форму **«Выбор_Категории»** в режиме конструктора (Design).

8.1. При включенном мастере элементов управления (Control Wizard) поместите на поле элемент управления **Кнопка (Button)**, выберите **категорию действий** – Работа с отчетом, **действие** – открыть отчет **«Категории_Товары»**. Поместите на кнопку текст **«Сформировать отчет»**.

9. Перейдите в **режим формы (Form View)** и проверьте работу формы.

10. Сохраните форму, подтвердив изменения.

Задание: аналогичным образом создайте форму, динамически формирующую отчет, содержащий список всех заказов, выполняемых конкретным сотрудником организации.

Вид формы:

Рис.4.12. Форма для динамического формирования отчета

Вид отчета:

Выполняемые заказы					
Фамилия	Код заказа	Код клиента	Дата заказа	Требуемая дата выполнения	Ожидаемая Дата Выполнения
Каплан	11173	ЯНОБО	16.03.2010	31.03.2010	31.03.2010
	11168	ЗИМКЕ	05.03.2010	19.03.2010	
	11081	БЕНМА	06.01.2010		07.01.2001
Всего заказов		3			

25 апреля 2010 г. Подсчет количества выполняемых заказов Стр. 1 из 1

Рис.4.13. Отчет, динамически сформированный на основании полученных из формы данных.

1. Создайте запрос «*Сотрудники_Заказы*», выбрав **НЕОБХОДИМЫЕ** поля из таблиц «*Сотрудники*» и «*Заказы*» (изучите пример сформированного отчета).
2. На основании запроса «*Сотрудники_Заказы*» создайте отчет «*Сотрудники_Заказы*», задав необходимую группировку и сортировку.
3. Создайте несвязную форму «*Сотрудники_Заказы*» для задания условий динамического формирования данных в отчете. На форму поместите все необходимые элементы управления (ЭУ), определите необходимые свойства ЭУ (для ЭУ **Поле со списком** задайте имя – *Сотрудники*).
4. В запросе «*Сотрудники_Заказы*» укажите условия отбора данных для динамически формируемого отчета.
5. В отчете «*Сотрудники_Заказы*» добавьте подсчет количества выполняемых заказов каждым сотрудником.
 - 5.1. Решите, какое поле в отчете будет присутствовать всегда (кроме поля, по которому проводится группировка) и задайте в **примечании группы Итоги** для этого поля **по количеству записей**.

- 5.2. В режиме **конструктора** добавьте подпись «*Всего заказов*» в примечание группы.
6. Оформите отчет и форму:
 - 6.1. Добавьте эмблему организации, работая со свойствами формы/отчета (файл «*Эмблема.bmp*»);
 - 6.2. Откорректируйте заголовки формы и отчета, колонтитулы отчета, размер и положение элементов и т.п.

Сохранение и закрытие базы данных

1. Сохраните все изменения.
2. Завершите работу с Microsoft Access.


Лабораторная работа №5: Макросы

Макрос — это средство для автоматизации задач и добавления функциональных возможностей в формы, отчеты и элементы управления.

В приложении Access макросы можно рассматривать как упрощенный язык программирования, на котором программа записывается в виде списка макрокоманд для выполнения. При создании макроса каждая макрокоманда выбирается из раскрывающегося списка, после чего к ней добавляется необходимая информация.

Макрос создается с помощью **конструктора макросов (Macro Design)**. Чтобы отобразить построитель макросов на вкладке **Создание (Create)** в группе **Макросы и код (Macro And Code)** щелкните **Макрос (Macro)**.

Создание изолированного макроса

3. На вкладке **Создание (Create)** в группе **Макросы и код (Macro And Code)** щелкните **Макрос(Macro)**. . Откроется **конструктор макросов(Macro Design)**.
4. Для добавления макрокоманды в макрос в **конструкторе макросов(Macro Design)** выберите макрос **ОкноСообщения** из выпадающего меню **Добавить новую макрокоманду (Create)** или перетащите макрокоманду из каталога макрокоманд из списка справа. Для поиска нужных макрокоманд рекомендуется использовать окно поиска в каталоге макрокоманд.

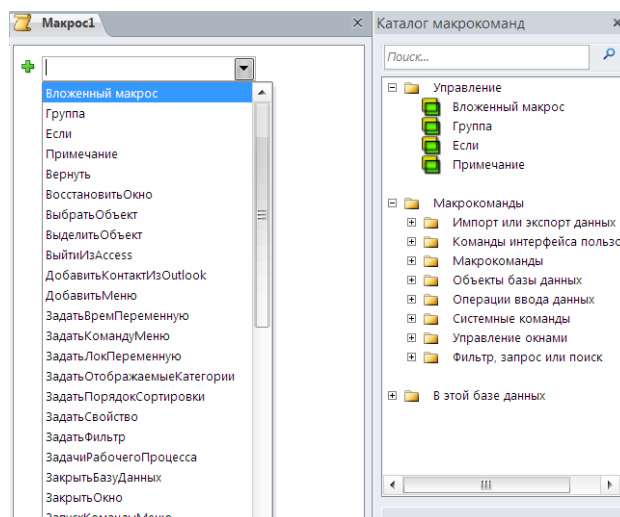


Рис. 5.1. Выбор макрокоманды

5. Укажите следующие аргументы для макрокоманды:

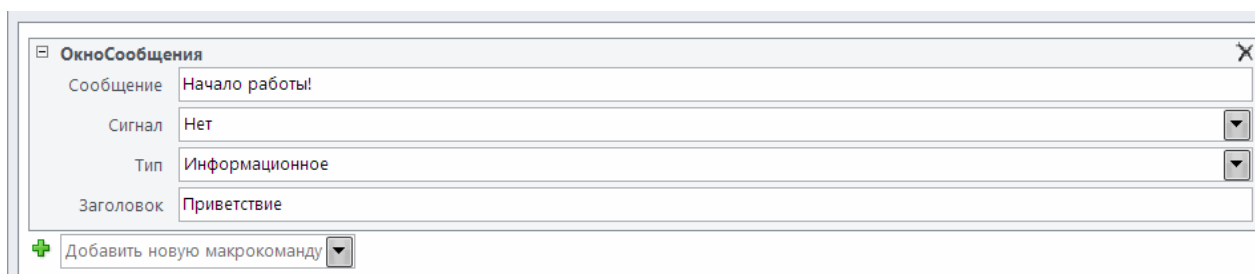



Рис. 5.2. Аргументы макрокоманды

6. Выполните макрос с помощью кнопки **Выполнить (Run)** . Подтвердите сохранение макроса, присвоив ему имя «**Сообщения**».
7. Переименуйте макрос, дав ему имя **Autoexec**.
8. Закройте базу данных, затем снова откройте ее и убедитесь, что теперь макрос выполняется автоматически при открытии базы.
9. Верните макросу прежнее имя.

Работа с вложенными макросами

Вложенные макросы разрешают использование в макросе именованных групп макрокоманд, которые могут быть вызваны только их непосредственным запуском.

1. Создание вложенных макросов
 - 1.1. Откройте макрос «**Сообщения**» в **режиме конструктора (Design)**.
 - 1.2. Добавьте **Вложенный макрос** из каталога макрокоманд.
 - 1.3. Присвойте имя вложенному макросу «**Окончание**» и добавьте в окно вложенного макроса макрокоманду **ОкноСообщения** со следующими аргументами:

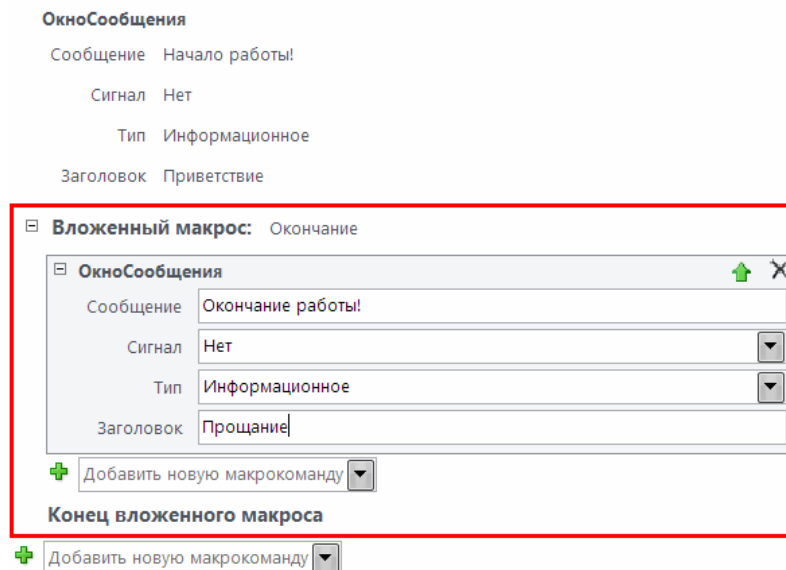


Рис. 5.3. Вложенный макрос

- 1.4. Сохраните изменения и закройте окно макроса.
- 1.5. Выполните макрос из окна базы данных. Будет выполнен только первый макрос из созданных.
2. Связь группы макросов с формой (работа с формой «Товары»).
 - 2.1. Откройте форму «Товары» в режиме конструктора (Design).
 - 2.2. Откройте **Окно свойств формы (Property Sheet)** (F4 или кнопка **Страница свойств (Property Sheet)** в группе **Сервис (Tools)**), выберите вкладку **Свойств формы - События (Events)**.
 - 2.3. Для свойства **Открытие (On open)** – выберите из списка макрос «Сообщения».
 - 2.4. Для свойства **Закрытие (on Close)** – выберите макрос «Сообщения.Окончание».
 - 2.5. Сохраните и закройте форму. Затем откройте форму «Товары» (выполнится основной макрос «Сообщения», а потом закройте (выполнится вложенный макрос «Сообщения.Окончание»)).

Экспорт данных в различные форматы

1. Создайте макрос, выполняющий макрокоманду **ЭкспортСФорматированием**.
2. Аргументы макрокоманды (Arguments):
 - тип объекта – форма;
 - имя объекта – Список_Заказов_По_Датам;
 - формат вывода – оставьте пустым
 - имя файла – ... <путь к файлу>.<расширение выбранного формата>;
 - качество вывода – печать.

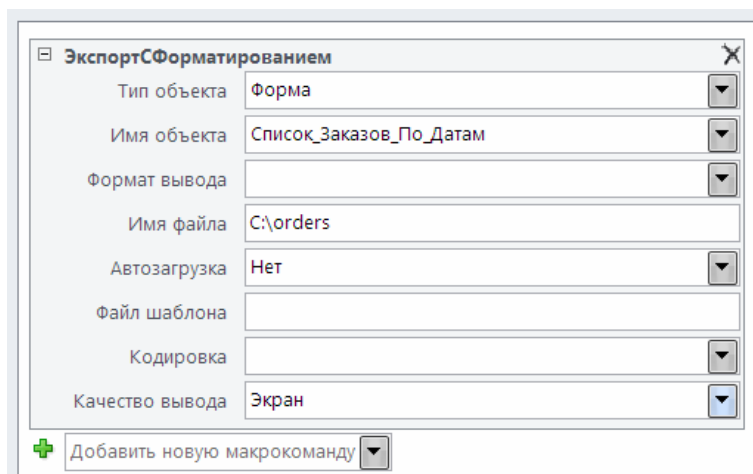


Рис. 5.4. Аргументы макрокоманды ЭкспортСФорматированием

3. Сохраните макрос с каким-либо форматом вывода (например, .pdf) под именем **«ФорматВывода»**.
4. Откройте форму **«Список_Заказов_По_Датам»** в режиме конструктора(**Design**) и добавьте на ее элемент управления **Кнопка (Button)**.
5. В **Окне свойств (Property Sheet)** элемента управления **Кнопка (Button)** выберите вкладку **События (Events)**, и свяжите событие **Нажатие кнопки (PressButton)** с макросом **«ФорматВывода»**. Укажите подпись кнопки – **«Экспорт данных»**.
6. Перейдите в **режим формы (Form View)** и выполните макрос. Убедитесь в корректности его выполнения.

Создание внедренного макроса

Процесс создания внедренного макроса аналогичен созданию стандартного макроса. Основное отличие состоит в том, что макрос внедряется в объект, с которым он связан, и не появляется в списке макросов в окне области переходов.

1. Откройте отчет **«Транспортная_Компания»** в режиме конструктора(**Design**).
2. Разместите на форме отчета элемент управления **Кнопка (Button)**.
3. Откройте **Окно свойств (Property Sheet)** для элемента управления **Кнопка (Button)**, вкладку **События(Events)**, событие **Нажатие кнопки (PressButton)** и выберите **Построитель (Choose Builder)** (нажав кнопку с троеточием). В открывшемся диалоговом окне **Построителя (Choose Builder)** выберите элемент **Макросы (Macro Builder)** – **ОК**. Откроется окно **Конструктора макросов (Macro Design)**.
4. Выберите макрокоманду **ОкноСообщения (MsgBox)**, произвольно назначьте **аргументы (Arguments)** (текст сообщения, тип сообщения и т.п.) и выполните макрос, сохранив его.

5. Обратите внимание, что созданных внедренный макрос не отобразился в списке макросов.

Задание: с помощью встроенных макросов обеспечить возможность перехода между объектами базы данных, а также закрытие объектов.

1. Откройте форму «*Доставка*» в режиме конструктора(**Design**).
2. На вкладке **Конструктор (Design)** в группе **Элемент управления (Controls)** активизируйте кнопку **Использование мастера (Control Wizard)** и выберите элемент управления **Кнопка (Button)**.
3. Разместите кнопку на форме. Откроется окно создания кнопок. Выберите категорию «*Работа с формой*», действие «*Открыть форму*» - **Далее** – Выберите из списка форму «*Транспорт*» - **Далее (Next)** – **Далее(Next)** - Текст «Транспорт» - **Готово (Finish)**. Откроется форма «*Транспорт*».
4. Откройте форму в режиме формы (**Form View**), нажмите кнопку «Транспорт».
5. Откройте форму «*Транспорт*» в режиме конструктора (**Design**) и поместите на форму элемент управления **Кнопка (Button)**. В окне **Создание кнопок** выберите категорию «*Работа с формой*», действие «*Закрыть форму*» - **Далее(Next)** – Разместите на кнопке текст или рисунок – **Готово (Finish)**.
6. Перейдите в режим формы (**Form View**) и нажмите на созданную кнопку. Форма «*Транспорт*» закроется.
7. Убедитесь, что ни один из созданных внедренных макросов не отобразился в списке макросов.

Создание макроса данных

Макросы данных — это новая функция Access 2010, которая позволяет добавлять логику в события, которые происходят в таблицах (например, добавление, обновление или удаление данных).

Задание: необходимо реализовать частичную автоматизацию отслеживания готовности заказа. Для этого необходимо добавить в таблицу «*Заказы*» дополнительные поля, создать форму, выводящую соответствующую информацию, а также создать макрос, частично автоматизирующий отслеживание готовности заказа.

1. В таблицу «*Заказы*» добавьте два поля:
 - a. «*СостояниеЗаказа*»: тип – текстовый; использовать подстановку - Список, тип источника строк – Список значений.
 - b. «*ПроцентЗавершения*»: тип – числовой; формат поля – длинное целое.

2. Перейдите в режим таблицы, создайте элементы списка поля «СостояниеЗаказа». Для этого кликните по пустой ячейке поля и откройте окно «Изменить элементы списка».

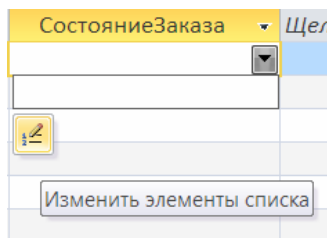


Рис.5.5.Создание элементов списка поля таблицы

Добавьте элементы «Не начат», «Выполняется», «Завершен»:

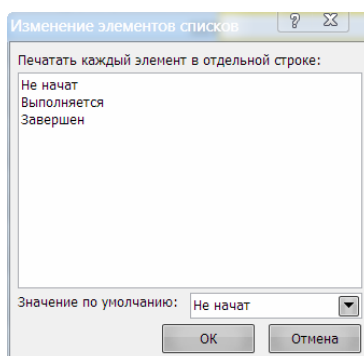


Рис. 5.6. Окно изменения элементов списка поля таблицы

3. Сохраните изменения, закройте таблицу.
4. На основании измененной таблицы «Заказы» создайте форму «Состояние_Заказов», включив в нее следующие поля: «КодЗаказа», «ДатаЗаказа», «СостояниеЗаказа», «ПроцентЗавершения».
5. Сохраните изменения, закройте форму.
6. Откройте таблицу «Заказы» в режиме таблицы. Перейдите на вкладку **Таблицы**, выберите макрос **До изменения**. Откроется окно макросов.
7. Выберите команду условия **Если** в каталоге макросов, раздел **Управление**. Введите критерий и его значение:

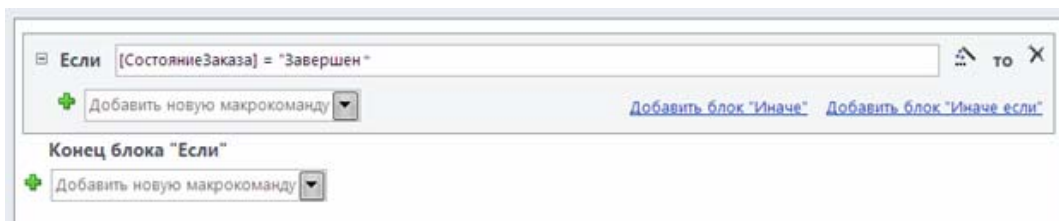


Рис. 5.7. Параметры условия макроса Если

8. Выберите макрокоманду **ЗадатьПоле** и добавьте ее в блок **Если**. Введите следующие аргументы:

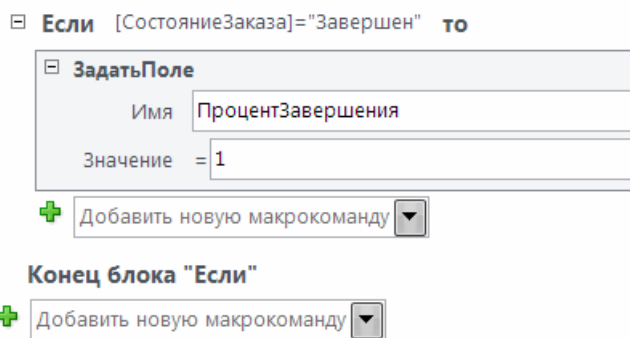


Рис.5.8. Параметры макрокоманды *ЗадатьПоле*

Далее необходимо добавить команду, присваивающую полю «ПроцентЗавершения» значение ноль при задании значения «Не начато» поля **Состояние**. Для этого добавьте оператор **Иначе Если** к оператору **Если** (для этого сначала выберите оператор **Если**, а затем щелкните на команде **Иначе Если**). Добавьте аргументы команды:

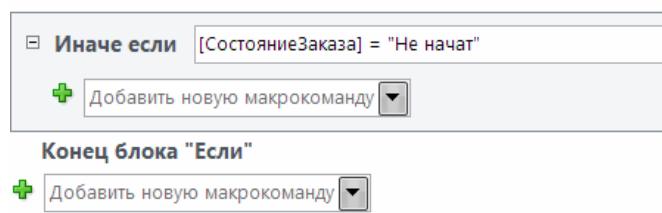


Рис. 5.9. Параметры условия *Иначе Если*

9. Затем добавьте команду *ЗадатьПоле*:

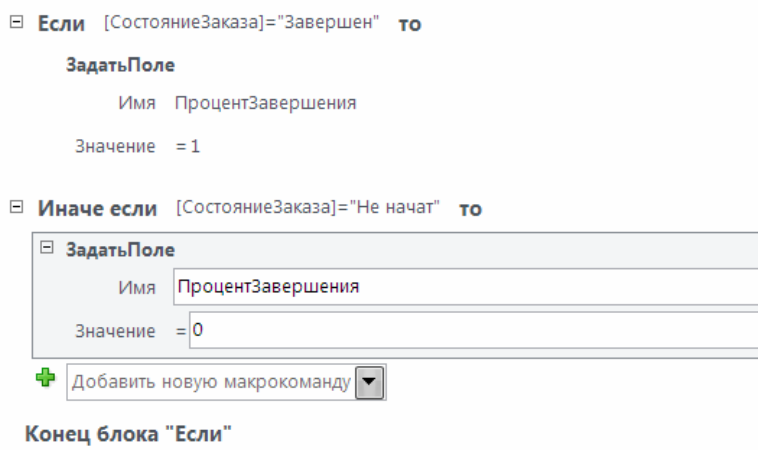


Рис.5.10. Параметры макрокоманды *ЗадатьПоле*

10. Сохраните изменения. Проверьте корректность работы макроса в таблице и в форме.

Создание макроса пользовательского интерфейса

Задание: необходимо создать форму, выводящую список клиентов в табличном виде с именем «*Список_Клиентов*», и реализовать макрос, который бы по нажатию на код клиента открывал форму с информацией по выбранному клиенту.

1. На основании таблицы «*Клиенты*» создайте форму «*Список_Клиентов*», которая выводила бы информацию о клиентах в табличном виде:

Код клиент	Имя	Фамилия	Адрес	Город	Штат	Почтовый индекс	Страна	(999) 999-9
АТКТЕ	Тереза	Аткинсон	89 Кедровая ул.	Портленд	Вашингтон	30293	США	(425) 555-019
АКЕКИ	Ким	Акерс	1932 52-ая Авеню	Ванкувер	Британская К	V3G 2T1	Канада	(604) 555-019
АЛБСТ	Стив	Албук	130 17-ая ул.	Ванкувер	Британская К	V3Y 2K1	Канада	(604) 555-019
АСКПИ	Пилар	Аскерман	8808 Бай-стрит	Олимпия	Вашингтон	88337	США	(206) 555-019

Рис.5.11. Форма «Список_Клиентов»

2. Необходимо создать такой внедренный макрос, который бы по щелчку мыши по коду клиента в форме «*Список_Клиентов*» выводил информацию о клиенте в более удобном виде из формы «*Клиенты*».

Рис. 5.12 Обмен данными между формами приложения

3. Откройте форму «*Список_Клиентов*» в режиме таблицы. Выделите поле с кодом клиента и откройте окно свойств.
4. В окне свойств перейдите на вкладку **Событие**, пункт **Нажатие кнопки** – **Макрос**. Откроется окно **конструктора макросов**.
5. В окне **конструктора макросов** выберите макрокоманду **ОткрытьФорму**, задайте для нее следующие параметры:

ОткрытьФорму	
Имя формы	Клиенты
Режим	Форма
Имя фильтра	
Условие отбора	= [КодКлиента]=[Формы]![Список_Клиентов]![КодКлиента]
Режим данных	Изменение
Режим окна	Окно диалога

Добавить новую макрокоманду

Рис. 5.13. Параметры макрокоманды **ОткрытьФорму**

6. Сохраните изменения, проверьте корректность работы макроса.

Сохранение и закрытие базы данных

1. Сохраните все изменения.
2. Завершите работу с Microsoft Access.

Лабораторная работа №6: Подготовка приложения к внедрению

До передачи приложения в эксплуатацию должны быть осуществлены следующие подготовительные действия:

1. Приложение должно быть основано исключительно на использовании форм и отчетов.
2. Необходимо предусмотреть параметры запуска базы данных и выхода из нее.
3. Приложение должно обеспечивать защиту данных.

Создание формы навигации

1. Создание навигационной формы включает в себя создание трех форм:
 - a. Форма навигации по созданным формам;
 - b. Форма навигации по созданным отчетам;
 - c. Собственно форма навигации.

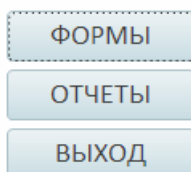


Рис 6.1. Кнопки главной навигационной формы.

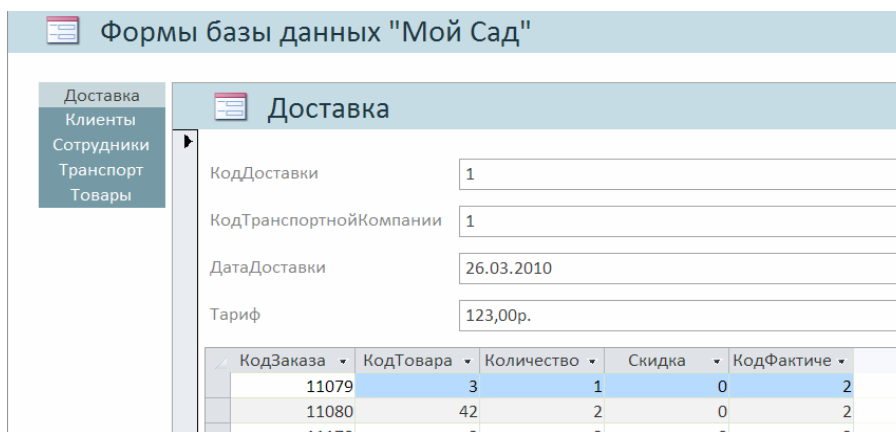


Рис. 6.2. Пример оформления формы навигации по формам базы данных

2. Для их разработки на вкладке **Создание** в группе **Формы** выберите **Навигация**.
3. Выберите макеты форм, разместите на них соответствующие вкладки для форм и отчетов.
4. После разработки форм перейдите на вкладку представления **Backstage** (вкладка **Файл**) – **Параметры** - **Текущая база данных** – **Параметры приложений** – **Значок приложений**: установите эмблему, которая использовалась в ходе разработки приложения (выберите файл Эмблема.bmp). Установите флаг **Значок форм и отчетов**. Сохраните изменения. Обратите внимание, что на вкладках форм и отчетов появилась установленная эмблема.
5. Установите следующие свойства формы: **кнопка закрытия** – нет; **кнопки размеров окна** – отсутствуют; **допускается перемещение** – нет.
6. Отредактируйте внешний вид формы навигации. Добавьте на форму фоновый рисунок – эмблему организации.

Создание макроса *AutoExec*

Макрос *AutoExec* автоматически запускается при старте приложения и выполняет открытие заданного объекта (например, формы) базы данных, а также блокирует область переходов.

1. Создайте новый макрос, выполняющий команду **ОткрытьФорму**. В качестве параметра задайте имя формы навигации по приложению.
2. Сохраните макрос под именем *AutoExec*.
3. Закройте базу данных и откройте ее вновь. Выполнится макрос *AutoExec*.
4. Для обходов параметров запуска (макроса *AutoExec*) необходимо при открытии базы данных удерживать нажатой клавишу SHIFT.

Работа с областью навигации приложения

Закройте для пользователей область навигации приложения: перейдите на вкладку представления **Backstage** (вкладка **Файл**) – **Параметры - Текущая база данных – Навигация** – снимите галочку с пункта **Область навигации**.

Обеспечение защиты информации

1. Если база данных открыта – закройте ее, но не выходите из приложения MS Access.
2. Щелкните на вкладке **Файл** и выберите команду **Открыть (Open)**.
3. В диалоговом окне **Открытие (Open)** найдите файл Вашей базы данных. Щелкните на стрелке рядом с кнопкой **Открыть (Open)** и в выпадающем списке выберите команду **Монопольно (Open Exclusive)**.
4. Перейдите на вкладку **Файл** - меню **Сведения** – **Зашифровать паролем**.

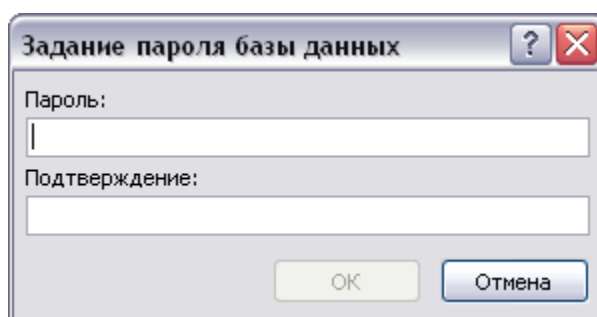


Рис. 6.3. Окно Задания пароля базы данных.

5. В поле **Пароль (Password)** введите пароль. Подтвердите его в поле **Подтверждение (Verify)** – **ОК**.
6. Закройте базу данных и откройте снова. Откроется диалоговое окно **Необходимо ввести пароль (Password Required)**. Введите пароль.
7. Чтобы удалить пароль, откройте базу данных в **монопольном режиме**, при открытии введите пароль. На вкладке **Файл** в меню **Сведения** щелкните на кнопке **Расшифровать базу данных (Decrypt Database)**. Введите пароль и щелкните **ОК**. Пароль будет удален.

Сохранение и закрытие базы данных

1. Сохраните все изменения.
2. Завершите работу с Microsoft Access.

Литература

1. Codd E.F. Normalized Data Base Structure: A Brief Tutorial //Proc. of 1971 ACM SIGFIDET Workshop on Data Description, Access and Control.- N.-Y.: ACM. - 1971. - P.1-17.
2. Groh M. Access 2010 Bible. Canada:Wiley Publishing, Inc., 2010 – 1395 p.
3. Jennings R. Microsoft Access 2010 In Depth. QUE, 2011 – 1200 p.
4. Алекс Кригель, Борис Трухнов. SQL. Библия пользователя. Язык запросов SQL 2-е издание. Диалектика.2009 - 752 стр.
5. Гурвиц Г. Microsoft Access 2010. Разработка приложений на реальном примере. СПб.: БХВ-СПб, 2010 – 469 с.
6. Дейт К. SQL и реляционная теория. Как грамотно писать код на SQL. М.: Символ-Плюс, 2010 – 480 с.
7. Кляйн К., Кляйн Д., Хант Б. SQL. Справочник, 3-е издание. М.: Символ-Плюс, 2009 – 656 с.
8. Коннолли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. М.: Вильямс, 2001 – 1120 с.
9. Кренке Д. Теория и практика построения баз данных. СПб.: Питер, 2005 – 864 с.
10. Сеннов А. Access 2010. СПб.: Питер, 2010 – 288 с.

Приложение 1. Описание таблиц

Таблица «Заказы»

Имя поля	Тип	Размер	Другие свойства
КодЗаказа	Счетчик	Длинное целое	Ключ
КодКлиента	Текстовый	255	Обязательное поле; Индексированное – Да (допускаются совпадения)
КодСотрудника	Числовой	Длинное целое	Обязательное поле; Индексированное – Да (допускаются совпадения)
ДатаЗаказа	Дата/время		Маска ввода - краткий формат даты
ТребуемаяДатаВыполнения	Дата/время		Отображать элемент выбора даты (Show Date Picker) – для дат
ОжидаемаяДатаВыполнения	Дата/время		Отображать элемент выбора даты (Show Date Picker) – для дат
НомерСчетаКлиента	Текстовый	15	Индексированное – Да (совпадения допускаются)

Таблица «Доставка»

Имя поля	Тип	Размер	Другие свойства
КодДоставки	Счетчик	Длинное целое	Ключ
КодТранспортнойКомпании	Числовой	Длинное целое	Обязательное поле
ДатаДоставки	Дата/время		Обязательное поле; Отображать элемент выбора даты (Show Date Picker) – для дат
Тариф	Денежный		Число десятичных знаков – 2.

Таблица «Фактический Адрес Доставки»

Имя поля	Тип	Размер	Другие свойства
КодФактАдреса	Счетчик	Длинное целое	Ключ
КодКлиента	Текстовый	255	Обязательное поле
Регион	Текстовый	50	Обязательное
Город	Текстовый	20	Обязательное
Улица	Текстовый	50	Обязательное
Дом	Текстовый	4	Обязательное
Корпус/строение	Текстовый	5	Необязательное
Индекс	Текстовый	7	Маска ввода 9999999;;
Телефон	Текстовый	14	Маска ввода (999) 000-0000;

Таблица «Поставщики»

Имя поля	Тип	Размер	Другие свойства
КодПоставщика	Счетчик	Длинное целое	Ключ
Организация	Текстовый	255	Обязательное, Индексированное – да (совпадения не допускаются)
Фамилия	Текстовый	20	Обязательное
Имя	Текстовый	20	Обязательное
Должность	Текстовый	50	Обязательное
РабочийТелефон	Текстовый	14	Маска ввода (999) 000-0000
МобильныйТелефон	Текстовый	14	Обязательное, Маска ввода (999) 000-0000
Факс	Текстовый	14	Маска ввода 000-0000
Адрес	Текстовый	100	Обязательное
Город	Текстовый	20	Обязательное
Штат	Текстовый	20	Обязательное
Индекс	Текстовый	7	Маска ввода 9999999;;

Таблица «Сотрудники»

Имя поля	Тип	Размер	Другие свойства
КодСотрудника	Счетчик	Длинное целое	Ключ
Имя	Текстовый	20	Обязательное
Фамилия	Текстовый	20	Обязательное поле; Индексированное – Да (совпадения допускаются)
Должность	Текстовый	50	Обязательное поле;

			Индексированное – Да (совпадения допускаются)
Адрес	Текстовый	100	Обязательное
Город	Текстовый	20	Обязательное
Штат	Текстовый	20	Обязательное
Индекс	Текстовый	7	Маска ввода 9999999;;
Телефон	Текстовый	14	Обязательное, Маска ввода (999) 000-0000
ДатаРождения	Дата/время		Обязательное поле; Маска ввода – краткий формат даты.
ДатаНайма	Дата/время		Обязательное поле; Условие на значение <=Now(); Сообщение об ошибке – «Дата должна быть не позднее текущей». Отображать элемент выбора даты (Show Date Picker) – для дат
Заметки	Поле МЕМО		Необязательное

Таблица «Транспортная Компания»

Имя поля	Тип	Размер	Другие свойства
КодТранспортнойКомпании	Счетчик	Длинное целое	Ключ
НазваниеКомпании	Текстовый	255	Обязательное, Индексированное – да (совпадения не допускаются)
Фамилия	Текстовый	20	Обязательное
Имя	Текстовый	20	Обязательное
Улица	Текстовый	50	Обязательное
Город	Текстовый	20	Обязательное
Штат	Текстовый	20	Обязательное
Индекс	Текстовый	7	Маска ввода 9999999;;
Телефон	Текстовый	14	Обязательное, Маска ввода (999) 000-0000

Таблица «*Подробности Заказа*»

Имя поля	Тип	Размер	Другие свойства
КодЗаказа	Числовой	Длинное целое	Составной ключ; Обязательное; Индексированное – ДА (допускаются совпадения)
КодТовара	Числовой	Длинное целое	
Количество	Числовой	Целое	Обязательное; Значение по умолчанию – 0
Скидка	Числовой	Двойное с плавающей точкой	Необязательное; Формат поля – процентный, число десятичных знаков – 0; значение по умолчанию – 0.
КодФактическогоАдреса	Числовой	Длинное целое	Необязательное поле
КодДоставки	Числовой	Длинное целое	Необязательное поле

Таблица «*Клиенты*»

Имя поля	Тип	Размер	Другие свойства
КодКлиента	Текстовый	5	Ключ
Имя	Текстовый	30	Обязательное
Фамилия	Текстовый	50	Обязательное
Адрес	Текстовый	255	Обязательное
Город	Текстовый	50	Обязательное
Штат	Текстовый	20	Обязательное
ПочтовыйИндекс	Текстовый	7	Маска ввода 9999999;
Страна	Текстовый	50	Обязательное
НомерТелефона	Текстовый	15	Маска ввода (999) 999- 9999
КредитнаяКарта	Числовой	Байт	Значение по умолчанию 0
НомерКредитнойКарты	Текстовый	255	Индексированное – ДА (совпадения не допускаются)

Приложение 2. Маска ввода

Маска ввода представляет собой набор текстовых констант и знаков маски, определяющий, какие данные и каким образом можно вводить. Например, с помощью маски ввода можно потребовать, чтобы пользователи вводили даты или телефонные номера так, как принято в определенной стране, например:

- ДД.ММ.ГГГГ
- () - - доб. -

Маска ввода может состоять из трех частей, которые разделены точкой с запятой. Первая часть является обязательной, а две остальные — необязательными. Пример маски ввода для телефонных номеров в формате США:

(999) 000-0000;0;_

Первая часть определяет строку маски и состоит из местозаполнителей и текстовых констант. Вторая часть указывает, следует ли сохранять знаки маски *вместе* с данными в базе данных. Введите **0**, если в базе данных необходимо сохранить маску и данные. Введите **1**, если необходимо сохранить только данные. Третья часть определяет местозаполнитель, который служит для обозначения позиции данных. По умолчанию в Access используется знак подчеркивания (_). Если нужно использовать другой знак, введите его в третью часть маски. По умолчанию позиция принимает только один знак или пробел.

Примеры масок ввода в следующей таблице демонстрируют некоторые способы применения масок ввода.

Маска ввода	Обеспечивает ввод значения в виде	Примечания
(000) 000-0000	(206) 555-0199	В данном случае пользователи должны вводить междугородный код, так как в этой области маски (000 в скобках) используется местозаполнитель 0.
(999) 000-0000!	(206) 555-0199 () 555-0199	В этом случае в области междугородного кода используется местозаполнитель 9, поэтому междугородные коды необязательны. Кроме того, восклицательный знак (!) обуславливает заполнение маски слева направо.

(000) AAA-AAAA	(206) 555-ТЕЛЕ	Предоставляется возможность заменить буквами четыре последних цифры телефонного номера в формате США. Обратите внимание на местозаполнитель 0 в области кода города, который делает междугородный код обязательным.
#999	-20 2000	Любое положительное или отрицательное число, включающее не более четырех знаков и не имеющее разделителей тысяч и дробной части.
>L????L?000L0	ЗЕЛЕНый339M3 МАЙ P 452B7	Сочетание обязательных (L) и необязательных (?) букв и обязательных цифр (0). Знак «больше» требует вводить все буквы в верхнем регистре. Чтобы использовать маску ввода этого типа, необходимо задать для типа данных поля таблицы значение Текстовый или Поле МЕМО .
00000-9999	98115- 98115-3007	Обязательный почтовый индекс и необязательная область четырехзначного расширения.
>L<??????????????	Мария Вадим	Имя или фамилия, первая буква которой автоматически вводится заглавной.
ISBN 0- &&&&&&&&-0	ISBN 1-55615-507- 7	Учетный номер книги с текстовой константой, обязательными первой и последней цифрами и произвольной комбинацией букв и знаков между ними.
>LL00000-0000	БД51392-0493	Сочетание обязательных букв и цифр, все в верхнем регистре. Данный тип маски ввода помогает, например, правильно вводить инвентарные номера или заполнять другие формы складского учета.

Символы, используемые для создания маски ввода	
Символ	Описание
0	Цифра (от 0 до 9, ввод обязателен). Символы плюс (+) и минус (-) не допускаются.

9	Цифра или пробел. Ввод не обязателен, символы плюс (+) и минус (-) не допускаются.
#	В эту позицию следует ввести цифру, пробел либо знак плюса или минуса. Если пользователь пропускает эту позицию, вставляется пробел.
L	Буква. В эту позицию обязательно следует ввести одну букву.
?	Буква. В эту позицию можно ввести одну букву, но необязательно.
A	Буква или цифра. В эту позицию обязательно следует ввести одну букву или цифру.
a	Буква или цифра. В эту позицию можно ввести одну букву или цифру, но необязательно.
&	Любой знак или пробел. В эту позицию обязательно следует ввести знак или пробел.
C	Любой знак или пробел. В эту позицию можно ввести знак или пробел, но необязательно.
., : ; - /	Разделитель целой и дробной части, групп разрядов, значений дат и времени. Выбираемый знак зависит от региональных параметров Windows.
>	Все последующие знаки отображаются в верхнем регистре.
<	Все последующие знаки отображаются в нижнем регистре.
!	Маска ввода заполняется слева направо, а не справа налево.
\	Указывает ввод любого следующего символа в качестве текстовой константы. Используется для отображения всех перечисленных в данной таблице символов как текстовых констант (например, \A выводится как символ «А»).
Пароль	В режиме конструктора таблиц или форм можно задать для свойства Маска ввода (Input Mask) значение Пароль ; при этом создается поле для ввода пароля. Когда пользователь вводит пароль в это поле, введенные знаки сохраняются, но вместо них отображаются звездочки (*).

Приложение 3. Схема данных БД «Мой Сад»





В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Кафедра Программных систем

Кафедра **Программных систем** входит в состав нового факультета **Инфокоммуникационные технологии**, созданного решением Ученого совета университета 17 декабря 2010 г. по предложению инициативной группы сотрудников, имеющих большой опыт в реализации инфокоммуникационных проектов федерального и регионального значения.

На кафедре ведется подготовка бакалавров и магистров по направлению **210700 «Инфокоммуникационные технологии и системы связи»:**

**210700.62.10 – ИНТЕЛЛЕКТУАЛЬНЫЕ
ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ (Бакалавр)**

**210700.68.10 – ИНТЕЛЛЕКТУАЛЬНЫЕ
ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ (Магистр)**

Выпускники кафедры получают фундаментальную подготовку по: математике, физике, электронике, моделированию и проектированию инфокоммуникационных систем (ИКС), информатике и программированию, теории связи и теории информации.

В рамках профессионального цикла изучаются дисциплины: архитектура ИКС, технологии программирования, ИКС в Интернете, сетевые технологии, администрирование сетей Windows и UNIX, создание программного обеспечения ИКС, Web программирование, создание клиент-серверных приложений.

Область профессиональной деятельности бакалавров и магистров включает:

- сервисно-эксплуатационная в сфере современных ИКС;
- расчетно-проектная при создании и поддержке сетевых услуг и сервисов;
- экспериментально-исследовательская;
- организационно-управленческая – в сфере информационного менеджмента ИКС.

Знания выпускников востребованы:

- в технических и программных системах;
- в системах и устройствах звукового вещания, электроакустики, речевой, и мультимедийной информатики;
- в средствах и методах защиты информации;
- в методах проектирования и моделирования сложных систем;
- в вопросах передачи и распределения информации в телекоммуникационных системах и сетях;
- в методах управления телекоммуникационными сетями и системами;
- в вопросах создания программного обеспечения ИКС.

Выпускники кафедры Программных систем обладают компетенциями:

- проектировщика и разработчика структур ИКС;
- специалиста по моделированию процессов сложных систем;
- разработчика алгоритмов решения задач ИКС;
- специалиста по безопасности жизнедеятельности ИКС;
- разработчика сетевых услуг и сервисов в ИКС;
- администратора сетей: UNIX и Windows;
- разработчика клиентских и клиент-серверных приложений;
- разработчика Web – приложений;
- специалиста по информационному менеджменту;
- менеджера проектов планирования развития ИКС.

Трудоустройство выпускников:

1. ОАО «Петербургская телефонная сеть»;
2. АО «ЛЕНГИПРОТРАНС»;
3. Акционерный коммерческий Сберегательный банк Российской Федерации;
4. ОАО «РИВЦ-Пулково»;
5. СПБ ГУП «Петербургский метрополитен»;
6. ООО «СоюзБалтКомплект»;
7. ООО «ОТИС Лифт»;
8. ОАО «Новые Информационные Технологии в Авиации»;
9. ООО «Т-Системс СиАйЭс» и др.

Кафедра сегодня имеет в своем составе высококвалифицированный преподавательский состав, в том числе:

- 5 кандидатов технических наук, имеющих ученые звания профессора и доцента;
- 4 старших преподавателя;
- 6 штатных совместителей, в том числе кандидатов наук, профессиональных IT - специалистов;
- 15 Сертифицированных тренеров, имеющих Западные Сертификаты фирм: Microsoft, Oracle, Cisco, Novell.

Современная техническая база; лицензионное программное обеспечение; специализированные лаборатории, оснащенные необходимым оборудованием и ПО; качественная методическая поддержка образовательных программ; широкие Партнерские связи существенно влияют на конкурентные преимущества подготовки специалистов.

Авторитет специализаций кафедры в области компьютерных технологий подтверждается Сертификатами на право проведения обучения по методикам ведущих Западных фирм - поставщиков аппаратного и программного обеспечения.

Заслуженной популярностью пользуются специализации кафедры ПС по подготовке и переподготовке профессиональных компьютерных специалистов с выдачей **Государственного Диплома** о профессиональной переподготовке по направлениям: **"Информационные технологии (инженер-программист)"** и **"Системный инженер"**, а также Диплома о дополнительном (к высшему) образовании с присвоением квалификации: **"Разработчик профессионально-ориентированных компьютерных технологий "**. В рамках этих специализаций высокопрофессиональные преподаватели готовят компетентных компьютерных специалистов по современным в России и за рубежом операционным системам, базам данных и языкам программирования ведущих фирм: Microsoft, Cisco, IBM, Intel, Oracle, Novell и др.

Профессионализм, компетентность, опыт, и качество программ подготовки и переподготовки IT- специалистов на кафедре ПС неоднократно были удостоены **высокими наградами «Компьютерная Элита» в номинации лучший учебный центр России.**

Партнеры:

1. **Microsoft** Certified Learning Solutions;
2. **Novell** Authorized Education Center;
3. **Cisco** Networking Academy;
4. **Oracle** Academy;
5. **Sun Java** Academy и др;
6. **Prometric**;
7. **VUE**.

Мы готовим квалифицированных инженеров в области инфокоммуникационных технологий с новыми знаниями, образом мышления и способностями быстрой адаптации к современным условиям труда.

Светлана Валерьевна Одиночкина

Разработка баз данных в Microsoft Access 2010

ПРАКТИКУМ

В авторской редакции
Редакционно-издательский отдел НИУ ИТМО
Зав. РИО

Н.Ф. Гусарова
Лицензия ИД № 00408 от 05.11.99
Подписано к печати
Заказ №
Тираж 100 экз.
Отпечатано на ризографе

Редакционно-издательский отдел
Санкт-Петербургского национального
исследовательского университета
информационных технологий, механики
и оптики
197101, Санкт-Петербург, Кронверкский пр., 49

