**Тема: MS Access.Структура реляционной БД. Основные объекты БД: таблицы, формы, запросы, отчеты.**

Учебные вопросы

1. Понятие реляционной базы данных

#### 2. Первичные и внешние ключи

#### 3. Объекты БД

#### 4. Создание таблицы в режиме Конструктор

#### 5. Задание типов данных и свойств полей таблицы

#### 6. Создание новых полей и ввод данных из списка

7. Запросы, формы, отчеты в СУБД Access

**1. Понятие реляционной базы данных**

Базой данных (БД) называется организованная в соответствии с определенными правилами и поддерживаемая в памяти компьютера совокупность сведений об объектах, процессах, событиях или явлениях, относящихся к некоторой предметной области, теме или задаче. Она организована таким образом, чтобы обеспечить информационные потребности пользователей, а также удобное хранение этой совокупности данных, как в целом, так и любой ее части.

**Реляционная база данных** представляет собой множество взаимосвязанных двумерных таблиц, каждая из которых содержит информацию об объектах определенного вида. Каждая строка таблицы содержит данные об одном объекте (например, автомобиле, компьютере, клиенте), а столбцы таблицы содержат различные характеристики этих объектов - атрибуты (например, номер двигателя, марка процессора, телефоны фирм или клиентов).

Строки таблицы называются записями. Все записи таблицы имеют одинаковую структуру - они состоят из полей (элементов данных), в которых хранятся атрибуты объекта. Каждое поле записи содержит одну характеристику объекта и представляет собой заданный данные одного типа (например, текстовая строка, число, дата). Для идентификации записей используется первичный ключ. Т.е. информация в реляционных базах данных сохраняется в таблицах, связь между которыми осуществляется путем совпадения значений одного или нескольких полей. Каждая строка таблицы в реляционных базах данных уникальна. Уникальность строк обеспечивают ключи (одно или несколько полей таблицы), которые сохраняются в упорядоченном виде. Это обеспечивает прямой допуск к записям в процессе поиска. Каждая таблица должна иметь первичный ключ (ключевое поле одно или более). Поля-ключи уникально идентифицируют каждую строку в таблице. Первичным ключом называется набор полей таблицы, комбинация значений которых однозначно определяет каждую запись в таблице.

Предметную область в реляционных моделях данных представляют в виде соответствующего числа таблиц. Таким образом, базу данных можно рассматривать как набор таблиц. Таблицы подвергаются классической обработке, так называемому обновлению основного файла.

Таблицы находятся между собой в соответствующих отношениях (связях). Представление предметной области в виде системы отношений – самая сложная и ответственная задача, которая относится к области логического проектирования.

Преимущества реляционных моделей баз данных:

- простота представления данных (таблица);

- минимальный избыток данных, что достигается путем нормализации таблиц;

- независимость приложений пользователя отданных (при включении или удалении таблиц);

- возможность изменения состава атрибутных отношений;

**Недостатки реляционных моделей**: нормализация таблиц приводит к значительной фрагментации данных, а при решении задач, как правило, их необходимо объединять.

#### **2. Первичные и внешние ключи**

Напомним, что ключ или возможный ключ – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Каждая сущность обладает хотя бы одним возможным ключом. Один из них принимается за первичный ключ. При выборе первичного ключа следует отдавать предпочтение несоставным ключам или ключам, составленным из минимального числа атрибутов. Нецелесообразно также использовать ключи с длинными текстовыми значениями (предпочтительнее использовать целочисленные атрибуты). Так, для идентификации студента можно использовать либо уникальный номер зачетной книжки, либо набор из фамилии, имени, отчества, номера группы и может быть дополнительных атрибутов, так как не исключено появление в группе двух студентов (а чаще студенток) с одинаковыми фамилиями, именами и отчествами. Плохо также использовать в качестве ключа не номер блюда, а его название, например, " Закуска из плавленых сырков "Дружба" с ветчиной и соленым огурцом" или "Заяц в сметане с картофельными крокетами и салатом из красной капусты".

Не допускается, чтобы первичный ключ стержневой сущности (любой атрибут, участвующий в первичном ключе) принимал неопределенное значение. Иначе возникнет противоречивая ситуация: появится не обладающий индивидуальностью, и, следовательно не существующий экземпляр стержневой сущности. По тем же причинам необходимо обеспечить уникальность первичного ключа.

Теперь о внешних ключах:

* Если сущность С связывает сущности А и В, то она должна включать внешние ключи, соответствующие первичным ключам сущностей А и В.
* Если сущность В обозначает сущность А, то она должна включать внешний ключ, соответствующий первичному ключу сущности А.

Связь между первичными и внешними ключами сущностей иллюстрируется рис. 1.



Рис. 1 Структуры: а - ассоциации; б - обозначения (характеристики)

Для каждого внешнего ключа в проекте проектировщик базы данных должен специфицировать не только поле или комбинацию полей, составляющих этот внешний ключ, и целевую таблицу, которая идентифицируется этим ключом, но также и ответы на указанные выше вопроса (три ограничения, которые относятся к этому внешнему ключу).

**3. *Объекты базы данных***

Под термином база данных будем понимать – совокупность объектов различного типа (и назначения), которые используются для хранения, отображения, поиска, преобразования и вывода данных. Данные структурированы в виде таблиц.

Аccess 2010 работает со следующими типами объектов: *таблицы, запросы, формы, отчёты, макросы и модули*. Все они сгруппированы по категориям и отображаются в области переходов базы данных.

*Данные* – это представленные в цифровом виде сведения об объектах окружающего нас мира. Ниже дана краткая характеристика основным видам объектов (таблицы, формы, запросы, отчёты, макросы), с которыми придётся работать при изучении приёмов построения базы данных.

*Таблицы* содержат данные, отображают в привычном формате строки и столбцы. В терминологии Access строки называют записями, а столбцы – полями. Поэтому все столбцы таблицы имеют наименования полей (первая строка таблицы). Строки содержат данные в тех форматах, которые назначает им разработчик. Поэтому при определении реляционной базы данных (табличной) говорят, что она представляется в виде совокупности однотипных записей. База данных обычно состоит из нескольких таблиц, объединённых посредством, так называемых связей. Благодаря связям информация из одной таблицы становится доступной для другой, что позволяет обеспечить целостность данных.

*Формы* – это вариант представления на экране компьютера одной записи, что даёт возможность пользователю просматривать последовательно сведения из таблицы, осуществлять поиск и быстрый доступ к любой записи, а также осуществлять операции коррекции данных в записях и создавать новые записи (пополнять таблицу). Фактически, форму можно рассматривать, как некий интерфейс пользователя при работе с таблицами.

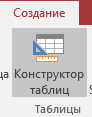
*Запросы* – есть не что иное, как инструмент для управления данными. С помощью запросов можно извлекать данные из одной или нескольких связанных таблиц, осуществлять логические и арифметические операции над данными, группировать данные в новые таблицы по определённым признакам.

*Отчёты* – средство для подготовки выходных файлов. Следует отметить, что слово отчёт прочно ассоциируется со словом «печать», тем не менее, среда генерации отчётов в Access позволяет сформировать конечный продукт в виде виртуальной формы. Любая выходная форма может быть отражена на бумаге или на экране компьютера в удобном представлении данных с использованием средств их форматирования, подведения итогов, фильтрации и графического преобразования в гистограммы и диаграммы.

*Макросы* – это простейшие программы, которые позволяют при обращении к заранее созданным запросам, формам, отчётам, выполнять определённые действия. В Access макрос не создаётся с помощью макрорекордера, а назначается разработчиком. Т. е. для создания макроса необходимо обратиться к режиму конструктора, в котором появляется возможность выбора действия и аргументов макроса.

*Модуль* – содержит одну или несколько процедур, написанных на языке Visual Basic for Application. С помощью модулей можно решить широкий класс задач по поиску и преобразованию информации в базе данных.

**4. Создание таблицы в режиме Конструктор**

 Создание новой пустой таблицы в режиме конструктора позволяет осуществлять операции: добавить поля, задать форматы полей, изменить ключ для полей таблицы, выбрать данные из готовых списков и таблиц, встроить логические и арифметические выражения для контроля вводимых данных. Рассмотрим по порядку возможности этого режима на примере создания заготовки таблицы (пустой). Предположим, что необходимо создать таблицу базы данных со сведениями о личных делах сотрудников организации, в дальнейшем этой таблице присвоим название «Личные сведения». Под личными сведениями сотрудников следует понимать только те данные, которые относятся к конкретной фамилии, например, год рождения, место рождения, пол, фотография, и многое другое. При постановке задачи о ведении базы данных, всегда стремятся выдвинуть требования к данным, которые будут использоваться подразделениями организации. Поэтому, постепенно можно будет добавлять новые поля в таблицу или создавать дополнительные таблицы, связанные с основной таблицей.

Для запуска режима конструктора таблиц необходимо открыть вкладку «Создание», а затем в открывшемся окне щелкнуть **Констурктор**. На рисунке 2 представлено окно конструктора таблицы (с наименованиями полей будущей таблицы), которое будет открыто в основном окне базы данных.

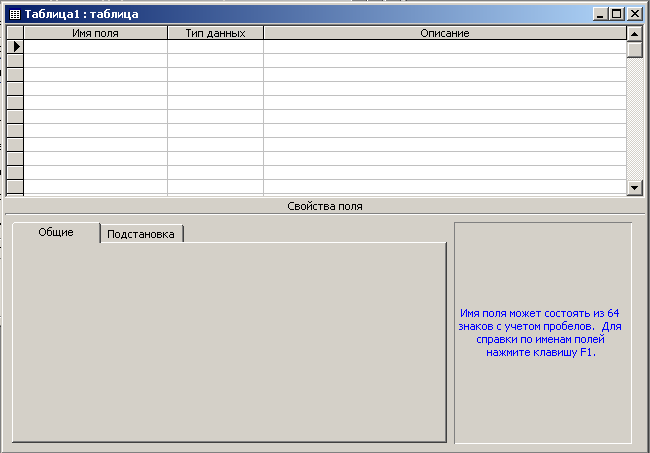


Рис. 2. Пример создания таблицы в режиме конструктора

Окно конструктора состоит из трёх разделов (столбцов): Имя поля, Тип данных и Описание. Имя поля – это заголовок столбца, тип данных – это формат содержащихся данных в столбце, а описание – это комментарий, которые пишет разработчик, чтобы было понятно, какие данные собираются хранить в этом столбце (этот столбец заполнять не обязательно).

Напомним, что начальная цель изучения работы с конструктором таблиц – научиться формировать список полей и устанавливать им свойства, поэтому, сначала создадим заготовку полей таблицы. В качестве начального шага в таблицу внесём следующие поля: Фамилия, Должность, Год поступления, Телефон, Фотография . Имена полей вводятся в строки заготовки будущей таблицы. Можно сначала в строки ввести наименования полей (после ввода наименования поля используйте клавишу ¯ - стрелка вниз), при этом по умолчанию всем полям система присвоит текстовый тип данных, а затем можно установить каждому полю свой тип данных, как это будет показано ниже. Закройте таблицу, нажав на значок справа , при этом система предложит вариант дальнейших действий (Рис. 3.), нажмите на кнопку .

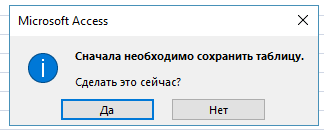


Рис. 3. Предложение системы перед закрытием созданной таблицы

В диалоговое окно «Сохранение» введите имя новой таблицы, нажмите на кнопку «ОК».

Когда осуществляется сохранение вновь созданной таблицы, тогда система всегда предлагает создать дополнительное ключевое поле в таблице, как показано на рисунке 4. Нажмите на кнопку **Да**.

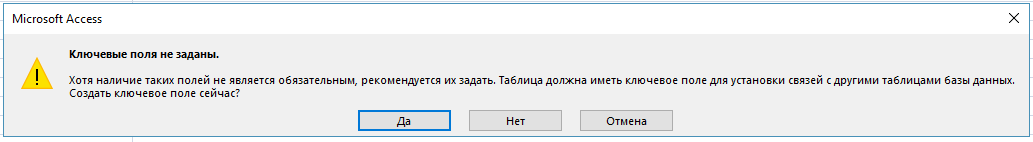


Рис. 4. Окно с предупреждением системы

**5. Задание типов данных и свойств полей таблицы**

Следует обратить внимание на тот факт, что по умолчанию система всем полям таблицы, создаваемым в конструкторе присваивает тип данных «Текстовый», причём свойству «Размер поля» устанавливает число символов такое, которое задано в начальных установках Access (обычно это максимальное число символов равное 255 знакам). Для экономии памяти и ускорения работы, а также для выполнения арифметических действий над данными, требуется решение разработчика базы данных, какой тип данных наиболее подходит к каждому полю. Для установки типа данных для полей таблицы, когда работают в режиме конструктора, следует воспользоваться раскрывающимся списком для выбора необходимого типа данного. Установим типы данных для некоторых полей уже созданной таблицы «Личные сведения».

Откройте таблицу «Личные сведения» в режиме конструктора, для этого в области переходов щёлкните правой клавишей мыши по названию таблицы, а затем в открывшемся меню выберите команду **Конструктор**. Установите курсор в столбце «Тип данных» против строки с нужным именем поля. Раскройте список типов данных, с помощью значка .

Выбор типа данных из списка позволяет полю назначить тип данных (в данном случае для фамилии зададим тип данных – Текстовый). В нижней части конструктора в разделе «Свойства поля» есть две закладки Общие и Подстановка. На закладке «Общие» отображаются свойства данного поля. Обратите внимание, что свойству «Размер поля» система задаёт самостоятельно 255 символов. Наверное, фамилии с таким количеством символов не бывает, измените эту цифру, например, установите размер поля 30 символов. На рисунке 6 показан окончательный вариант, рассматриваемого примера работы с таблицей в режиме конструктора, для которой установлены типы данных для созданных полей.

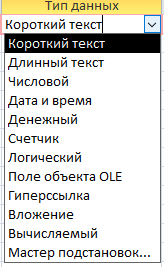


Рис. 6. Пример установки типов данных в режиме конструктора

В конструкторе таблицы (столбец «Описание») даны комментарии, относящиеся к наименованиям полей. Комментарии следует понимать так: для полей «Фамилия», «Год поступления», «Фотография», типы данных установлены на данном этапе, а для полей «Должность», «Телефон» типы данных будут установлены позже. Обратите внимание, что для поля «Год поступления» в качестве свойств выбраны следующие параметры: Размер поля – Длинное целое; Сообщение об ошибке – (текст, который создаёт разработчик); Обязательное поле – Да. Так как телефонные номера содержат не только цифры, то в поле «Телефон» типом данных является текст.

После того, как сформированы наименования полей, установлены для них типы данных можно закрыть режим конструктора (не забывайте соглашаться с системой, отвечайте «Да» - сохранить таблицу). Режим конструктора тем и удобен, что сначала достаточно создать только перечень полей таблицы, а наполнять её можно. Ввод данных в таблицу осуществляется различными способами: вручную в режиме таблица, с помощью мастера подстановок в режиме таблицы, с помощью импорта данных из других источников, с помощью присоединения файлов, с помощью мастера подстановок в режиме конструктора. Ниже будет уделено внимание перечисленным способам заполнения таблиц данными.

**6. Создание новых полей и ввод данных из списка**

Часто приходится выбирать из списка определённые данные, которые жёстко фиксированы. Такие списки создают непосредственно при проектировании полей таблицы. Предположим, необходимо осуществить штатную расстановку сотрудников, в соответствии с имеющимся наименованием подразделений, создадим фиксированный список с наименованиями подразделений в организации. Для этого следует воспользоваться мастером подстановки в режиме таблица, и выполнить несколько шагов.

1. Откройте таблицу в режиме таблицы.

2. Выберите команду «Новое поле», для этого на Вкладке «Режим таблицы» активизируйте ярлык «Поля», на котором щёлкните по Другие поля .

3. В раскрывшемся списке выберите команду Столбец подстановки. Будет открыто диалоговое окно «Создание подстановки». Установите пометку против надписи, нажмите на кнопку ОК. На следующем шаге потребуется создать список в виде таблицы, поэтому, сначала следует указать количество столбцов в таблице подстановки, а затем заполнить наименования строк каждого столбца (ввести данные). В рассматриваемом примере создаётся 1 столбец и три строки, как показано на рисунке 7.

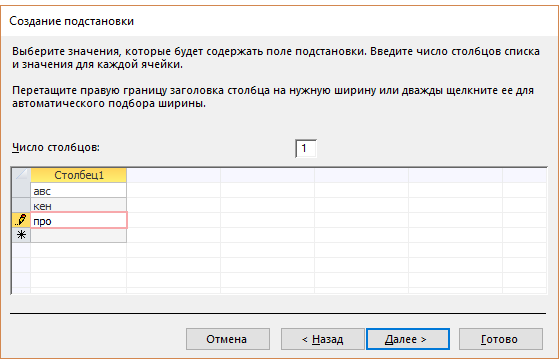


Рис.7. Диалоговое окно для создания таблицы подстановки

Следующий шаг, на котором система предложит задать имя нового поля, желательно выполнить сразу, хотя эту операцию можно выполнить позже. На рисунке 8 показано диалоговое окно мастера подстановки, которое было открыто после нажатия на кнопку **Далее** .

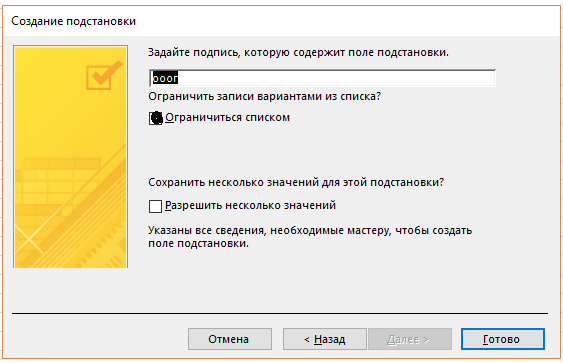


Рис. 8. Диалоговое окно для задания имени добавляемого поля

Обратите внимание, что в диалоговом окне установлена пометка против текста «Ограничиться списком». При нажатии на кнопку **Далее**, система проанализирует правильность записи данных.

Следует внести исправления в наименование поля, например, написать: «Наим\_отдела», а затем нажать на кнопку **Готово** .

4. Заполните данными поле «Наим\_отдела, выбирая из списка (Рис. 9) необходимое наименование, а затем сохраните таблицу».

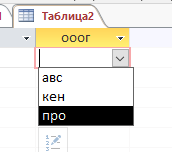


Рис. 9. Раскрывающийся список с наименованиями отделов

**7. Запросы, формы, отчеты в СУБД Fccess**

Запрос — это объект базы данных, являющийся основным инструментом выборки, обновления и обработки данных в таблицах базы данных. Запрос на выборку позволяет сформировать пользовательское представление о данных, Не обязательно отвечающее требованиям нормализации. Результат выполнения запроса на выборку — это новая, чаще всего временная, таблица, которая существует до закрытия запроса. Структура такой таблицы определяется выбранными из одной или нескольких взаимосвязанных таблиц полями. Записи формируются путем объединения записей таблиц, участвующих в запросе. Способ объединения записей различных таблиц указывается при определении их связи. Условия отбора, сформулированные в запросе, позволяют фильтровать записи, составляющие результат объединения таблиц. Простейшие запросы могут быть созданы с помощью мастера. Любой запрос можно создать в режиме конструктора. Конструктор предоставляет удобное для пользователя диалоговое графическое средство формирования, с помощью которого легко может быть построен сложный запрос.

Запрос может получать данные из одной или нескольких таблиц, из существующих запросов или из комбинаций таблиц и запросов, объединяя данные, содержащиеся в них. Таблицы и запросы, объединенные для получения данных, образуют источник записей для запроса. Запрос, построенный на другом запросе, использует его временную таблицу с результатами в качестве источника записей.

Назначение и виды запросов

Запрос позволяет выбрать необходимые данные из одной или нескольких взаимосвязанных таблиц, произвести вычисления и получить результат в виде виртуальной таблицы. Полученная таблица может использоваться в качестве источника данных в формах, отчетах, других запросах. Через запрос можно производить обновление данных, добавление и удаление записей и таблицах.

С помощью запроса можно выполнить следующие виды обработки данных:

-сформировать на основе объединения записей взаимосвязанных таблиц новую виртуальную таблицу;

-включить в результирующую таблицу запроса заданные пользователем поля;

-выбрать записи, удовлетворяющие условиям отбора;

-произвести вычисления в каждой из полученных записей;

-сгруппировать записи, которые имеют одинаковые значения в одном или нескольких полях, в одну запись с одновременным выполнением над другими полями статистических функций;

-добавить в результирующую таблицу запроса строку итогов;

-произвести обновление полей в выбранном подмножестве записей;

-создать новую таблицу базы данных, используя данные из существующих таблиц;

В Access может быть создано несколько видов запроса:

-запрос на выборку — выбирает данные из взаимосвязанных таблиц базы данных и таблиц запросов. Результатом является таблица, которая существует до закрытия запроса. На основе такого запроса могут строиться запросы других видов;

Таблицу с результатами запроса на выборку можно использовать для работы с данными в таблицах, на которых построен запрос. Например, через таблицу запроса можно корректировать данные в исходных таблицах базы данных. Запросы могут служить источниками записей для форм и отчетов.

- запрос на создание таблицы — также выбирает данные из взаимосвязанных таблиц и других запросов, но в отличие от запроса на выборку результат сохраняется в новой постоянной таблице базы данных;

-запросы на обновление, добавление, удаление — являются запросами, в результате выполнения которых изменяются данные в таблицах.

Для создания запроса может быть использован либо режим конструктора, либо мастер. Если пользователь знаком с созданием инструкций SQL (таких как SELECT, UPDATE или DELET, включающих предложения, например WHERE, ORDER BY или GROUP BY), можно создать запрос в режиме SQL.

Вычисляемые поля

В запросе для каждой записи могут производиться вычисления с числовыми, строковыми значениями или значениями дат с использованием данных из одного или нескольких полей. Результат вычисления образует в таблице запроса новое вычисляемое поле. В исходных таблицах базы данных новых полей не создается. При каждом выполнении запроса производится вычисление на основе текущих значений полей.

В выражениях вычисляемых полей помимо имен полей могут использоваться константы и функции. В результате обработки выражения может получаться только одно значение.

Имя вычисляемого поля становится заголовком столбца в таблице с результатами выполнения запроса. Это имя можно изменить. Для вычисляемых полей допускается сортировка, задание условий отбора и расчет итоговых значений, как и для любых других полей.

Групповые операции в запросах Назначение групповых операций

Групповые операции позволяют выделить группы записей с одинаковыми значениями в указанных полях и использовать для этих групп одну из сташ стических функций. В Access предусматривается девять статистических функций:

Sum — сумма значений некоторого поля для группы;

Avg — среднее от всех значений поля в группе;

Max, Min — максимальное, минимальное значение поля в группе;

Count — число значений поля в группе без учета пустых значений;

StDev – среднеквадратичное отклонение от среднего значения поля в группе.

Var -дисперсия значений поля в группе;

First, Last — значение поля из первой или последней записи в группе.

Результат запроса с использованием групповых операций содержит по одной записи для каждой группы.

Многотабличные запросы

Многотабличный запрос позволяет сформировать записи результата путем объединения взаимосвязанных записей из таблиц базы данных и выбора т них нужных полей и записей. Многотабличный запрос часто осуществляет объединение данных, которые на этапе проектирования были разделены им множество объектов в соответствии с требованиями нормализации. Разделение на объекты обеспечивает, прежде всего, отсутствие дублируемости данных и базе, повторяются только значения ключевых полей. В результате выполнения запроса формируется таблица с повторяющимися данными, в которой каждая запись собирает необходимые данные из разных объектов — таблиц.

При проектировании и конструировании запроса важнейшим условием является правильное представление о том, как идет объединение записей таблиц при формировании результата.

Формы

Формы являются основой разработки диалоговых приложений пользователя ля работы с базой данных. Формы, адекватные формам первичных документов, позволяют выполнить загрузку справочных, плановых и оперативно-учетных данных, в любой момент просмотреть и отредактировать содержимое ранее введенных в базу данных документов, оформить новый документ.

Формы обеспечивают удобную работу с данными одной или нескольких взаимосвязанных таблиц, которые выводятся на экран с использованием ее макета, разработанного пользователем. Работая с формой, пользователь может добавлять, удалять и изменять записи таблиц, получать расчетные данные. В процессе работы может осуществляться контроль вводимых данных, могут проверяться ограничения на доступ к данным, выводиться необходимые дополнительные сведения.

Форма состоит из элементов управления, которые отображают поля таблиц, и графические элементы, не связанные с полями таблиц. Графические элементы управления предназначены, прежде всего, для разработки макета формы: надписей, внедряемых объектов (рисунков, диаграмм), вычисляемых полей, кнопок, выполняющих печать, открывающих другие объекты или задачи.

Как форма в целом, так и каждый из ее элементов обладает множеством свойств. Посредством их изменения можно настроить внешний вид, размер, местоположение элементов в форме, определить источник данных формы, режим ввода/вывода, привязать к элементу выражение, макрос или программу. Набор свойств доступен в соответствующем окне, где они разбиты на категории, каждая из которых представлена на своей вкладке. Основными вкладками в окне свойств являются:

Макет — представляет свойства, ориентированные на определение внешнего вида формы или ее элементов;

Данные— представляет свойства для определения источника данных формы или ее элементов, режима использования формы (только разрешение на изменение, добавление, удаление и т. п.);

События — событиями называют определенные действия, возникающие при работе с конкретным объектом или элементом: нажатие кнопки мыши, изменение данных, до обновления, после обновления, открытие или закрытие формы и т. д. Они могут быть инициированы пользователем! или системой. С событием может связываться макрос или процедура обработки события на языке VBA, выполняющая некоторые действия или рассчитывающая значения. Например, в процедуре можно организовать открытие связанной формы, обновление данных таблицы расчетными значениями, печать формы, вывод отчета. Запрограммировав в процедурах вызов различных объектов базы данных, можно автоматизировать выполнение задач приложения.

Для быстрого создания формы предназначены мастера Access. Однако точное формирование макета формы, отвечающего заданным требованиям дополнение процедурами обработки событий, возникающих в форме, обеспечивается средствами конструирования. Конструктор форм можно пользовать как для создания новой формы, так и для редактирования формы, созданной мастером. Кроме того, в Access 2007 включены новые функциональные возможности, позволяющие выполнить доработку форм в режиме макета.

В процессе создания формы выбираются поля таблицы, которые должны быть представлены в форме, осуществляется их размещение в форме, создаются вычисляемые поля, графические элементы — кнопки, выключатели, элементы оформления, поясняющий текст и рисунки. Для настройки различных элементов формы используется типовой набор их свойств.

Формы в Access могут быть представлены в трех режимах.

-Режим формы предназначен для ввода, просмотра и корректировки данных таблиц, на которых основана форма.

-Режим макета обеспечивает просмотр данных почти в таком виде, в каком они отображаются в режиме формы, и в то же время позволяет изменять форму. В этом режиме элементы формы становятся выделяемыми, их можно перетаскивать в другие места, редактировать содержимое надписей полей, изменять формат, размер и т. п. Режим макета позволяет удобно настраивать внешний вид формы и может использоваться для внесения большинства структурных изменений. Если некоторую часть невозможно выполнить в режиме макета, следует переключиться в режим конструктора. В ряде случаев в Access отображается сообщение о том, что для внесения изменений надо переключиться в режим конструктора.

Конструктор предназначен для разработки формы с помощью полного набора инструментов, обеспечивающего более детальную проработку структуры формы, использование всех элементов управления. В этом режиме форму можно разработать с нуля или доработать ее после создания мастером. Просмотр данных при внесении изменений в этом режиме не предусматривается.

Этапы создания интерфейса в Access

Отчеты

Средства Access по разработке отчетов предназначены для конструирования макета отчета, в соответствии с которым осуществляется вывод данных из определенного источника записей отчета в виде выходного печатного документа. Эти средства позволяют создавать отчет любой сложности, обеспечивающий вывод взаимосвязанных данных из многих таблиц, их группировку, вычисления итоговых значений. При этом могут быть выполнены самые высокие требования к оформлению документа.

Перед началом конструирования пользователь должен спроектировать макет отчета. При этом определяется состав и содержание разделов отчета, размещение в нем значений, выводимых из полей таблиц (запросов) базы данных, и вычисляемых реквизитов, определяются поля, по которым нужно группировать данные. Для каждого уровня группировки определяются заголовки и примечания, вычисляемые итоговые значения. Кроме того, оформляются заголовки и подписи реквизитов отчета. Определяется также порядок вывода данных в отчете.

Отчет может создаваться с помощью мастера или в режиме конструктора отчетов. Во многих случаях удобно использовать мастера отчетов. Созданный мастером отчет можно доработать в режиме конструктора.

При необходимости вывода в отчете результатов решения задачи в качестве основы для отчета может быть использован многотабличный запрос, представляющий эти результаты. На запрос могут быть возложены наиболее сложные виды выборки и предварительной обработки данных. Разнообразные возможности конструктора отчетов позволят легко структурировать и оформить полученные в запросе данные.

Отчеты.

Новые средства Access 2007 позволяют создать профессионально оформленные отчеты не только с помощью мастера или конструктора, но и в режиме макета. При этом простыми средствами перетаскивания в отчет нужных полей из таблиц базы данных строится запрос — источник записей отчета, а использование свойств WYSIWYG позволяет сразу видеть, как именно будут выглядеть содержащиеся в нем данные на странице, и усовершенствовать макет.

Чтобы правильно создавать отчеты, необходимо понимать назначение каждого его раздела. Например, от выбора раздела, в который будет помещен вычисляемый элемент управления, зависит способ вычисления результата. Создание и изменение макета отчета осуществляется в расчете на структуру отчета.

Назначение каждого из разделов:

-Заголовок отчета обычно включает эмблему компании, название отчета, дату. Заголовок отображается перед верхним колонтитулом только один раз в начале отчета;

-Верхний колонтитул отображается вверху каждой страницы и используется в случае, когда нужно, чтобы название отчета и другая общая информация повторялись на каждой странице;

-Заголовок группы (Report Headler) используется при группировке записей отчета для вывода названия группы и однократного отображения полей, по которым производится группировка. Отображается перед каждой новой группой записей. Например, если отчет сгруппирован по покупателям, в заголовке группы можно указать название покупателя, а также адрес, телефон и другие реквизиты. Допускается до 10 уровней группировки выводимых записей;

-Область данных (Detail) отображает записи из источника данных, составляющие основное содержание отчета;

-Примечание группы (Footer) используется для отображения итогов и другой сводной информации по группе в конце каждой группы записей. Если поместить в примечание группы вычисляемый элемент управления, использующий статистическую функцию Sum, сумма будет рассчитываться для текущей группы;

-Нижний колонтитул применяется для нумерации страниц и отображения другой информации внизу каждой страницы;

-Примечание отчета служит для отображения итогов и другой сводной информации по всему отчету один раз в конце отчета. Если в примечании отчета поместить вычисляемый элемент управления, использующий статистическую функцию Sum, сумма рассчитывается для всего отчета.

В Access существуют два представления, в которых можно вносить изменения в отчет: режим макета и режим конструктора. Режим макета являем и наиболее удобным для внесения изменений в отчет, поскольку пользователь сразу видит данные отчета. В этом режиме предусмотрено большинство инструментов, необходимых для его настройки. В нем можно изменить шрифт столбцов, поменять их местами, добавить или изменить уровни группировки и итоги. Можно также разместить в макете отчета новые поля, а также задавать свойства отчета и элементов управления.

В режиме конструктора отображаются разделы отчета и предусмотрены дополнительные инструменты и возможности разработки. Переходите в режим Конструктора, если не удается выполнить изменения в режиме макета. В определенных случаях в Access отображается сообщение о том, что для внесения изменений следует переключиться в режим конструктора. Просматривать отчет можно в режимах Представление отчета (Report View), Предварительный просмотр (Print Preview) или Макет (Layout Preview). В режиме Представление отчета можно отфильтровать данные для отображения только заданных строк, найти нужные данные, скопировать текст отчета или его часть в буфер обмена. Режим предварительного просмотра предназначен для просмотра отчета перед печатью. В этом режиме можно увеличивать масштаб для просмотра деталей или уменьшать его для проверки размещения данных на странице, изменить параметры страницы. Режим макета позволяет, просматривая данные отчета, изменять его макет.

**Задание для самостоятельной практической работы (из сборника практических работ)**

**Контрольные вопросы**

1. Что такое первичный ключ? Внешний ключ?
2. Может ли данный внешний ключ принимать неопределенные значения (NULL-значения)?
3. Какими способами можно задать имя поля в таблице?
4. Как осуществляется добавление нового поля в таблицу?
5. Как задавать формат поля в таблице, можно ли его изменить для уже созданного поля?
6. Как осуществляется добавление нового поля в таблицу?
7. Как задавать формат поля в таблице, можно ли его изменить для уже созданного поля?
8. Какие есть возможности у конструктора построения таблиц?
9. Из каких структурных элементов состоит окно конструктора таблиц, каково их основное назначение?
10. Каким образом можно задавать типы полей в таблице, а затем их изменять?
11. Попробуйте охарактеризовать каждый тип данных, доступный в режиме конструктора?
12. Какими средствами можно добавить в таблицы новое поле?
13. Чем удобно средство Мастер подстановок при создании полей в таблице?
14. Сколько шагов потребуется выполнить для создания фиксированного списка в поле таблица с помощью Мастера подстановок?
15. Как отобразится фиксированный список, если в окне «создание подстановки на втором шаге указать число столбцов больше единицы?

**Рекомендуемая литература**

1. Зотова С.И. Практикум по Access - М., Финансы и статистика, 2007
2. Практикум по экономической информатике. Учебное пособие под редакцией проф. Шуремова Е.Л. –М:,2014
3. Быковская Т.И. Экономическая информатика и информационные технологии: лабораторный практикум., Фолиант, Астана, 2011