

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное
образовательное бюджетное учреждение
высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»

М. Б. ВОЛЬФСОН, Е. В. СТРИГИНА

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ

Учебное пособие

СПб ГУТ)))

Санкт-Петербург
2014

УДК 004 65 (075.8)
ББК 32.973.26–018.2я73
В72

Рецензенты:

кандидат технических наук, доцент кафедры ЭиУС СПбГУТ
А. А. Степаненко,
доктор технических наук, профессор кафедры ИТЭ СПбГУТ
А. Д. Сотников

*Утверждено редакционно-издательским советом СПбГУТ
в качестве учебного пособия*

Вольфсон, М. Б.

В72 Информационное обеспечение бизнес-процессов предприятия :
учебное пособие / М. Б. Вольфсон, Е. В. Стригина ; СПбГУТ. – СПб.,
2014. – 56 с.

Рассмотрены вопросы обработки, анализа и отображения формализованных данных в табличном процессоре MS Excel 2007, создания реляционных многотабличных баз данных, их форм, запросов и отчетов на основе СУБД MS Access и возможность публикации данных в сети Интернет.

Приведены примеры формирования пользовательского интерфейса с применением программирования на VBA.

Предназначено для студентов, обучающихся по направлениям (специальности) 080200.62 «Менеджмент» и 080500.62 «Бизнес-информатика».

УДК 004.65(075.8)
ББК 32.973.26–018.2я73

© Вольфсон М. Б., Стригина Е. В., 2014

© Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», 2014

СОДЕРЖАНИЕ

I. ТАБЛИЧНЫЙ ПРОЦЕССОР MS EXCEL.....	4
1. ВЫЧИСЛЕНИЯ В EXCEL.....	5
1.1. Файл Excel.....	5
1.2. Ввод данных.....	6
1.3. Адресация в Excel.....	6
1.4. Использование формул.....	7
1.5. Форматирование.....	9
1.6. Диаграммы в Excel.....	10
1.7. Работа с функциями.....	12
2. ОБРАБОТКА ДАННЫХ В EXCEL.....	17
2.1. Сортировка.....	17
2.2. Фильтрация данных.....	19
2.3. Связывание.....	22
2.4. Консолидация.....	23
2.5. Сводные таблицы.....	25
3. ИНТЕРАКТИВНАЯ РАБОТА С ДАННЫМИ.....	27
3.1. Работа с макросами.....	27
3.2. Встроенные диалоговые окна.....	29
II. СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ MS ACCESS.....	38
4. ОСОБЕННОСТИ СУБД MS ACCESS.....	38
5. СОЗДАНИЕ ТАБЛИЦ.....	39
6. СВЯЗЫВАНИЕ ТАБЛИЦ.....	43
7. РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА.....	44
7.1. Формы.....	44
7.2. Отчеты.....	47
8. СОЗДАНИЕ ЗАПРОСОВ.....	49
9. ИСПОЛЬЗОВАНИЕ СРЕДСТВ АВТОМАТИЗАЦИИ РАБОТЫ БД.....	53
СПИСОК ЛИТЕРАТУРЫ.....	55

I. ТАБЛИЧНЫЙ ПРОЦЕССОР MS EXCEL

При всей многогранности и сложности, информационные технологии базируются на нескольких фундаментальных видах работ, выполняемых с данными на разных этапах. К ним относятся: получение (сбор) данных, обработка данных, хранение данных, представление данных.

В различных задачах на первое место выходят те или иные виды работ, однако в целом они неотделимы друг от друга. Действительно, невозможно представить данные, если они не собраны и сохранены. Нет смысла собирать и хранить данные, если не предполагается их обработка и использование, подразумевающее доступ к собранным данным и сохранение результатов обработки.

Первая часть пособия посвящена главным образом вопросам обработки данных и, в меньшей степени, представлению данных (разделы посвященные форматированию и построению диаграмм).

Обработка данных предполагает их изменение и может быть разделена на несколько видов в зависимости от того, что изменяется в обрабатываемых данных. Можно выделить три основных вида обработки.

1. Обработка, которая связана с изменением значений данных. Например, для числовых типов это соответствует вычислениям, которые выполняются с помощью базовых арифметических операций MS Excel (сложение, умножение и т. д.) или с помощью функций, которые вместе с числовыми константами и переменными величинами объединяются в арифметическое выражение с помощью «формул». Аналогично с помощью логических функций (И, ИЛИ, НЕ) и логических констант (переменных) создаются логические выражения. Подобным образом конструируются и текстовые выражения.

2. Обработка, которая связана с изменением структуры, состава или взаимного расположения данных, но не связана с изменением их значений. Примерами такой обработки служат «сортировка» и «группировка» данных. Такие виды обработки выполняются, как правило, с помощью соответствующих инструментов MS Excel, доступ к которым осуществляется через меню.

3. Комбинированная обработка, которая объединяет в себе оба предыдущих вида. К ней относятся: «консолидация» данных, «промежуточные итоги» или обработка, выполняемая инструментом «сводные таблицы» и др.

Следует отметить, что иногда одинаковые, по сути, виды обработки можно выполнить с помощью специализированного инструмента или с помощью функций. Это относится к вычислению «промежуточных итогов». Общее правило состоит в том, что «инструменты» MS Excel используются, как правило, для агрегированных данных (массивов, диапазонов и т. п.), но обладают ограниченными возможностями в смысле вычислений, в то время

как «формулы» и «функции» ориентированы преимущественно на выполнение сложных расчетных алгоритмов (вычислений), с произвольно расположенными данными. Это существенное отличие нивелируется возможностью копирования и переноса формул.

1. ВЫЧИСЛЕНИЯ В EXCEL

1.1. Файл Excel

Одним из главных свойств программы Excel является возможность работы с большим количеством форматов файлов. Программы Excel 2007, Excel 2010 поддерживают файлы всех форматов, созданных в более ранних версиях Excel.

Excel 2007 поддерживает следующие основные форматы файлов:

XLSX – файлы рабочих книг, не содержащих макросы;

XLSM – файлы рабочих книг, содержащих макросы;

XLTX – файлы шаблонов рабочих книг, не содержащих макросы;

XLTM – файлы шаблонов рабочих книг, содержащих макросы;

XLSB – формат двоичных файлов;

XLSK – формат файлов резервного копирования.

Файл Excel (рабочая книга) содержит по умолчанию три листа, на которых могут быть размещены таблицы или диаграммы.

Табличный лист содержит 1 048 576 (2^{20}) строк и 16 384 (2^{14}) столбцов, таким образом, лист содержит около 17 млрд ячеек.

В Excel 2007 использован новый пользовательский интерфейс, который заменил интерфейс, основанный на системе меню и панелях инструментов. Новый интерфейс получил название «ленточный», поскольку его основными элементами стали ленты и вкладки. Из новшеств интерфейса можно также отметить шесть добавленных шрифтов и панель быстрого доступа. Лента Excel 2007 показана на рис. 1.

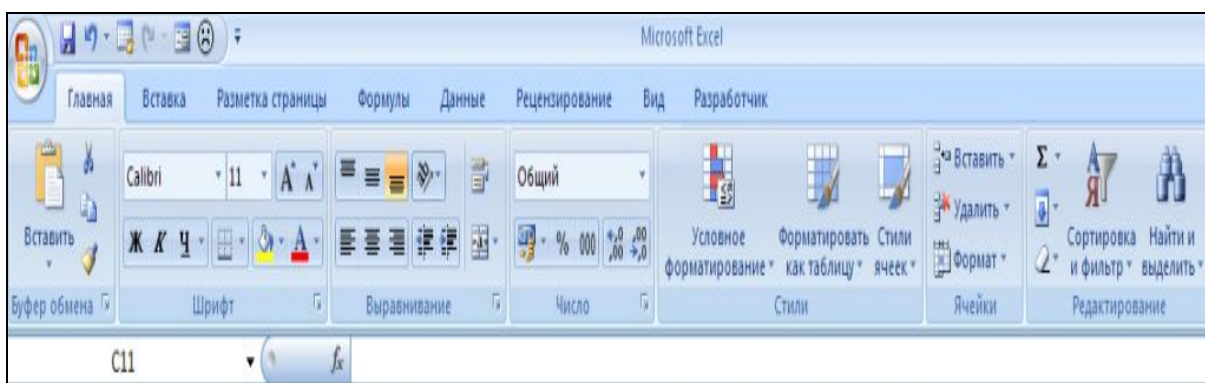


Рис. 1. Ленточный интерфейс

1.2. Ввод данных

В Excel различаются следующие «типы» данных: *Число, Текст, Дата и время.*

Числа по умолчанию выравниваются по правому краю, содержат символы от 0 до 9, +, минус, (), %, E и знаки денежных единиц. В случае недостаточной ширины ячейки в ней отображаются символы #.

Текст может содержать до 32000 символов и по умолчанию выравнивается по левому краю. В неоднозначных случаях для идентификации данных как текста их предваряют апострофом.

При вводе дат и времени Excel преобразует их в «порядковые числа». Самая ранняя дата, с которой оперирует Excel это 1 января 1900 года. Этой дате присваивается порядковый номер – 1. Последующие даты имеют порядковые номера, зависящие от того, сколько дней прошло от 1 января 1900 года до вводимой даты. Такая система представления значительно облегчает использования дат в формулах.

При вводе времени Excel трактует время как дробную часть суток. Так полдень соответствует значению 0,5.

Редактирование введенных данных осуществляется либо в строке формул, при установке курсора на соответствующую ячейку, либо при нажатии клавиши F2, которая позволит отредактировать данные непосредственно в ячейке.

Для упрощения ввода данных используется ряд приемов, из которых стоит обратить внимание на использования списков (встроенные списки позволяют вводить названия дней недели и месяцев при протаскивании маркера автозаполнения).

Заполнение ячеек данными арифметической или геометрической прогрессии с произвольным шагом также облегчает ввод данных.

При вводе даты ячейки могут быть заполнены по дням, месяцам, годам или рабочим дням. Применение прогрессии и заполнение датами обеспечивается использованием контекстного меню и протаскиванием маркера автозаполнения.

1.3. Адресация в Excel

В Excel различают следующие виды адресации:

- абсолютную,
- относительную,
- смешанную,
- с помощью имен.

В таблице на рис. 2, необходимо вычислить стоимость товаров и найти долю стоимости каждого товара в общей сумме стоимостей.

При копировании формулы, записанной в ячейку D2 в диапазон D3:D6, адреса ячеек автоматически изменяются относительно активной ячейки, т. е. в ссылках ячейки D2 применена *относительная адресация*.

Попытка использования относительной адресации в диапазоне E2:E8 привела к ошибке (деление на ноль), поскольку при относительной адресации в ячейке E3 будет сформирована формула D3/D7, т. е. сделана попытка деления на пустую (нулевую) ячейку. Ссылка на ячейку D7 не должна изменяться, другими словами, быть *абсолютной*, а поскольку копирование производится в пределах одного столбца, достаточно сделать ее смешанной, как показано в выноске рис. 2.

	A	B	C	D	E	F	G	H
1	Название	Цена	Количество	Стоимость	Отношение			
2	Товар1	3000	5	=B2*C2	=D2/D7			
3	Товар2	350	10	=B3*C3	#ДЕЛ/0!			
4	Товар3	550	15	=B4*C4				
5	Товар4	2100	3	=B5*C5				
6	Товар5	1000	12	=B6*C6				
7				=сумм(D2:D6)				

Рис. 2. Типы адресации

Изменить тип ссылки можно последовательным нажатием клавиши F4 при установке курсора перед ссылкой, подлежащей изменению.

Ячейка или диапазон могут иметь имя, которое может быть использовано в формуле. Так, если присвоить ячейке D7 имя Итог, то формула в ячейке E2 преобразуется к виду: D2/Итог.

Чтобы присвоить имя ячейке или диапазону, нужно воспользоваться командой **Формулы – Определенные имена – Присвоить имя**.

1.4. Использование формул

Вычисления в Excel производятся по формулам. Формула представляет собой арифметическое или логическое выражение. Это совокупность констант, ссылок, функций, имен диапазонов, соединенных знаками арифметических или логических операций. Результатом вычисления арифметического выражения является число, а логического – значение ИСТИНА или ЛОЖЬ. В случае использования в выражении текстовых операндов результатом также может быть текст. Формуле предшествует знак равенства (=).

Арифметические операции в порядке уменьшения их приоритета следующие:

- [-] (одноместный минус),
- [%] (процент),
- [^] (возведение в степень),

[*], [/] (умножение и деление),
[+], [-] (сложение и вычитание).

Логические операции: [>], [<], [>=], [<=], [=], [<>] (не равно).

Для строк используется операция конкатенации [&].

Кроме этих операций в Excel используются операции диапазон [:], объединение [;] и пересечение [] – знак пробела (рис. 3).

Excel 2007 предлагает новый способ записи ссылок на данные, организованные в виде таблицы (Вставка – Таблицы – Таблица) (рис. 4). Такая таблица имеет собственное имя, например Таблица 1, поэтому можно ссылаться на все ячейки этой таблицы, используя ее имя: =СУММ(Таблица 1). Данная формула просуммирует все ячейки таблицы. В формулах можно использовать также заголовки столбцов таблицы, например =[Количество]*[Цена], если «Количество» и «Цена» – заголовки соответствующих столбцов таблицы.

	A	B	C	D	E	F	G
1	1	2	3	4			
2	11	22	33	44			
3	12	21	31	41			
4							
5			22				
6			225				
7							
8							
9							
10							

Рис. 3. Операции над диапазонами

Еще одно преимущество использования такого вида таблиц заключается в упрощенном вводе формул. Достаточно ввести формулу в одну ячейку столбца, в остальные ячейки она распространяется автоматически.

Формулы могут ссылаться на ячейки других листов – Лист2!С3, других книг – [Отчет]Лист2!С3, и ячейки закрытых книг – 'E:\Мои документы\[Отчет]Лист2!'С3.

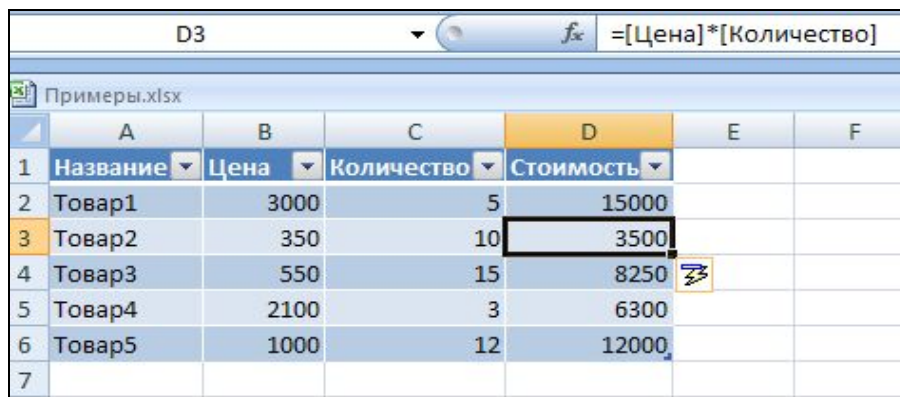
При возникновении синтаксических ошибок, Excel выдает сообщение об ошибке, например #ДЕЛ/0!, и прекращает вычисления.

Для отслеживания ошибок можно воспользоваться просмотром взаимосвязи ячеек, этот механизм включается командой **Формулы – Зависимости формул – Влияющие или Зависимые ячейки**. При этом для активной ячейки будут показаны ячейки, которые влияют на результат вычислений и ячейки, которые зависят от полученного результата.

Помимо синтаксических ошибок, в ходе вычислений могут возникать так называемые «смысловые» ошибки, которые не фиксируются Excel

(например, ввод значения 1000 в поле «вес»). Для их предотвращения используется механизм применения ограничений на величину или тип вводимых данных.

Этот механизм реализован с помощью команды **Данные – Работа с данными – Проверка данных**. Для диапазона, предназначенного для ввода данных, можно установить границы вводимых данных, указать сообщения, которые пользователь увидит при попытке ввода некорректных данных и в начале работы с диапазоном ввода. Особенно удачным приемом, минимизирующим ошибки ввода, является создание списка возможных значений вводимых данных.



	A	B	C	D	E	F
1	Название	Цена	Количество	Стоимость		
2	Товар1	3000	5	15000		
3	Товар2	350	10	3500		
4	Товар3	550	15	8250		
5	Товар4	2100	3	6300		
6	Товар5	1000	12	12000		
7						

Рис. 4. Организация данных в виде таблицы

1.5. Форматирование

Форматирование позволяет улучшить восприятие табличной информации. В Excel 2007 значительно расширены возможности условного форматирования.

Условное форматирование позволяет применять различные форматы в зависимости от значений, содержащихся в ячейках.

Форматированию подвергаются как значения, хранящиеся в ячейках таблицы, так и структурные ее составляющие.

Диалоговое окно «Формат ячейки» позволяет установить вид числовых данных (количество знаков после запятой, цвет данных с отрицательными значениями, разделение троек разрядов), выбрать формат дат и денежных единиц (**Главная – Число**).

Выравнивание (**Главная – Выравнивание**) дает возможность выполнить традиционные действия с содержимым ячейки (выровнять по левому, правому краю или по центру), а также объединить ячейки, перенести данные по словам и применить требуемую ориентацию.

Шрифтовое оформление осуществляется с использованием команды **Главная – Шрифт**.

Изменение свойств структурных составляющих – ширины столбцов и высоты строк осуществляется с помощью команды **Главная – Ячейки**. Эта

же команда позволяет осуществлять автоподбор ширины столбцов и высоты строк.

Команда **Главная – Стили** дает возможность применять стили как целиком к таблице, так и к отдельным ячейкам.

Команда **Главная – Стили – Условное форматирование** позволяет применять правила выбора ячеек в зависимости от их содержимого, применять гистограммы в ячейках, использовать наборы значков и цветовые шкалы. На рис. 5 показано использование набора значков для выделения значений <33 %, от 33 % до 66 % и >66 % выполнения дипломной работы. Снятие форматирования обеспечивается командой **Главная – Редактирование – Очистить форматы**. Копирование формата происходит с помощью пиктограммы «Формат по образцу».

	А	В	С
1	Выполнение дипломной работы		
2	Фамилия	15.апр	15.май
3	Анненков	15%	60%
4	Борисова	50%	80%
5	Валеев	10%	10%
6	Грибов	0%	0%
7	Давыдова	55%	80%
8	Емельянова	45%	90%
9	Павлова	30%	55%
10	Родионов	0%	25%
11	Семенов	45%	85%
12	Тимофеева	25%	75%
13	Циммерман	35%	90%
14	Яковенко	10%	40%
15			

Рис. 5. Условное форматирование

1.6. Диаграммы в Excel

Создание диаграмм – это способ наглядного представления данных, приведенных в таблице в виде чисел. Анализ чисел и их сравнение существенно упрощается при графическом отображении данных.

Диаграмма представляет собой объект, который формируется на основе рядов данных, расположенных в строках или столбцах таблицы. Каждому ряду соответствует свой *маркер* диаграммы. Маркеры показаны в *легенде*, расположенной рядом с диаграммой.

Диаграммы могут располагаться непосредственно в рабочем листе (такая диаграмма называется *внедренной*) или на отдельном листе, который называется *листом диаграммы*.

При выделении объекта внедренной диаграммы или при переходе на лист диаграммы активизируются три новые ленточные вкладки под общим

названием «Работа с диаграммами». Эти вкладки содержат все необходимые команды, чтобы форматировать, редактировать диаграмму и ее элементы.

Excel содержит около сотни вариантов построения диаграмм, выбор которых зависит от конкретной задачи.

Для создания диаграмм нужно воспользоваться командой **Вставка – Диаграмма**.

На рис. 6 представлена гистограмма и ее элементы, а также исходная таблица и ленточные вкладки для работы с диаграммами («Конструктор», «Макет» и «Формат»).

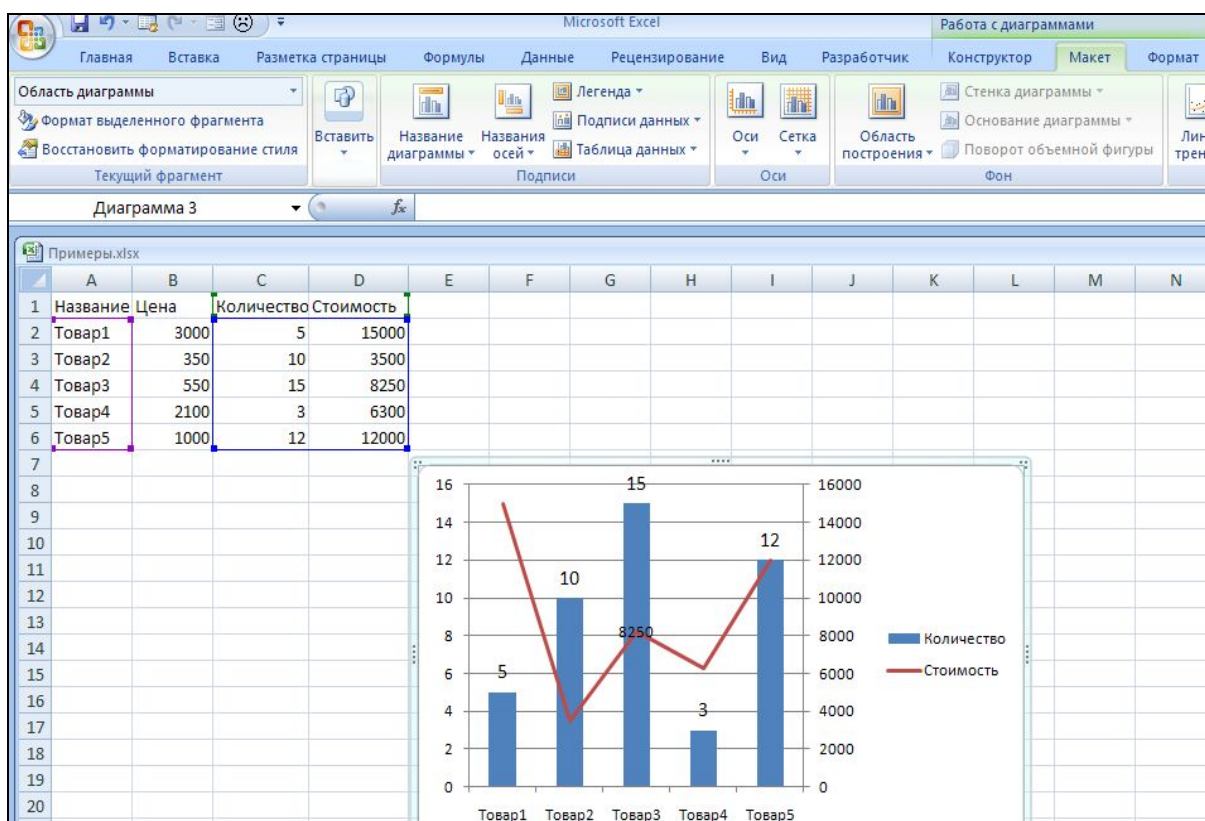


Рис. 6. Внедренная диаграмма

Ряды данных, отображенные в диаграмме, расположены в столбцах таблицы. Для их выделения использованы несмежные области A1:A6 и C1:D6. Диапазон A1:A6 используется для подписей по *оси категорий*.

Поскольку данные, отображаемые в диаграмме, резко разнятся по значениям, применена вспомогательная ось для визуализации данных столбца «Стоимость» и изменен тип диаграммы (график) для показа данных столбца «Количество».

Изменить расположение диаграммы можно, применив команду **Работа с диаграммами – Конструктор – Расположение – Переместить диаграмму**.

1.7. Работа с функциями

Встроенные функции позволяют быстро и просто выполнять необходимые вычисления. Excel имеет более трехсот встроенных функций. При необходимости пользователь может создать собственную (пользовательскую) функцию.

Для удобства использования функции сгруппированы по категориям:

- финансовые;
- дата и время;
- математические;
- статистические;
- ссылки и массивы;
- работа с базой данных;
- текстовые;
- логические;
- проверка свойств и значений;
- определенные пользователем;
- инженерные;
- аналитические.

Ввод функций может осуществляться вручную или с помощью мастера функций.

В первом случае нужно следить за правильностью соблюдения синтаксиса функции, а именно: после имени функции в скобках должны быть записаны необходимые аргументы через точку с запятой. При этом необходимо следить за числом и типом аргументов. При ручном вводе функций можно использовать автозавершение при нажатии клавиши Tab.

Значительно удобнее использовать мастер функций, который активируется при использовании команды **Формулы – Библиотека функций** или кнопки **Вставить функцию**.

Первое диалоговое окно мастера функций предназначено для выбора категории и функции из этой категории (рис. 7), а второе – для выбора аргументов (рис. 8).

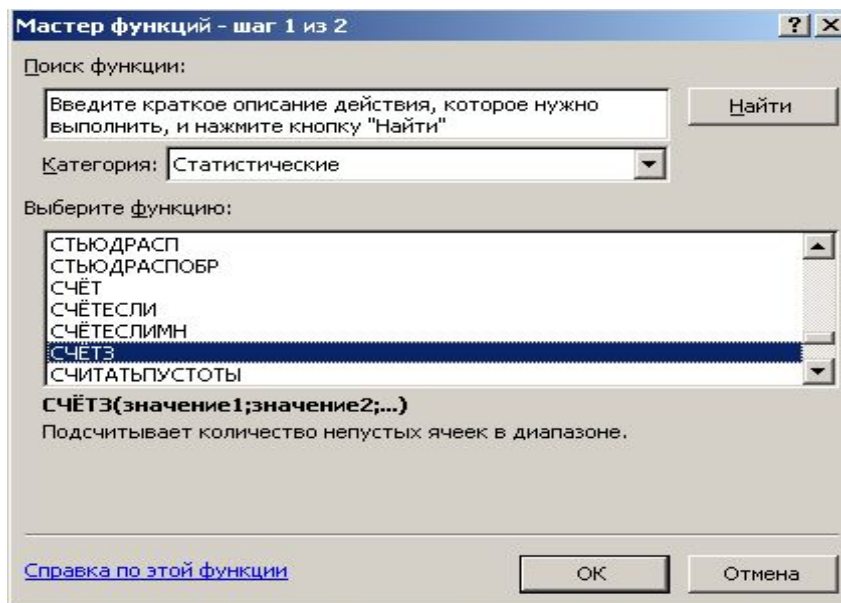


Рис. 7. Выбор категории и функции

Аргументы вводятся в поля, слева от которых находятся названия аргумента, а справа – тип аргумента. Названия обязательных аргументов записываются жирным шрифтом.

Внизу приведена информация, поясняющая смысл аргумента и текущее значение вводимой функции.

На каждом шаге формирования функции можно получить справку по функции с помощью ссылки **Справка по этой функции**.

Формирование функции заканчивается нажатием клавиши **Enter** или **OK**. Для возврата на второй шаг мастера функций после нажатия клавиши **Enter** нужно щелкнуть мышкой на имени функции в строке формул.

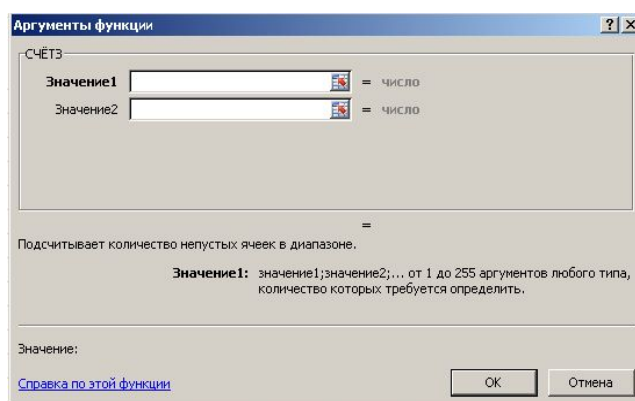


Рис. 8. Выбор аргумента функции

1.7.1. Математические функции

Примеры некоторых математических функций приведены в табл. 1.

Исходные данные для примеров приведены на рис. 9.

Таблица 1

№	Функция	Действие	Пример
1	ПРОИЗВЕД (arg1;arg2;...)	Возвращает произведение аргументов	ПРОИЗВЕД(A1;A2)→330
2	СУММЕСЛИ (диапазон; критерий; диапазон суммирования)	Возвращает сумму значений из заданной области, удовлетворяющих критерию	СУММЕСЛИ(A1:D2;10)→30 СУММЕСЛИ (A1:D2;">16")→53 СУММЕСЛИ (A1:A2;10;B1:B2)→15
3	ОСТАТ (число; делитель)	Возвращает остаток от деления	ОСТАТ(A2;6)→4
4	ОТБР(число;[число рядов])	Отбрасывает дробную часть числа	ОТБР(12,34)→12
5	ОКРУГЛ(число; кол-во цифр)	Округляет число до указанного количества разрядов	ОКРУГЛ(123,456;0)→123 ОКРУГЛ(123,456;1)→123,5 ОКРУГЛ(123,456;-1)→120
6	МОПРЕД(диапазон)	Вычисляет определитель матрицы	МОПРЕД(A1:B2)→395
7	СЛЧИС()	Генерирует случайные числа в диапазоне от 0 до 1	

	A	B	C	D	E
1	33	10	Аня	20	
2	10	15		10	
3	09.09.2012				
4					

Рис. 9. Исходные данные

1.7.2. Логические функции

Основные логические функции приведены в табл. 2.
Исходные данные для примеров приведены на рис. 9.

Таблица 2

№	Функция	Действие	Пример
1	И(арг1;арг2;...)	Возвращает значение ИСТИНА, если все аргументы имеют значение ИСТИНА	И(A1>20;B1>5; C1="Аня")→ ИСТИНА
2	ИЛИ(арг1;арг2;...)	Возвращает значение ИСТИНА, если хотя бы один аргумент имеет значение ИСТИНА	ИЛИ(A1>20;B1<5; C2="Аня")→ ИСТИНА
3	НЕ(арг)	Меняет логическое значение аргумента на противоположное	НЕ(A1>20)→ЛОЖЬ
4	ЕСЛИ(арг1;арг2;арг3)	Возвращает арг2, если арг1 имеет значение ИСТИНА и возвращает арг3, если арг1 имеет значение ЛОЖЬ	ЕСЛИ(A1>20;B1+1;B1-1)→11

1.7.3. Функции категории «Ссылки и массивы»

Примеры функций из этой категории приведены в табл. 3.
Исходные данные для примеров приведены на рис. 9.

Таблица 3

№	Функция	Действие	Пример
1	ВЫБОР (номер_индекса; значение1; значение2; значение3;...)	Выбирает значение из списка значений, порядковый номер которого совпадает с аргументом номер индекса	ВЫБОР(2;A1;B1)→B1→10
2	ПРОСМОТР (искомое значение; вектор просмотра; вектор результата)	Ищет значение в векторе результата, соответствующее значению вектора просмотра	ПРОСМОТР (A2;B1:B2; D1:D2) →D1→20

1.7.4. Функции даты и времени

Примеры функций из этой категории приведены в табл. 4.
Исходные данные для примеров приведены на рис. 9.

Таблица 4

№	Функция	Действие	Пример
1	СЕГОДНЯ()	Возвращает сегодняшнюю дату	СЕГОДНЯ()→08.03.2012
2	ТДАТА()	Возвращает сегодняшнюю дату и время	ТДАТА()→08.03.2012 20:24
3	ДАТА(год; месяц; день)	Возвращает дату	ДАТА(2012;3;25)→25.03.2012
4	ДЕНЬНЕД(дата; тип)	Возвращает номер дня недели	ДЕНЬНЕД(А3;2)→7
5	ГОД(дата)	Возвращает год	ГОД(А3)→2012
6	МЕСЯЦ(дата)	Возвращает месяц	МЕСЯЦ(А3)→9
7	ДЕНЬ(дата)	Возвращает день	ДЕНЬ(А3)→9

1.7.5. Статистические функции

Примеры функций из этой категории приведены в табл. 5.

Исходные данные для примеров приведены на рис. 9.

Таблица 5

№	Функция	Действие	Пример															
1	МИН(арг1;арг2;...)	Возвращает минимальное значение в списке аргументов	МИН(А1:В2)→10															
2	МАКС(арг1;арг2;...)	Возвращает максимальное значение в списке аргументов	МАКС(А1:В2)→33															
3	СРЗНАЧ(арг1;арг2;...)	Возвращает среднее значение списка аргументов	СРЗНАЧ(А1:В2)→17															
4	СЧЕТ(арг1;арг2;...)	Возвращает количество чисел в списке аргументов	СЧЕТ(А1:Д2)→6															
5	СЧЕТЗ(арг1;арг2;...)	Возвращает количество не пустых значений в списке аргументов	СЧЕТЗ(А1:Д2)→7															
6	СЧЕТЕСЛИ(диапазон; условие)	Возвращает количество чисел в диапазоне, удовлетворяющих условию	СЧЕТЕСЛИ(А1:В2;А2)→2															
7	РАНГ(число;ссылка; порядок)	Возвращает ранг числа (в диапазоне)	РАНГ(А1;А1:А2)→1															
8	ПРЕДСКАЗАНИЕ(х; изв_у; изв_х)	Возвращает значение функции в точке х на основе линейной регрессии для известных значений х и у	<p>В ячейку D3 введена функция ПРЕДСКАЗ(D1;A2:C2;A1:C1)→43,3</p> <table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2010</td> <td>2011</td> <td>2012</td> <td>2013</td> </tr> <tr> <td>2</td> <td>45</td> <td>60</td> <td>40</td> <td>43,3</td> </tr> </tbody> </table>		A	B	C	D	1	2010	2011	2012	2013	2	45	60	40	43,3
	A	B	C	D														
1	2010	2011	2012	2013														
2	45	60	40	43,3														

1.7.6. Финансовые функции

Примеры функций вычисления параметров ссуды приведены в табл. 6. В функциях используются следующие аргументы:

- ставка – процентная ставка за один период;
- кпер – общее количество выплат (периодов);
- период – заданный период времени, который должен быть меньше или равен значению кпер;
- плт – взнос, который выплачивается каждый период;
- бс – необязательный аргумент, равный будущей стоимости после последней выплаты, если аргумент опущен, он полагается равным нулю;
- пс – приведенная стоимость;
- тип – необязательный аргумент, указывает, когда должна производиться выплата, если выплата производится в конце периода, аргумент равен нулю и может быть опущен, в противном случае он равен единице.

Таблица 6

№	Функция	Действие	Пример
1	ПЛТ(ставка; кпер; пс; бс; тип)	Возвращает объем выплат по ссуде	Объем ежемесячных выплат по ссуде, 55 000 руб., взятой на 3 года при процентной ставке 6 % годовых, вычисляется по формуле: $ПЛТ(0,06/12;3*12;-55000)$ и составит $\rightarrow 1\,673,21$ р.
2	КПЕР(ставка; плт; пс; бс; тип)	Возвращает количество выплат по ссуде	Ссуда 55000 р. взята под 6 % годовых. Объем ежемесячных выплат по ссуде 1 673,21 руб. Количество периодов выплат по ссуде вычисляется по формуле: $КПЕР(0,06/12;1673,21;-55000)$ и составит $\rightarrow 36$ (месяцев)
3	ПС(ставка; кпер; плт; бс; тип)	Возвращает приведенную сумму ссуды	При процентной ставке 6 % и ежемесячных выплатах 1 673,21 руб., объем ссуды, взятой на 36 месяцев вычисляется по формуле: $ПС(0,06/12;36;1673,21)$ и составит $\rightarrow 55000$ руб.
4	ОСПЛТ(ставка; период; кпер; пс; бс; тип)	Возвращает основную часть выплат по ссуде за определенный период	Следующая формула возвращает основную часть выплаты за первый месяц по ссуде 55 000 руб., взятой под 6 % годовых сроком на три года: $ОСПЛТ(0,06/12;1;36;-55000) \rightarrow 1\,398,21$ руб.
5	ПРПЛТ(ставка; период; кпер; пс; бс; тип)	Возвращает часть выплат по ссуде, которая идет на выплату процентов	Следующая формула возвращает объем выплат по процентам за первый месяц по ссуде 55 000 руб., взятой под 6 % годовых сроком на три года: $ПРПЛТ(0,06/12;1;36;-55000) \rightarrow 275,00$ руб.

2. ОБРАБОТКА ДАННЫХ В EXCEL

2.1. Сортировка

Сортировка данных позволяет упорядочить данные в соответствии с выбранным полем таблицы. В Excel 2007 возможно применить любое количество уровней сортировки.

Для сортировки данных нужно воспользоваться командой **Данные – Сортировка и фильтр – Сортировка** (рис. 10).

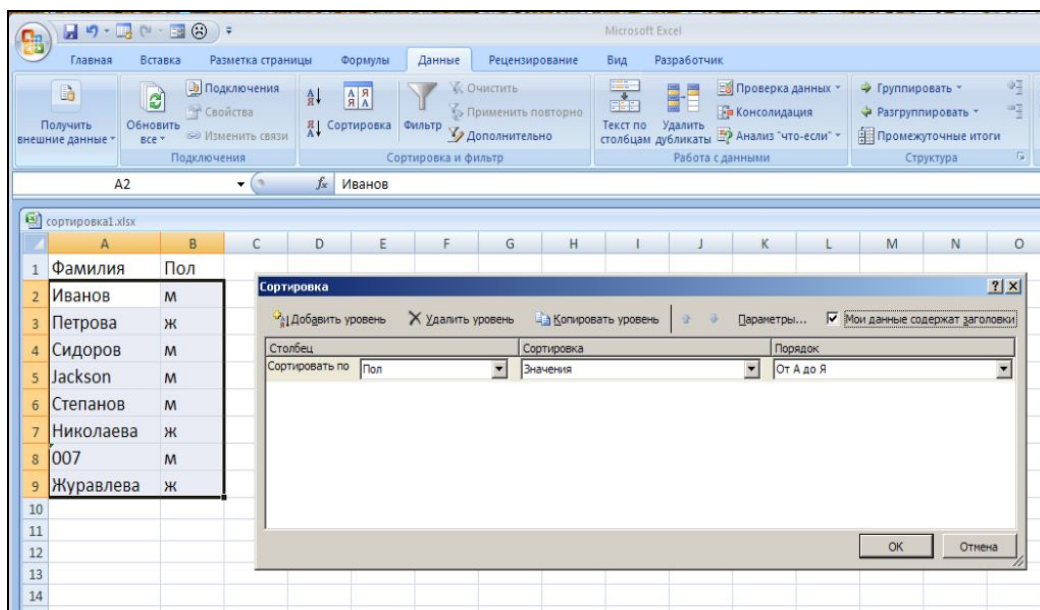


Рис. 10. Использование диалогового окна «Сортировка»

Результаты работы команды «Сортировка» представлены на рис. 11. Исходная таблица приведена в диапазоне A1:B9. В диапазоне D1:E9 приведена таблица, отсортированная по полю **Фамилия по возрастанию** (по алфавиту). В диапазоне G1:H9 приведена таблица, отсортированная по полю **Пол по возрастанию**.

	A	B	C	D	E	F	G	H	I	J	K
1	Фамилия	Пол		Фамилия	Пол		Фамилия	Пол		Фамилия	Пол
2	Иванов	м		007	м		Петрова	ж		Журавлева	ж
3	Петрова	ж		Jackson	м		Николаева	ж		Николаева	ж
4	Сидоров	м		Журавлева	ж		Журавлева	ж		Петрова	ж
5	Jackson	м		Иванов	м		Иванов	м		007	м
6	Степанов	м		Николаева	ж		Сидоров	м		Jackson	м
7	Николаева	ж		Петрова	ж		Jackson	м		Иванов	м
8	007	м		Сидоров	м		Степанов	м		Сидоров	м
9	Журавлева	ж		Степанов	м		007	м		Степанов	м

Рис. 11. Результаты работы команды «Сортировка»

В диапазоне J1:K9 приведена таблица, отсортированная по полю Пол по возрастанию, а затем по полю Фамилия по возрастанию.

Обработка отсортированных данных с помощью команды «Промежуточные итоги» (**Данные – Структура – Промежуточные итоги**) начинается с активизации данной команды (рис. 12).

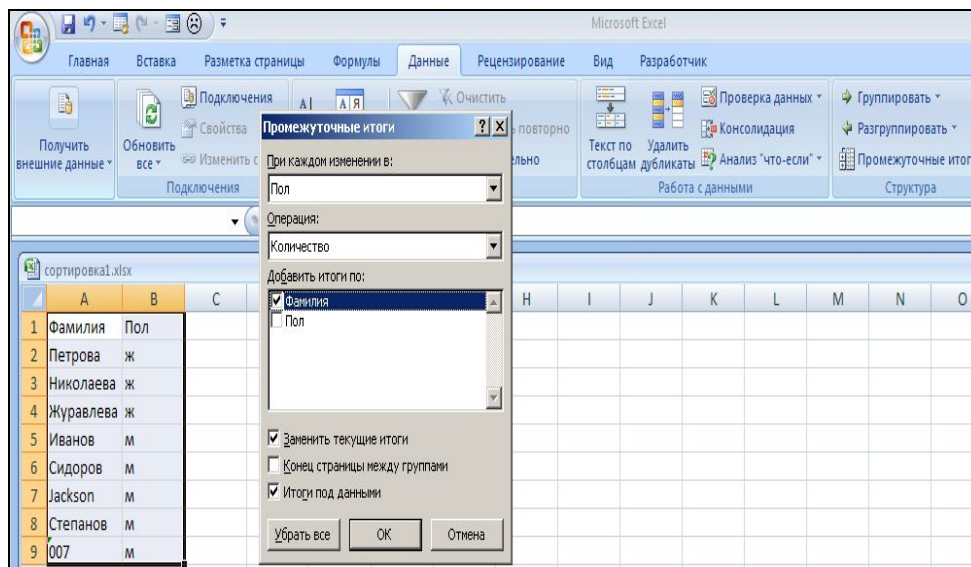


Рис. 12. Использование команды «Промежуточные итоги»

В результате использования команды «Промежуточные итоги» данные структурируются и обрабатываются с помощью выбранной функции (рис. 13).

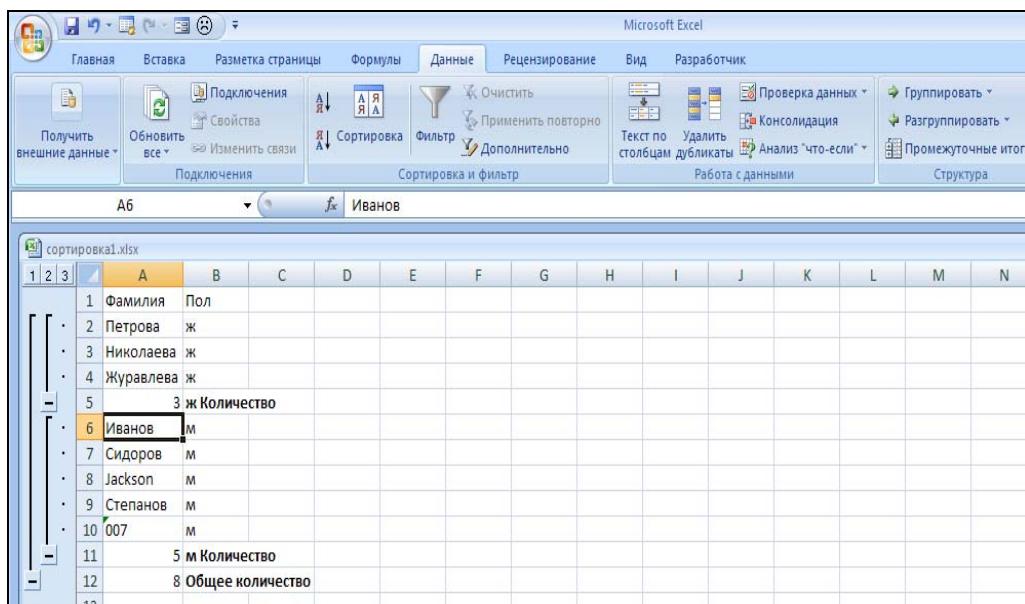
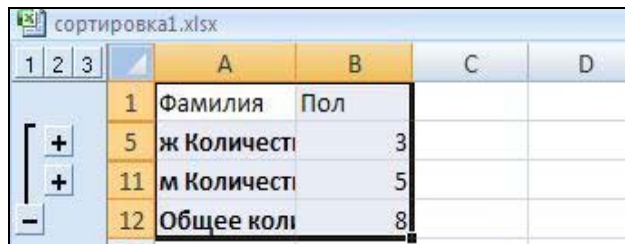


Рис. 13. Результат работы команды «Промежуточные итоги»

Используя управление уровнями структуры, можно получить итоги вычислений по отсортированным группам (по полу), скрыв сами значения

(рис. 14) или результирующие вычисления, оставив только верхний уровень структуры (1).



	A	B	C	D
1	Фамилия	Пол		
5	ж Количест	3		
11	м Количест	5		
12	Общее коли	8		

Рис. 14. Результат сокрытия промежуточных данных

2.2. Фильтрация данных

2.2.1. Автофильтр

Фильтрация позволяет скрыть данные, не удовлетворяющие заданным критериям, и отобразить только те данные, которые критериям удовлетворяют. Фильтры различаются по способам создания критериев. В автофильтре критерии заданы разработчиками, а пользователь должен только выбрать необходимый для решения конкретной задачи (при использовании расширенного фильтра критерии создаются самим пользователем).

Автофильтр активизируется выбором команды **Данные – Сортировка и фильтрация – Фильтр**.

Диалоговые окна со списком критериев представлены на рис. 15.

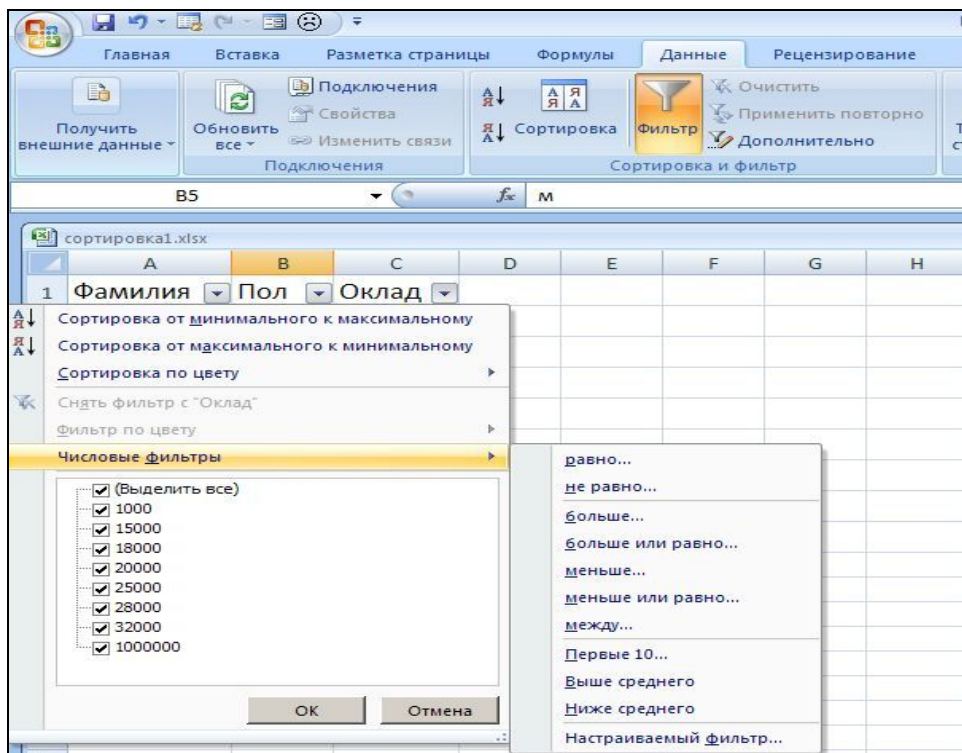


Рис. 15. Активизация фильтра

Настраиваемый фильтр позволяет создавать критерии для одного поля, связанные логическими операциями И и ИЛИ. Пример приведен на рис. 16.

Для обработки отфильтрованных данных используется функция ПРОМЕЖУТОЧНЫЕ.ИТОГИ (код;диапазон). Значения кодов можно посмотреть в справке (1 – СРЗНАЧ, 2 – СЧЕТ, 3 – СЧЕТЗ и т. д.).

2.2.2. Расширенный фильтр

Использование расширенного фильтра обусловлено тем, что он позволяет создавать более сложные критерии, а именно, связанные логическими операциями И и ИЛИ для одного и нескольких полей.

Активизация расширенного фильтра происходит выбором команды **Данные – Сортировка и фильтрация – Дополнительно** (рис. 17).

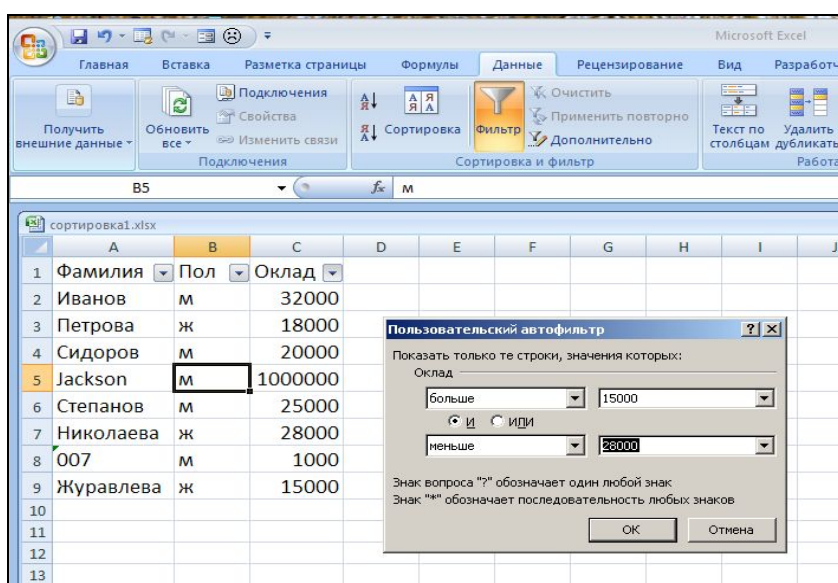


Рис. 16. Пример использования настраиваемого фильтра

Для функционирования расширенного фильтра необходимо создать критерий для фильтрации и поместить его в табличную область, не соприкасающуюся с исходным диапазоном, и диапазоном, в который помещается результат.

Область критериев должна содержать не менее чем две строки. В первой строке указывается название поля, по которому происходит фильтрация, а во второй и следующих указываются значения критерия. В примере, приведенном на рис. 18, область критериев находится в диапазоне E1:E2. В E1 записано название поля – Оклад, а в E2 – значение >20000. Таким образом, из всех данных будут отфильтрованы только те записи, у которых значение поля Оклад >20000.

Область критериев должна быть указана в диалоговом окне «Расширенный фильтр» в поле «Диапазон условий».

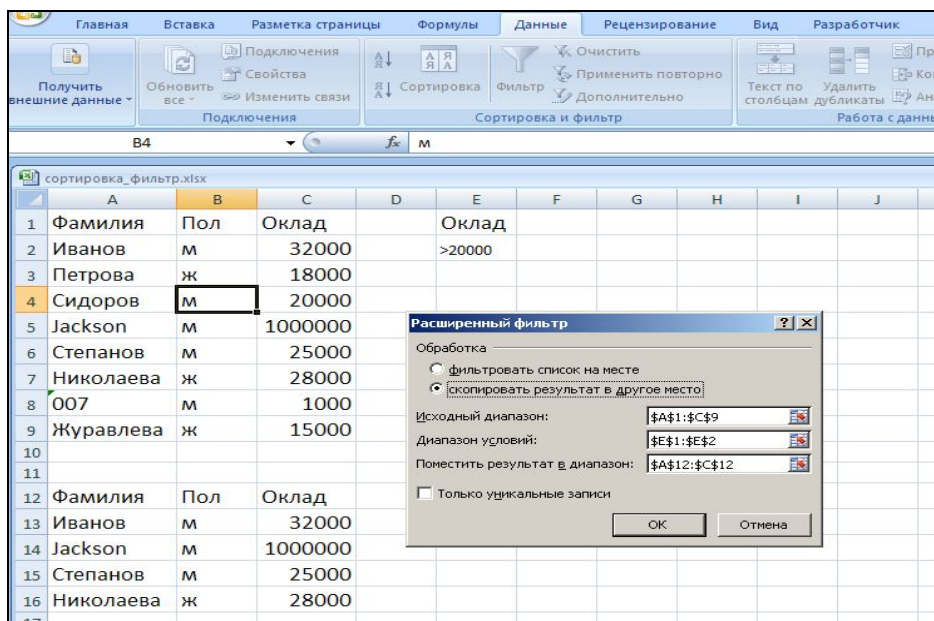


Рис. 17. Активизация расширенного фильтра

Результаты фильтрации могут быть помещены в отдельный диапазон, для чего он должен быть указан в диалоговом окне «Расширенный фильтр» в поле «Поместить результат в диапазон». В этом же окне предварительно должен быть установлен переключатель «Скопировать результат в другое место».

Результаты фильтрации помещены в примере в диапазон A12:C12. Выходной диапазон определяется только одной пустой верхней левой ячейкой в том случае, если он должен содержать все поля исходной таблицы. Для получения результатов фильтрации только в отдельных полях таблицы, их названия нужно поместить в выходной диапазон и указать его в диалоговом окне «Расширенный фильтр» в поле «Поместить результат в диапазон».

Значения критериев, связанных логической операцией И, должны располагаться в одной строке, а связанные логической функцией ИЛИ – в разных. На рис. 18 приведены примеры таких критериев.

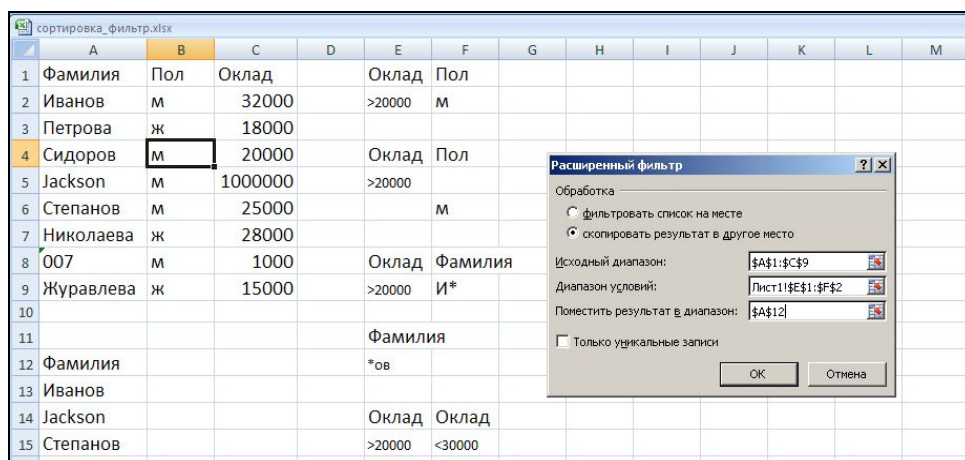


Рис. 18. Примеры использования критериев

При использовании критерия, расположенного в диапазоне E1:F2, выбираются записи, в которых значение поля Оклад >20000, а поля Пол – м, т. е. выбираются мужчины с окладом >20000.

В качестве выходного диапазона указана ячейка A12, в которую помещено название поля Фамилия.

Для последующих примеров в качестве выходного диапазона взят диапазон A12:B12, в который помещены названия полей Фамилия и Оклад.

Критерий, расположенный в диапазоне E4:F6, позволяет найти записи, в которых в поле Оклад значение >20000, или те, в которых поле Пол находится значение «м». Результат фильтрации:

Фамилия	Оклад
Иванов	32000
Сидоров	20000
Jackson	1E+06
Степанов	25000
Николаева	28000
007	1000

Критерий, расположенный в диапазоне E8:F9, позволяет найти записи, в которых в поле Оклад значение >20000 и в поле Фамилия значения начинаются с буквы И. Результат фильтрации:

Фамилия	Оклад
Иванов	32000

Критерий, расположенный в диапазоне E11:E12, позволяет найти записи, в которых в поле Фамилия есть сочетание символов «ов». Результат фильтрации:

Фамилия	Оклад
Иванов	32000
Петрова	18000
Сидоров	20000
Степанов	25000

Критерий, расположенный в диапазоне E14:F15, позволяет найти записи, в которых в поле Оклад значение >20000 и <30000. Результат фильтрации:

Фамилия	Оклад
Степанов	25000
Николаева	28000

2.3. Связывание

Связывание позволяет сберечь память и время на обновление вычислений.

Связывание может быть осуществлено непосредственно с помощью формул, например:

обращение к ячейке

другого листа: =A5*Лист1!C2

обращение к ячейке

другой книги: =A5*[Книга1.xlsx]Лист1!C2

обращение к ячейке

закрытой книги: =A5*'C:\Мои документы\[Книга1.xlsx]Лист1'!C2

Связывание может быть осуществлено с помощью команды **Главная – Буфер обмена – Вставить – Специальная вставка – Вставить связь**.

2.4. Консолидация

Консолидация позволяет объединить данные из исходных листов в итоговом листе. Консолидация может быть динамической или статической. При динамической консолидации сохраняется связь с исходными листами и изменения, происходящие в исходных листах, будут автоматически отображаться в итоговом листе. При статической консолидации связь с исходными диапазонами не сохраняется.

Консолидация данных может производиться по расположению или по категории. При консолидации по расположению объединяются данные, находящиеся в одинаково расположенных ячейках. Консолидация по категории предусматривает использование в качестве основы для объединения листов заголовки строк и столбцов.

Консолидация данных происходит с помощью команды **Данные – Работа с данными – Консолидация**. Исходные листы с данными для консолидации по расположению представлены на рис. 19.

Книга1:1		Книга1:2		Книга1:3			
	А	В		А	В	С	Д
1	Фамилия	Оценка	1	Фамилия	Оценка		
2	Иванов	4	2	Иванов	3		
3	Петрова	5	3	Петрова	5		
4	Сидоров	3	4	Сидоров	4		
5	Степанова	2	5	Степанов	4		
6	Яковлев	3	6	Яковлев	4		

Рис. 19. Исходные листы с данными для консолидации

Консолидируемые данные имеют одинаковое расположение во всех исходных листах. На итоговом листе требуется найти среднюю оценку для каждого студента. На этом листе нужно внести заголовки столбцов «Фамилия» и «Оценка» и установить курсор в ячейку A2, затем активизировать команду «Консолидация».

Диалоговое окно «Консолидация» и его использование представлено на рис. 20. Для осуществления консолидации необходимо выбрать функ-

цию, которая будет использована для консолидированных данных и указать диапазоны данных на исходных листах, подлежащих консолидации.

Динамическая консолидация происходит при установке флажка «Создавать связи с исходными диапазонами». В этом случае образуется структура, уровни которой показаны слева от табличной области. Раскрыв ее, можно увидеть значения консолидируемых данных в исходных листах и проследить за динамикой изменения исходных и результирующих данных.

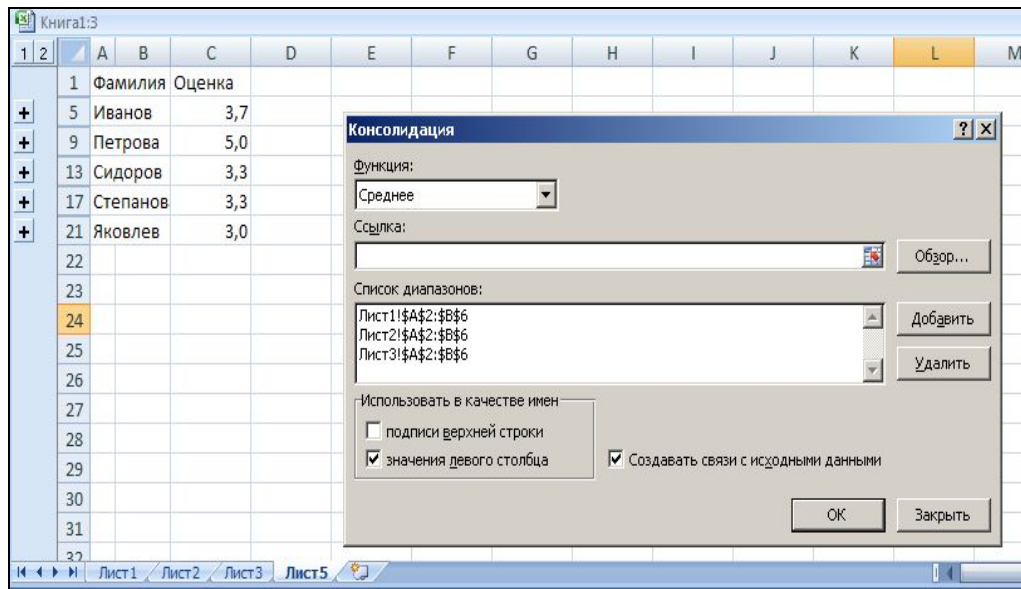


Рис. 20. Диалоговое окно «Консолидация»

Консолидация по категории используется, если исходные данные не расположены в одинаковых ячейках. Пример приведен на рис. 21.

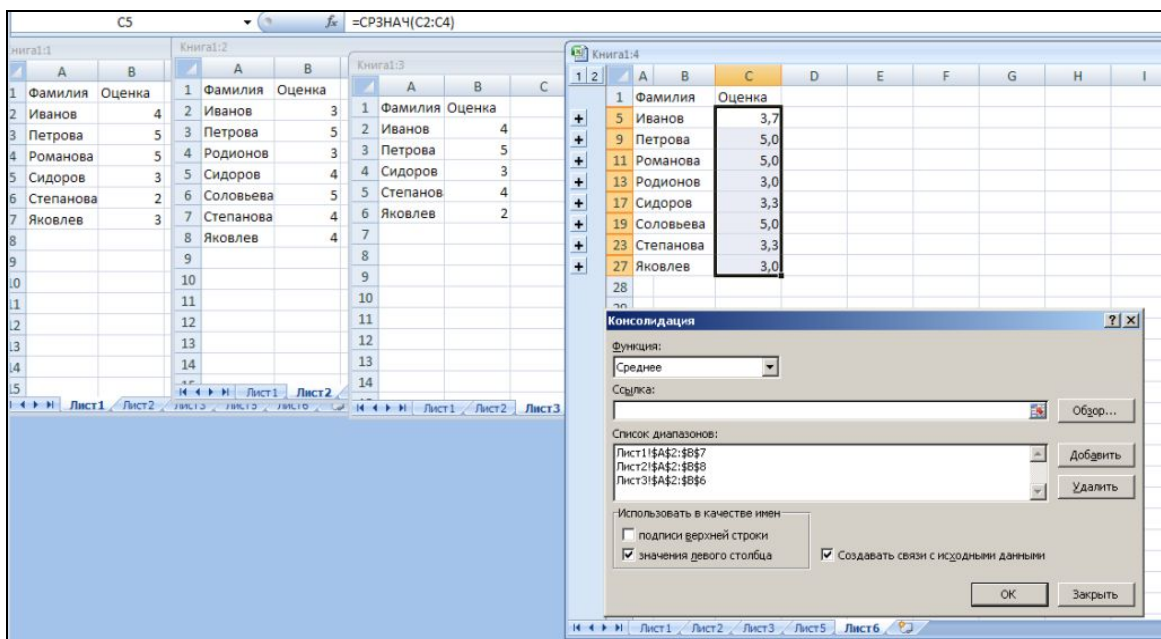


Рис. 21. Консолидация по категории

2.5. Сводные таблицы

Сводные таблицы представляют собой интегральный инструмент обработки, который включает в себя вычисление функций, использование консолидации, преобразование структуры таблицы, фильтрацию и другие операции.

Для создания сводной таблицы нужно использовать команду **Вставка – Таблицы – Сводная таблица**. Сводная таблица позволяет конструировать из исходной таблицы с произвольно расположенными строками и столбцами, применять стандартные функции для обработки данных таблицы или использовать вычисления, необходимые пользователю. Возможность фильтрации, группировки данных, построения диаграмм и другие возможности делают механизм сводных таблиц одним из самых эффективных механизмов обработки данных.

Сводная таблица строится из исходной (рис. 22) посредством заполнения панели «Список полей сводной таблицы», которая определяет макет сводной таблицы.

ФИРМА	МАРКА	ГОД ВЫПУСКА	ЦЕНА	ДАТА ПРОДАЖИ
Альфа	Опель	2000	12000	15.01.2009
Альфа	Опель	1998	10000	20.02.2009
Альфа	Опель	1995	9000	22.02.2009
Альфа	Вольво	2003	15000	03.03.2009
Бета	Вольво	2001	14000	26.03.2009
Бета	Вольво	1999	11000	18.04.2009
Бета	Опель	2005	13000	23.05.2009
Бета	Опель	2001	12000	25.05.2009
Гамма	Вольво	1998	11000	30.05.2009
Гамма	Опель	2006	15000	03.06.2009
Гамма	Опель	2002	13000	05.06.2009
Гамма	Вольво	1997	10000	05.06.2009

Рис. 22. Исходная таблица

На рис. 23 представлена панель «Список полей сводной таблицы» и сводная таблица, построенная на его основе.

При создании сводной таблицы появляются вкладка «Работа со сводными таблицами – Параметры» и «Работа со сводными таблицами – Конструктор».

На вкладке «Работа со сводными таблицами – Параметры» можно изменять активное поле, проводить групповые операции, а также управлять показом заголовков полей и другими элементами сводной таблицы.

Команда **Активное поле – Параметры поля – Операция** позволяет изменять функцию, которая применяется в сводной таблице, а также имя поля.

На вкладке «Работа со сводными таблицами – Конструктор» можно выбрать стиль представления сводной таблицы, а также отображение промежуточных и итоговых вычислений.

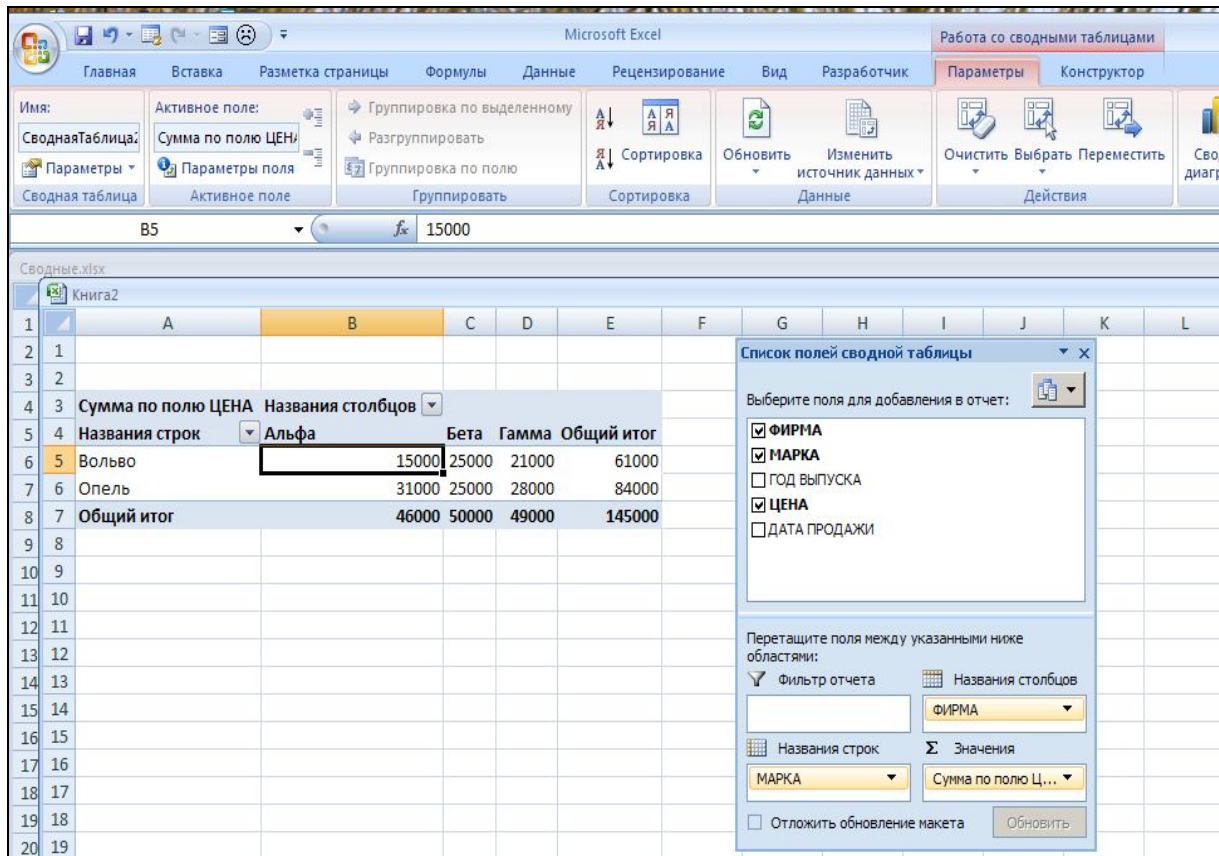


Рис. 23. Макет и сводная таблица

Для создания вычисляемого поля нужно воспользоваться командой **Работа со сводными таблицами – Параметры – Сервис – Формулы – Вычисляемое поле**. В диалоговом окне нужно дать полю имя и ввести формулу для вычислений. Пример приведен на рис. 24. В примере использована формула =30*Цена.

ФИРМА	(Все)	
	Сумма в \$	Сумма в руб
Вольво	\$61 000,00	1 830 000,00р.
Опель	\$84 000,00	2 520 000,00р.
Общий итог	\$145 000,00	4 350 000,00р.

Рис. 24. Пример вычисляемого поля

Для группировки данных из таблицы (рис. 22) нужно установить курсор на ячейку с датами (A5) и воспользоваться командой **Работа со сводными таблицами – Параметры – Группировать**. Результаты группировки по полю представлены на рис. 25, а результаты группировки по выделенному на рис. 26.

	A	B	C	D
1	ФИРМА	(Все) ▾		
2				
3	Сумма			
4		Вольво	Опель	Общий итог
5	▢ Кв-л1			
6	январь		12000	12000
7	февраль		19000	19000
8	март	29000		29000
9	▢ Кв-л2			
10	апрель	11000		11000
11	май	11000	25000	36000
12	июнь	10000	28000	38000
13	Общий итог	61000	84000	145000

Рис. 25. Группировка по полю Дата продаж

	A	B	C	D
1	ФИРМА	(Все) ▾		
2				
3	Сумма			
4		Вольво	Опель	Общий итог
5	▢ зима-лето			
6	15.01.2009		12000	12000
7	20.02.2009		10000	10000
8	22.02.2009		9000	9000
9	03.06.2009		15000	15000
10	05.06.2009	10000	13000	23000
11	▢ весна			
12	03.03.2009	15000		15000
13	26.03.2009	14000		14000
14	18.04.2009	11000		11000
15	23.05.2009		13000	13000
16	25.05.2009		12000	12000
17	30.05.2009	11000		11000
18	Общий итог	61000	84000	145000

Рис. 26. Группировка по выделенному

3. ИНТЕРАКТИВНАЯ РАБОТА С ДАННЫМИ

3.1. Работа с макросами

Excel имеет возможности автоматизации вычислений, создания приложений и их интерфейсов, а также обеспечения взаимодействия с другими компонентами MS Office.

Эти возможности обусловлены применением программирования. Программирование в Excel основано на использовании языка VBA (Visual Basic for Applications), который является языком визуального проектирования и на объектной модели Excel.

Файл Excel, содержащий программный код, должен иметь расширение .xlsm, т. е. сохраняться как «Книга Excel с поддержкой макросов».

Основной программной единицей VBA является *процедура*.

Процедуры разделяются на *процедуры-подпрограммы* и *процедуры-функции*.

Макросы относятся к процедурам-подпрограммам и чаще всего создаются в автоматическом режиме, однако могут быть созданы и вручную.

Макросы хранятся в контейнерном элементе, называемом модулем. При создании макроса в автоматическом режиме модуль создается автоматически.

Для выполнения макроса достаточно вызвать его по имени. Имя макроса не должно содержать пробелов и должно начинаться с буквы. Чтобы макрос автоматически выполнялся при открытии рабочей книги, ему нужно дать имя `Auto_Open`, а при закрытии – `Auto_Close`.

Программный код макроса, как любой процедуры-подпрограммы, содержит код, ограниченный ключевыми словами `Sub` и `End Sub`:

Sub имяМакроса()

Код макроса

End Sub

Программный код функции содержит код функции, ограниченный ключевыми словами `Function` и `End Function`:

Function имяФункции(арг1, арг2,...)

Код функции

End Function

Функции в отличие от макросов не производят каких-либо действий с помощью команд Excel и не изменяют рабочее пространство, поэтому создаются только вручную и их функционирование заключается в преобразовании аргументов. Функции относятся к категории «Пользовательских» и их вызов ничем не отличается от вызова функций других категорий.

Для создания макроса нужно воспользоваться командой **Разработчик – Код – Запись макроса**, дать макросу имя, указать сочетание клавиш для его запуска и объект Excel, выбранный для его хранения, затем выполнить действия, которые должны быть записаны в макросе, и остановить запись с помощью команды **Разработчик – Код – Остановить запись**. Выполнить макрос можно с помощью команды **Разработчик – Код – Макросы – Выполнить** или воспользоваться выбранным сочетанием клавиш.

Увидеть содержимое макроса можно с помощью редактора VBA. Его можно вызвать командой **Разработчик – Код – Visual Basic** или **Разработчик – Код – Макросы – Изменить** или сочетанием клавиш `Alt+F11`. Окно редактора VBA показано на рис. 27.

В верхнем левом углу окна редактора находится окно проектов (Project Explorer), отображающее иерархическую структуру открытых в данный момент проектов VBA. Проект VBA содержит совокупность программных кодов и объектов VBA, обеспечивающих работу данного проекта и сохраненных в одной рабочей книге.

Компоненты, содержащие коды, представлены в виде иерархии папок – Microsoft Excel Objects. К ним относятся листы книги Excel модули (Modules) и формы (Forms), создаваемые пользователем.

Стандартный модуль с кодом макроса находится справа от окна проектов. При необходимости могут быть созданы новые модули с помощью

команды VBA **Insert – Module**. В модулях могут располагаться не только коды макросов, но и другие коды по усмотрению разработчика.

Место расположения кодов не принципиально, поскольку можно обращаться к кодам независимо от того, где они находятся.

Под окном проектов находится окно свойств (Properties), в котором отображаются свойства выделенного объекта.

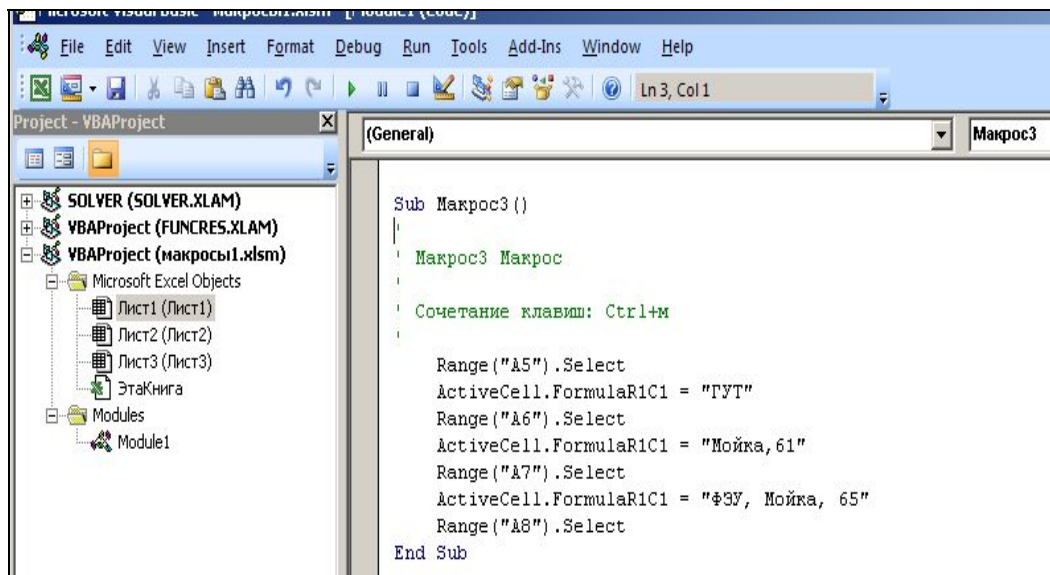


Рис. 27. Окно редактора VBA

3.2. Встроенные диалоговые окна

3.2.1. Окно ввода

Окно ввода формируется с помощью функции InputBox:

InputBox(сообщение [, заголовок] [, по_умолчанию])

«Сообщение» – текст, отображаемый в окне «заголовок» – текст в строке заголовка «по умолчанию» – текст, отображаемый в окне ввода по умолчанию. Если введенные данные должны использоваться для дальнейших вычислений, удобно сохранить результат работы функции в некоторой переменной, например:

a=InputBox(“введите число”)

Для записи числа в конкретную ячейку, например в D5, код следующий:

a=InputBox(“введите число”)

Range(“D5”).Value= a

Код, позволяющий ввести числа в диапазон D5:D10, представлен далее:

For counter=5 to 10

a=InputBox(“введите число”)

Cells(counter,4).Value=a

Next counter

3.2.2. Окно вывода

Окно вывода формируется с помощью функции MsgBox: **MsgBox(сообщение [, кнопки][, заголовок])**

«Сообщение» – текст, отображаемый в окне, «Кнопки» – числовое значение, определяющее кнопки и значки, фокус по умолчанию «Заголовок» – текст в строке заголовка.

Значения констант, используемые для формирования аргумента «кнопки», приведены в табл. 7.

Таблица 7

Группа	Константа	Значение	Отображаемые кнопки и значки
1	vbOKOnly	0	ok
	vbOKCancel	1	Ok и Cancel
	vbAbortRetryIgnore	2	Abort, Retry, Ignore
	vbYesNoCancel	3	Yes, No, Cancel
	vbYesNo	4	Yes, No
	vbRetryCancel	5	Retry и Cancel
2	vbCritical	16	Критическое сообщение
	vbQuestion	32	Вопросительный знак (предупреждение)
	vbExclamaion	48	Восклицательный знак (предупреждение)
	vbInformation	64	Информационное сообщение
3	vbDefaultButton1	0	Фокус на первой кнопке
	vbDefaultButton2	256	Фокус на второй кнопке
	vbDefaultButton3	512	Фокус на третьей кнопке

Функция MsgBox возвращает значение, которое зависит от нажатой кнопки. Для проверки этого значения используются коды возврата, которые присвоены встроенным константам Visual Basic. Коды возврата и имена констант перечислены в табл. 8.

Таблица 8

Константа	Значение	Нажатая кнопка
vbOk	1	Ok
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vbIgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

Для создания диалогового окна, приведенного на рис. 28, нужно использовать следующий код:

```

Sub Выход()
  msg=MsgBox("Завершить работу?", vbYesNo+vbQuestion+vbDefaultButton2, "Выход")
  If msg=vbYes Then
    ActiveWorkbook.Save
    ActiveWindow.Close
  Else
    Exit Sub
  End If
End Sub

```

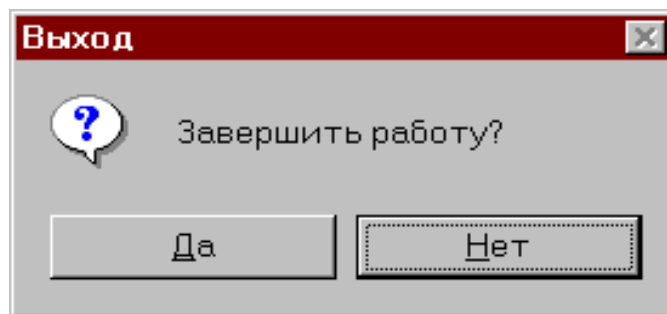


Рис. 28. Диалоговое окно MsgBox

Функция MsgBox может быть использована в виде отдельного оператора, если отсутствуют константы первой группы, например:

```
MsgBox("Для продолжения работы щелкните на кнопке ОК")
```

3.2.3. Работа с экранными формами

Экранные формы представляют собой программируемый интерфейсный элемент, позволяющий создать для приложений Excel визуальный пользовательский интерфейс, обеспечивающий взаимодействие с пользователем по заданному алгоритму. Экранные формы (диалоговые окна) содержат программируемые элементы управления, которые обычно встречаются в диалоговых окнах программы Excel.

Экранная форма имеет три компонента:

- окно со строкой заголовка;
- элементы управления, составляющие ее визуальный и функциональный интерфейс;
- коды VBA, управляющие экранной формой и ее элементами управления.

Форма представлена объектом UserForm. Объект UserForm, как и размещенные в окне формы элементы управления, обладает свойствами, методами и событиями, использование которых определяет внешний вид и функциональные характеристики экранной формы.

3.2.3.1. Использование редактора VBA для создания экранных форм

Создание экранной формы происходит в редакторе VBA. Для создания экранной формы необходимо выполнить команду меню **Insert – UserForm**. По умолчанию вновь созданным формам даются стандартные имена UserForm1, UserForm2 и т. д.

Окно редактора с экранной формой представлено на рис. 29.

В рабочей области находится окно создаваемой формы и панель инструментов (Toolbox), отображаемая в случае активизации формы и содержащая элементы управления, которые могут быть размещены в форме.

В верхней части окна проектов находятся три кнопки. Левая – View Code (Просмотр кода) отображает код активного объекта (элемента управления или формы), средняя – View Object (Просмотр объекта) отображает элементы управления активного объекта (доступна только для экранных форм), правая кнопка – Toggle Folders (Переключение папок) переключает режим отображения элементов в окне проектов.

Для того чтобы открыть окно редактирования кодов формы или элемента управления (рис. 30), нужно воспользоваться кнопкой View Code (Просмотр кода) или дважды щелкнуть мышью на форме или элементе управления. Окно редактирования кодов откроется справа от окна проектов в рабочей области редактора VBA.

В левой нижней части находится окно свойств (Properties), в котором отображаются свойства выделенного объекта. При работе с формой такими объектами может быть сама форма или элементы управления, находящиеся в ней.

Свойствами можно управлять как вручную, устанавливая их значения в окне свойств, так и программным способом на этапе выполнения.

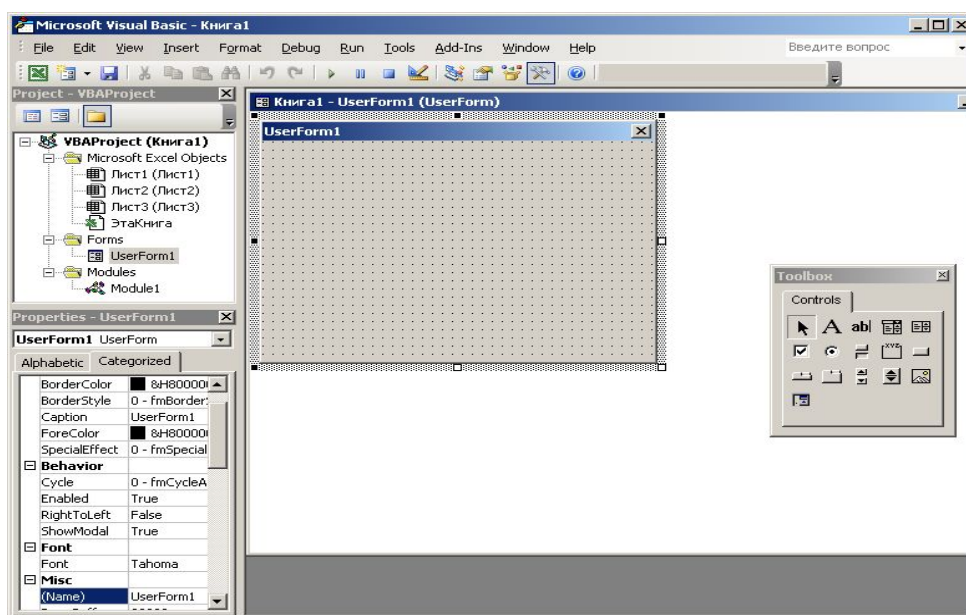


Рис. 29. Окно редактора с экранной формой

3.2.3.2. Свойства и методы экранных форм

В окне свойств выделенной формы можно изменить предложенные свойства или установить незаданные изначально значения. В левой части окна находится список свойств, а правая часть предназначена для указания их значений. Свойства могут быть сгруппированы по категориям (вкладка Categorized) или расположены по алфавиту (вкладка Alphabetic).

Рассмотрим основные свойства экранных форм.

Свойство *Name* определяет имя объекта UserForm, которое будет использовано для обращения к этому объекту.

Свойство *Caption* задает заголовок, отображаемый в строке заголовка окна формы.

Свойства *BackColor*, *ForeColor* и *Font* дают возможность установить цвет фона, символов (переднего плана) и тип шрифта для разрабатываемой формы.

Свойства *Position*:

Свойства *Width* и *Height* определяют ширину и высоту формы, а свойства *Left* и *Top* – положение экранной формы.

Свойство *StartPosition* позволяет установить окно формы по центру экрана (CenterScreen), по центру окна Excel (CenterOwner), произвольно (Manual), по умолчанию в верхней левой части экрана (WindowDefault).

Свойства *Behavior*:

Свойство *ShowModal* определяет модальность окна. При значении True пользователь должен закрыть форму, прежде чем он будет переходить к выполнению другого кода. Если свойству присвоено значение *False*, пользователь может оставить эту форму отображенной на экране и перейти к использованию других частей приложения.

Рисунок, который может быть расположен на форме, определяется свойствами категории *Picture*.

Для получения справки по определенному свойству нужно выделить это свойство в окне Properties и нажать клавишу F1.

Объект UserForm имеет несколько методов:

Hide скрывает форму, если она отображена;

Show отображает форму;

Move (Left, Top, Width, Height) перемещает форму и изменяет ее размеры согласно заданным аргументам.

3.2.3.3. Элементы управления экранных форм

Элементы управления позволяют решать задачи отображения и ввода данных. Каждый элемент обладает собственным набором свойств, методов и событий, в то же время существует стандартный перечень, общий для всех элементов. Для размещения элемента управления на экранной форме достаточно щелкнуть на нем в панели инструментов Toolbox и перетащить в форму.

Рассмотрим наиболее часто используемые элементы управления. Они представлены в табл. 9.

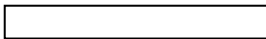


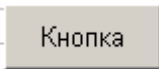
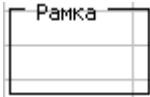

Обращение к элементам 1–4 происходит с помощью конструкции:

Имя формы.Имя элемента.Value или **Имя формы.Имя элемента.Text**

Таким образом, обращение к содержимому текстового поля с именем TextBox1, находящемуся в экранной форме с именем UserForm1, будет: **UserForm1.TextBox1.Value**

При обращении к элементу из самой формы имя формы можно опустить: **TextBox1.Value**

Таблица 9

№	Название	Имя	Отображение в форме
1	Текстовое поле	TextBox	
2	Флажок	CheckBox	<input checked="" type="checkbox"/>
3	Переключатель	OptionButton	<input type="radio"/> <input type="radio"/> <input checked="" type="radio"/>
4	Комбинированный список	ComboBox	
5	Список	ListBox	
6	Командная кнопка	CommandButton	
7	Рамка	Frame	
8	Рисунок	Image	
9	Надпись	Label	А

Текстовое поле

В текстовом поле (TextBox) по умолчанию отображается одна текстовая строка.

Если необходимо отобразить более одной строки, свойству *MultiLine* нужно присвоить значение *True*.

Переход на другую строку происходит по нажатию клавиш Shift+Enter, при этом свойству *WordWrap* присваивается значение *False*.

В текстовое поле могут быть добавлены полосы прокрутки с помощью свойства *ScrollBars*.

Флажки

Свойство Value флажков (CheckBox) возвращает значение True, если они установлены, и False, если нет.

Комбинированный список

Элемент комбинированный список (ComboBox) используется для выбора элемента, отображаемого в списке.

Для добавления элемента в список используется метод `AddItem` «Элемент». Например, для списка с именем `ComboBox1` с помощью приведенного далее кода можно добавить несколько элементов:

```
ComboBox1.AddItem "Книги"  
ComboBox1.AddItem "DVD-диски"  
ComboBox1.AddItem "Аудиокниги"  
ComboBox1.AddItem "Музыка"
```

Каждый элемент, добавленный в список, получает числовой индекс. Первый элемент имеет индекс 0, второй 1 и т. д. Свойство `ListIndex` отображает индекс выбранного элемента. Для начального отображения в списке первого элемента, нужно задать свойству `ListIndex` значение 0:

```
ComboBox1.ListIndex=0
```

Для сокращения записи кода добавления в список, приведенного выше, используется конструкция `With...End With`:

```
With ComboBox1  
    .AddItem "Книги"  
    .AddItem "DVD-диски"  
    .AddItem "Аудиокниги"  
    .AddItem "Музыка"  
    .ListIndex=0  
End With
```

Свойство `ListCount` содержит число элементов, внесенных в список, свойство `List` (номер элемента) позволяет обратиться к любому элементу списка.

Метод `RemoveItem` позволяет удалить элемент из списка, например, при использовании кода `ComboBox1.RemoveItem (2)` удалит из списка третий элемент списка.

Метод `Clear` удаляет все элементы списка.

Список

Элемент управления `ListBox` отображает список. Он может использоваться в двух режимах: для выбора элементов списка и для отображения информации из табличной области Excel.

Добавление и удаление элементов аналогично комбинированному списку.

Для вывода диапазона данных из диапазона `A1:D5`, первая строка которого содержит названия столбцов, можно применить следующий код:

```
With ListBox1  
    .ColumnWidths="50;50;50;40"  
    .ColumnCount=4  
    .RowSource="A1:D5"  
    .ColumnHeads=True  
End With
```

В данном коде используются свойства `ColumnWidths` – ширина столбцов (в пикселах $8,43 \text{ симв}=64\text{px}$), `ColumnCount` – число столбцов, `RowSource` – источник строк, `ColumnHeads` – наличие заголовков столбцов.

Командная кнопка

Элемент управления командная кнопка (`CommandButton`) используется для выполнения некоторых действий после ее нажатия. Иногда требуется, чтобы в окне формы была определена кнопка, выбранная по умолчанию, и кнопка отмены. Чтобы сделать кнопку выбранной по умолчанию, ее свойству `Default` нужно присвоить значение `True`. Командная кнопка становится кнопкой отмены, если ее свойству `Cancel` присваивается значение `True`.

Рамка

Элемент управления рамка (`Frame`) служит для объединения в группу других элементов управления.

Переключатели

Переключатели (`OptionButton`) допускают выбор только одного переключателя из группы. Обычно переключатели объединяются в группы или с помощью элемента `Frame`, или приданием свойству `GroupName` всех переключателей группы одного и того же значения. В экранной форме может быть создано несколько групп переключателей.

3.2.3.4. События экранных форм и элементов управления

Экранная форма и ее элементы управления могут реагировать на события определенных типов. Эти события, как правило, являются результатом действий пользователя – щелчков мышки на элементах управления или нажатий клавиш:

- **Initialize** – происходит при активизации формы;
- **Click, DblClick** – происходит при одинарном или двойном щелчке мыши на объекте;
- **Change** – происходит при изменении данных элементов управления – **CheckBox, ComboBox, OptionButton, TextBox**;
- **KeyDown** – происходит, когда элемент управления находится в фокусе и пользователь нажимает клавишу, доступно для **TextBox** и других элементов управления;
- **MouseMove** – происходит, когда курсор мышки перемещается над объектом, доступно для форм и отдельных элементов управления.

3.2.3.5. Процедуры обработки события

Отклик на событие размещается в коде процедуры обработки события, которая автоматически выполняется в момент наступления события.

Название процедуры обработки события складывается из двух частей, разделенных дефисом: имени объекта, в котором произошло событие,

и названия события. Например, имя процедуры инициализации формы, которая выполняется при наступлении события Initialize, выглядит следующим образом: **UserForm_Initialize**, а имя процедуры обработки события Click командной кнопки с именем cmdBegin будет такой: **cmdBegin_Click**.

Процедуры обработки событий размещаются в окне редактирования кодов, которое активизируется при выборе кнопки View Code или двойном щелчке мышки на элементе управления, в котором происходит событие.

В левой верхней части окна редактирования кодов расположен раскрывающийся список объектов, соответствующих элементам управления, а в правой части находится список событий, характерных для выбранного элемента управления. На рис. 30 показано окно редактирования кодов, в котором отображается процедура обработки события Click для командной кнопки с именем cmdBegin.

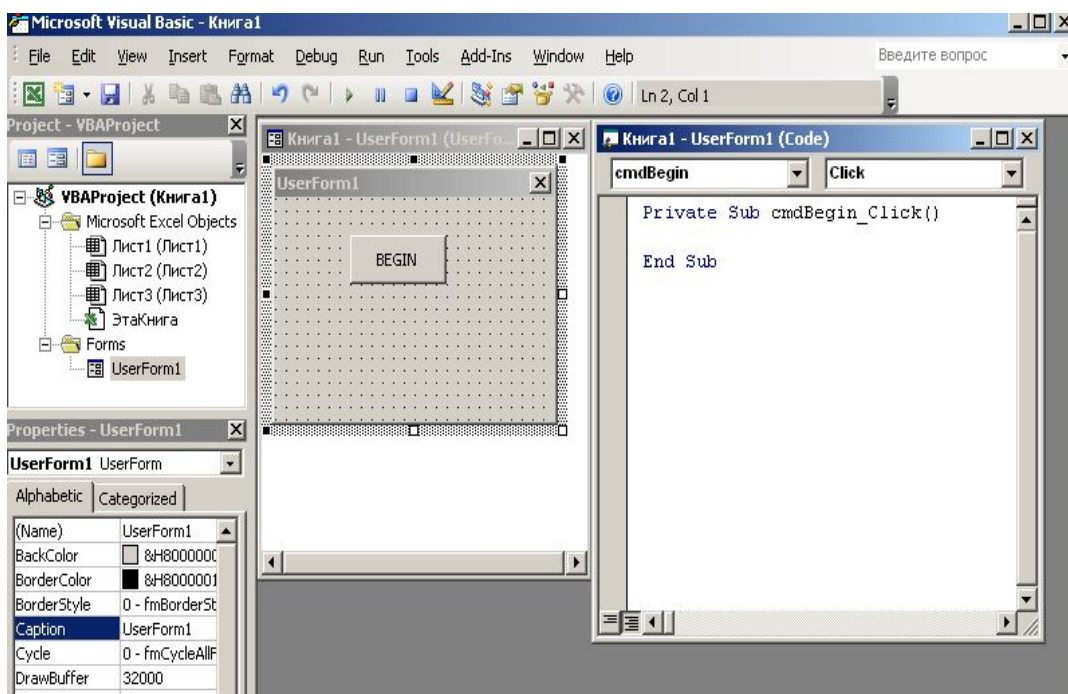


Рис. 30. Окно редактирования кодов

Ключевое слово Private определяет область видимости процедуры и означает локальный уровень видимости, т. е. данная процедура будет видна только в своем модуле, в этом случае – кодах процедур экранной формы.

Чтобы процедура была доступна для всех модулей проекта, можно применить ключевое слово Public, однако этот уровень доступности установлен по умолчанию и не требует дополнительных указаний.

3.2.3.6. Активизация формы

Активизировать форму в отладочном режиме из редактора VBA можно нажатием клавиши F5 или командой **Run – Run Sub/UserForm**.

II. СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ MS ACCESS

4. ОСОБЕННОСТИ СУБД MS ACCESS

Рассмотрим процесс создания БД на основе реляционной СУБД MS Access. Microsoft Access в настоящее время является одной из самых популярных среди настольных СУБД. Однако, как и все файл-серверные СУБД, он не лишен ряда недостатков. Не вызывают сомнения упреки в плохой защищенности данных, медленной работе, проблемах с дублированием данных при резервном копировании, трудностях администрирования, катастрофическом снижении скорости обработки при возрастании объемов данных и т. п.

Однако используемые для решения проблемы средства должны соответствовать сложности самой проблемы. Так, вряд ли имеет смысл тратить на разработку и внедрение информационной системы средства, существенно большие, чем весь годовой оборот предприятия, а для многих предприятий сферы малого и среднего бизнеса дело обстоит именно так.

Итак, где же используется на сегодняшний день MS Access? Прежде всего, в государственных (муниципальных) учреждениях, сфере образования, сфере обслуживания, малом и среднем бизнесе. Специфика возникающих задач заключается в том, что объемы данных не являются катастрофически большими, частота обновлений не бывает слишком высокой, организация территориально обычно расположена в одном небольшом здании, количество пользователей колеблется от одного до 10–15 человек. В подобных условиях использование настольных СУБД для управления информационными системами является вполне оправданным, и они с успехом применяются.

Среди особенностей MS Access следует отметить:

- высокую степень универсальности и продуманности интерфейса, который рассчитан на работу с пользователями самой различной квалификации. В частности реализована система управления объектами БД, позволяющая гибко и оперативно переходить из режима конструирования в режим их непосредственной эксплуатации;

- глубоко развитые возможности интеграции с другими программными продуктами, входящими в состав Microsoft Office (публикация в Word, анализ в Excel), а также с любыми программными продуктами, поддерживающими технологию OLE;

- богатый набор визуальных средств разработки, включающий средства автоматической, автоматизированной (мастера и шаблоны) и ручной разработки (конструкторы).

- хранение всех объектов, в том числе и программного кода, в одном файле;
- возможность публикации данных в сети Интернет;
- благодаря встроенному языку VBA, в самом Access можно писать приложения, работающие с базами данных.

Любая СУБД должна выполнять ряд обязательных функций:

- определение структуры БД;
- заполнение БД;
- поиск и выборку данных;
- представление данных;
- поддержку языков БД;
- защиту физической и логической целостности данных;
- управление доступом к БД и др.

Основные элементы (объекты) MS Access включают в себя:

- таблицы (tables) для хранения данных;
- связи (relationships) между таблицами для обеспечения единства данных;
- формы (forms) для ввода и просмотра данных на экране;
- отчеты (reports) для вывода данных на печать;
- запросы (queries) для поиска необходимых данных по критерию;
- модули (modules) для хранения процедур на языке Visual Basic for Applications;
- макросы (macros) для автоматизации работы базы данных.

5. СОЗДАНИЕ ТАБЛИЦ

После выбора подходящей для решаемой задачи СУБД необходимо перейти непосредственно к реализации БД в соответствии с ТЗ. Первым шагом будет преобразование сущностей и отношений в таблицы, а атрибутов сущностей – в поля таблиц.

Для создания таблиц потребуется перечень сущностей и отношений из ER-модели и перечень неповторяющихся данных (атрибутов сущностей).

При создании таблиц необходимо определить имена этих таблиц и параметры полей.

Для каждого поля необходимо:

- установить тип поля, размер и допустимый диапазон значений;
- определить вычисляемость значений атрибута с использованием другой информации (если атрибут вычисляемый, его необходимо исключить из таблицы);
- произвести внешнее кодирование, т. е. заменить длинные названия полей на короткие;
- создать механизмы повышения надежности ввода данных.

Все эти мероприятия служат одной важнейшей цели – снизить вероятность ошибок, допускаемых пользователями при вводе данных. Источниками неверных данных в таблице могут быть ошибки действий (ошибки пользователя) и ошибки создания (ошибки разработчика).

Остановимся более подробно на ошибках действий. Пользователь может допускать ошибки при вводе данных в двух случаях: случайно (опечатки) и когда ему что-то непонятно. Таким образом, задачей разработчика является снижение вероятности таких ошибок до минимума.

Для этих целей в MS Access существует целый ряд инструментов, содержащихся в Конструкторе таблиц (рис. 31), которые будут перечислены ниже.

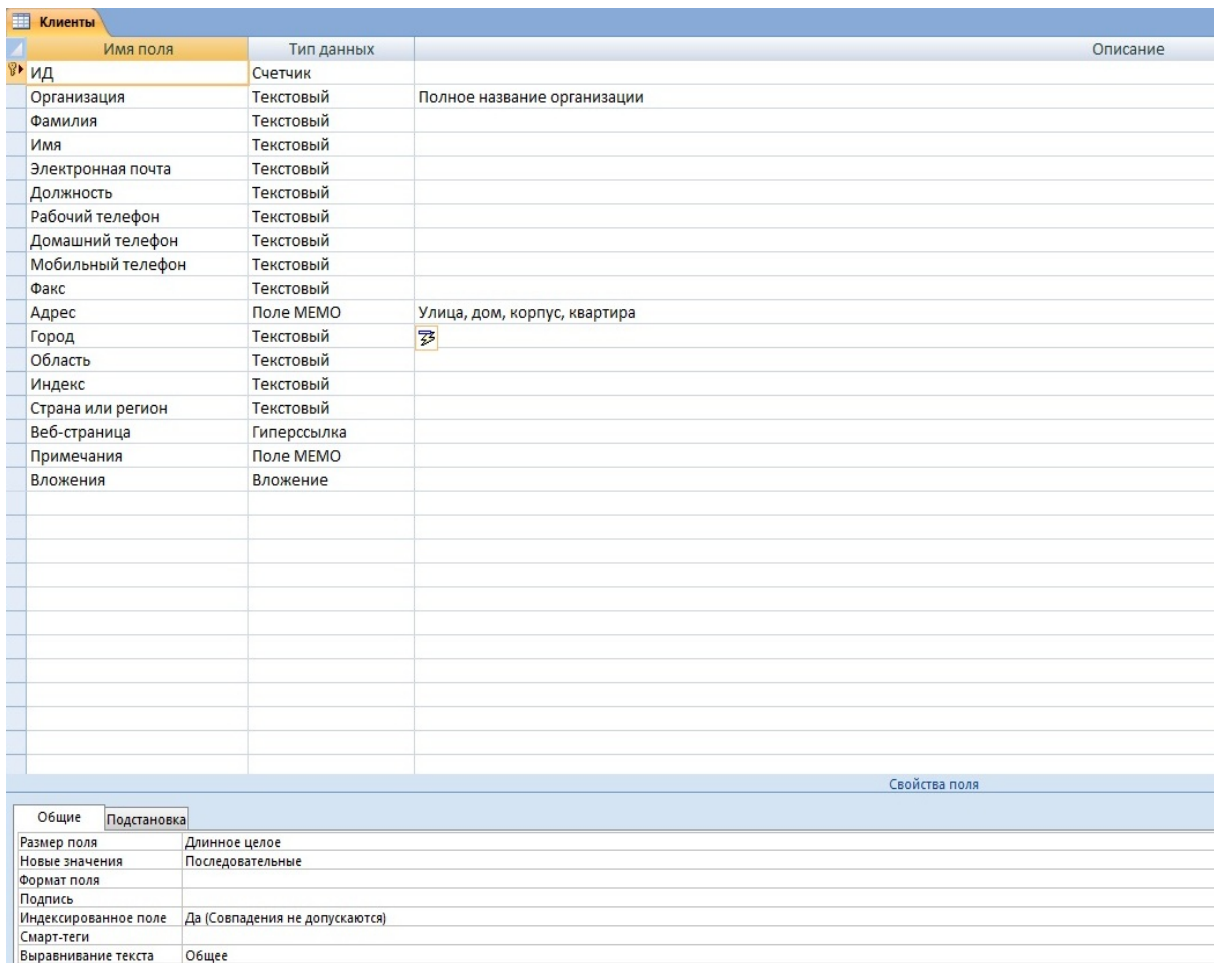


Рис. 31. Конструктор таблиц

Имя поля – определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц), является обязательным параметром поля и не может превышать 64 символа. При этом полное имя поля содержит в себе еще и имя таблицы (например, Студенты.Фамилия). Чем более понятно имя поля, тем меньше ошибок допустит

пользователь при вводе данных (например, имя поля ПочтКод не является интуитивно понятным для пользователя в отличие от имени Почтовый индекс).

Тип поля – определяет тип данных, которые могут содержаться в данном поле, является обязательным свойством и в MS Access может принимать следующие значения (табл. 10).

Благодаря правильно выбранному типу поля, снижается вероятность опечаток (так, например, в числовое поле нельзя случайно ввести букву вместо цифры).

Описание – комментарий к полю, выводимый в строке состояния. Описание поля позволяет пользователю более подробно объяснить назначение поля и соответственно снизить вероятность ошибок.

Таблица 10

Тип поля	Описание	Размер
Текстовый	Для хранения обычного неформатированного текста ограниченного размера. Над числами в текстовом поле нельзя производить вычисления	До 255 символов
Поле Мемо	Хранит длинный текст или числа, например пометки или описание. Не допускает регламентирования вводимых данных	До 65 535 символов
Числовой	Для хранения действительных чисел, над которыми можно производить математические вычисления	1, 2, 4 или 8 байт
Дата/время	Для хранения календарных дат и времени	8 байт
Денежный	Для хранения денежных сумм с денежной единицей (точность до 15 знаков в целой части и 4 – в дробной)	8 байт
Счетчик	Специальный тип для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием или случайных чисел	4 байта
Логический	Тип для хранения логических данных, которые могут принимать только два значения (например, «Да/Нет», «Истина/Ложь», «Включено/Выключено»)	1 бит
Поле объекта OLE	Специальное поле, предназначенное для хранения объектов, созданных другими приложениями (например, документы Word, электронные таблицы Excel, рисунки, звуки). Объекты используют протокол OLE (Object Linking and Embedding) и могут быть связанными или внедренными	До 1 Гбайт
Гиперссылка	Специальное поле для хранения адресов внешних объектов (URL, ссылки на документы Word и т. п.)	До 64 000 символов
Вложение	Специальное поле, предназначенное для хранения объектов, созданных другими приложениями. В отличие от поля объекта OLE в одну запись можно вложить несколько типов данных	

Размер поля – определяет предельную длину данных, которые могут размещаться в данном поле. Для текстовых данных указывает максималь-

ное количество символов, вводимых в поле. Для числовых данных размер поля – это поименованный диапазон значений (например, размер поля «байт» позволяет ввести целые числа от 0 до 255).

Размер поля также позволяет избежать случайных ошибок (например, при установке размера поля ПОЛ, равным 1, пользователь не сможет ввести «муж», «мужской» и т. п.).

Формат поля – определяет способ форматирования данных в поле, позволяет улучшить восприятие данных (например, можно указать на необходимость отображения всех вводимых символов как прописные).

Маска ввода – регламентирует ввод каждого символа. Задает строку символов, каждый из которых кодирует обязательную или необязательную букву или цифру. Кроме этого, текстовые символы, прописанные в маске, будут автоматически отображаться (или храниться) в поле.

Например, маска ввода для поля Телефон может быть задана в виде (000) 000-0000 (0 – обязательная цифра). При этом в поле будут автоматически вводиться скобки и дефис.

Подпись – определяет заголовок столбца таблицы для данного поля. Подпись поля, по сути, является «псевдонимом» поля и может достигать 2048 символов. Имя поля должно быть коротким, так как при разработке БД использование длинных полей крайне неудобно. В то же время пользователю необходимо видеть длинные и понятные имена полей. Для этого используется подпись, заменяющая имя поля в таблицах, формах и отчетах. Если подпись не указана, то в качестве заголовка столбца используется Имя поля.

Значение по умолчанию – то значение, которое вводится в ячейки поля автоматически.

В качестве значения по умолчанию имеет смысл использовать наиболее часто встречающееся значение. Это позволит существенно ускорить процесс ввода и уменьшить количество опечаток. Например, если в поле Страна по умолчанию задать «Россия», то для всех российских студентов в поле Страна это значение будет появляться автоматически.

Условие на значение – ограничение, используемое для проверки правильности ввода данных.

Существует условие на значение поля и записей. Условие на значение поля является логическим ограничением на вводимые данные в поле, поддерживающее непротиворечивость данных (например, Возраст > 18). Условие на значение записей позволяет сравнить данные, хранимые в разных полях (например, Год смерти > Год рождения).

Сообщение об ошибке – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка выполняется автоматически, если задано свойство Условие на значение).

Обязательное поле – свойство, определяющее обязательность заполнения данного поля при наполнении базы.

Пустые строки – свойство, разрешающее ввод пустых строковых данных (значение Null).

Индексированное поле – поле, обладающее свойством, когда все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, чтобы значения в записях проверялись по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных (уникальный индекс).

Важно отметить, что наличие или отсутствие вышеперечисленных свойств зависит от типа поля. Например для полей, в которых хранятся рисунки, звукозаписи, видеоклипы, большинство вышеуказанных свойств не имеют смысла.

6. СВЯЗЫВАНИЕ ТАБЛИЦ

Для реализации связей используются специальные **ключевые поля**. Различают первичные и внешние ключи.

Первичный ключ – одно или несколько полей таблицы, однозначно идентифицирующих каждую запись.

Первичный ключ обладает следующими свойствами:

- значения первичного ключа не могут повторяться;
- первичный ключ не может принимать неопределенное значение

(быть пустым).

В случае если в таблице нет ни одного поля, подходящего под эти требования можно создать **составной первичный ключ**, состоящий из нескольких полей, которые в своей совокупности не повторяются. При этом необходимо соблюдать требование минимальности: ни один из атрибутов не может быть исключен из набора без нарушения уникальности.

В качестве первичных ключей нежелательно использовать поля, содержащие длинный текст (например, название вуза). В этом случае проще добавить в таблицу новое поле (например, код вуза).

Внешний ключ – одно или несколько полей, связанных с первичным ключом из другой таблицы. При этом данные во внешнем ключе могут повторяться.

При создании связей первичные и внешние ключи используются следующим образом:

1) «*один-к-одному*» – в таблицах А и В создаются одинаковые первичные ключи;

2) «*один-ко-многим*» и «*многие-к-одному*» – на стороне «один» создается первичный ключ, а на стороне «многие» – соответствующий ему внешний ключ;

3) «многие-ко-многим» – связь М:М невозможно реализовать напрямую. Для установления связи используют третью таблицу, которую будем называть отношением. Данная таблица будет содержать внешние ключи связываемых таблиц.

Для соблюдения ссылочной целостности рекомендуется установить каскадное обеспечение целостности данных: каскадное обновление связанных полей и каскадное удаление связанных записей. В этом случае изменение значения в первичном ключе приводит к автоматическому обновлению соответствующих значений во всех внешних ключах. Удаление же записи в главной таблице приводит к автоматическому удалению связанных записей в подчиненных таблицах.

7. РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

7.1. Формы

Формы – это интерфейсный элемент, предназначенный для просмотра и ввода данных (рис. 32). Их цель – предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно с этим в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и пр.) для автоматизации ввода. Источником формы может быть таблица (таблицы) или запрос (запросы). При этом происходит наследование свойств полей источника.

Рис. 32. Интерфейсный элемент «Формы»

Формы позволяют пользователям вводить данные в таблицы БД без непосредственного доступа к самим таблицам. Это крайне важно по целому ряду причин.

Во-первых, для обеспечения конфиденциальности данных. Используя набор форм, можно эффективно разграничить доступ пользователей к данным в строгом соответствии с кругом персональных обязанностей. В банках, например, одни сотрудники имеют доступ к таблицам данных о клиентах, другие – к их расчетным счетам, третьи – к таблицам активов банка. Если и есть специальные службы, имеющие доступ ко всем информационным ресурсам банка (в целях контроля и анализа), то они лишены средств для внесения изменений. Все сделано так, чтобы один человек не мог совершить фиктивную операцию, независимо от того, какую должность он занимает.

Во-вторых, в целях безопасности: снижается риск того, что неумелыми действиями они повредят данные в таблицах.

В-третьих, в целях удобства ввода: на экране будут отображаться не все записи, а только одна.

Если структура БД хорошо продумана, то исполнители, работающие с базой, должны навсегда забыть о том, что в базе есть таблицы, а еще лучше, если они об этом вообще ничего не знают. Таблицы – слишком ценные объекты базы, чтобы с ними имел дело кто-либо, кроме разработчика.

Преимущества форм раскрываются особенно наглядно, когда происходит ввод данных с заполненных бланков. В этом случае форму делают графическими средствами так, чтобы она повторяла оформление бланка. Это заметно упрощает работу пользователя, снижает его утомление и предотвращает появление опечаток.

И, наконец, в формах можно производить вычисления, создавая новые вычисляемые поля, что невозможно сделать в таблицах.

Создаваемые формы должны соответствовать функциям пользователей, определяемых на этапе проектирования. Так, в данном примере необходимо осуществлять ввод и просмотр данных об абитуриентах, экзаменах и специальностях.

По выполняемым функциям выделяют следующие виды форм:

- для управления транзакциями (их функции: добавление новых записей в таблицу и изменение существующих);
- для доступа к данным (их функции: представление информации для анализа – диаграммы, формы со статистическими сведениями);
- управляющие, или кнопочные (их функции: управление доступом к объектам базы данных).

По дизайну и структуре формы следующие:

- «в столбец» (отображает все поля одной записи, она удобна для ввода и редактирования данных);

- ленточная (отображает одновременно группу записей, ее удобно использовать для оформления вывода данных, их сравнения и анализа);
- табличная (по внешнему виду ничем не отличается от таблицы, на которой она основана, однако позволяет выводить только необходимые поля, в том числе и вычисляемые).

Форма имеет три основных раздела: *область заголовка*, *область данных* и *область примечания*. Разделы заголовка и примечания имеют чисто оформительское назначение, их содержимое напрямую не связано с таблицей или запросом, на котором основана форма. В этих разделах имеет смысл располагать название формы или итоговые вычисления (например, количество записей, суммарный доход, средний балл и т. д.). Раздел области данных имеет содержательное значение – в нем представлены элементы управления, с помощью которых выполняется отображение данных или их ввод.

Элементы управления бывают *динамические* и *статические*. Динамические элементы управления связаны с полями таблиц, а статические отображают свободные данные и хранятся внутри формы.

К *динамическим* элементам управления можно отнести:

- текстовые поля – служат для отображения данных из таблицы, ввода и редактирования данных, а также создания вычисляемых полей;
- флажки, переключатели, выключатели и их группы – служат для ввода и отображения данных из логических полей (например, отметка о зачете: да/нет);
- раскрывающиеся списки – позволяют осуществлять ввод посредством выбора значений из готового фиксированного списка;
- присоединенные рамки объектов – служат для отображения полей объектов OLE и полей вложений из таблицы (фотографий, рисунков, документов и т. д.)

К *статическим* элементам управления относят:

- надписи – произвольный текст;
- кнопки – служат для назначения им макросов или программ на VBA;
- вкладки – служат для создания многостраничных форм;
- линии, прямоугольники – для графического оформления формы;
- свободные рамки объектов – содержат внедренные в форму объекты (например, графическая эмблема компании). Для хранения рисунков можно использовать и элемент управления «Рисунок».

Для создания вычисляемых полей в форме необходимо создать новое поле и разместить там соответствующее выражение. Для удобства создания формул можно использовать «Построитель выражений».

Для анализа данных в формах применяют диаграммы. Диаграммы служат для наглядного графического представления информации, облегчая для пользователей сравнение и выявление тенденций и закономерностей в

данных. При этом диаграмма может как сопровождать данные в форме, так и занимать всю форму. Диаграммы могут быть глобальными (включающими все данные) или связанными с отдельной записью (отражающими данные только текущей записи и обновляющимися при переходе на другую запись).

Для создания кнопочных форм существует несколько путей, можно использовать:

1) элемент управления «Кнопку» для создания кнопок и назначения им макросов или процедур на языке VBA;

2) «мастер кнопок», позволяющий создать процедуру обработки события и назначить ее Кнопке;

3) «диспетчер кнопочных форм» для создания специальной таблицы (Switchboard Items), которая и хранит всю информацию о кнопочной форме.

Однако обычные формы, созданные на основе конкретной таблицы, не дают возможности одновременно видеть и вводить данные в связанные таблицы. Это крайне неудобно, так как не дает воспользоваться в полной мере всеми преимуществами установленных связей. Если необходимо обращаться к данным из разных таблиц, целесообразно создать *составные формы*.

При этом та форма, которая создана на основе таблицы со стороны «один», становится главной, а со стороны «много», – подчиненной. Встраивание подчиненной формы в главную можно осуществить в «Конструкторе формы» с помощью соответствующего элемента управления, методом Drag-and-Drop или с помощью Мастера форм.

7.2. Отчеты

Отчеты – это интерфейсный элемент, предназначенный для вывода данных на печать (рис. 33).

По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на печатающее устройство (принтер). В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов (верхний и нижний колонтитулы, номера страниц, служебная информация о времени создания отчета и т. п.).

В отличие от форм отчеты:

- предназначены только для печати и не предназначены для вывода в окне;

- позволяют изменять значения исходных данных;

- имеют два режима работы: предварительный просмотр и режим конструктора.

Приемы создания и редактирования отчетов те же, что и для форм. Элементы управления в данном случае выполняют функции элементов

оформления, поскольку печатный отчет – не интерактивный объект, в отличие от электронных форм.

Адресная книга клиентов

14 марта 2012 г.

19:50:27

Контакт	Адрес	Город	Область	Индекс	Страна или регион
Б					
Березин Артур	28-я улица, д. 7	Курск	(нет)	999999	Россия
Борисов Сергей	25-я улица, д. 7	Саратов	(нет)	999999	Россия
В					
Важин Филип	17-я улица, д. 4	Сочи	(нет)	999999	Россия
Верный Григорий	2-я улица, д. 12	Иваново	(нет)	999999	Россия
Володин Виктор	11-я улица, д. 12	Тюмень	(нет)	999999	Россия
Вронский Юрий	7-я улица, д. 12	Воронеж	(нет)	999999	Россия
Г					
Горноженко Дмитрий	14-я улица, д. 4	Вологда	(нет)	999999	Россия
Грачев Николай	10-я улица, д. 12	Саратов	(нет)	999999	Россия
Е					
Егоров Владимир	3-я улица, д. 12	Уфа	(нет)	999999	Россия
Ерёменко Алексей	9-я улица, д. 12	Пермь	(нет)	999999	Россия
Ефимов Александр	19-я улица, д. 7	Уфа	(нет)	999999	Россия
И					
Иванов Андрей	26-я улица, д. 7	Орел	(нет)	999999	Россия
Игнатов Степан	22-я улица, д. 7	Казань	(нет)	999999	Россия
К					
Костерина Ольга	1-я улица, д. 12	Сочи	(нет)	999999	Россия
М					
Мишкова Екатерина	18-я улица, д. 4	Иваново	(нет)	999999	Россия
О					
Омельченко Светлана	4-я улица, д. 12	Москва	(нет)	999999	Россия
Орехов Алексей	11-я улица, д. 12	Орел	(нет)	999999	Россия

Рис. 33. Интерфейсный элемент «Отчеты»

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

Группировка записей может осуществляться как по полному значению (каждой категории соответствует уникальное значение), так и по диапазону значений (по первым знакам, временному или числовому интервалам). В случае необходимости подведения итогов не по отчету в целом, а по группам, выражение, включающее статистическую функцию, размещают не в примечании отчета, а в примечании группы.

8. СОЗДАНИЕ ЗАПРОСОВ

Запросы – служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции, как отбор данных, их сортировку и фильтрацию, а также преобразование данных по заданному алгоритму, создание новых таблиц, автоматическое наполнение таблиц данными, импортированными из других источников, вычисления в таблицах и многое другое. Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную результирующую таблицу. Основой запроса является критерий.

Критерий – это совокупность условий, связанных логическими операторами (И, ИЛИ).

Для создания запросов используется язык SQL (Structured Query Language). Это универсальный компьютерный язык, применяемый для создания и модификации данных, а также управления ими в реляционных БД. SQL является информационно-логическим языком, а не языком программирования и основывается на реляционной алгебре.

Преимущества языка SQL:

- независимость от конкретной СУБД (несмотря на наличие диалектов и различий в синтаксисе, в большинстве своем тексты SQL-запросов могут быть достаточно легко перенесены из одной СУБД в другую);

- наличие стандартов и набора тестов (служит для выявления совместимости и соответствия конкретной реализации SQL общепринятому стандарту, что способствует «стабилизации» языка);

- декларативность (с помощью SQL программист описывает только данные, которые нужно извлечь или модифицировать, а каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса).

Основные (элементарные) инструкции SQL:

- SELECT (возвращает результирующий набор записей);
- FROM (определяет источник);
- WHERE (определяет выражения для условий отбора);
- ORDER BY (определяет поля сортировки).

Язык SQL был задуман как средство работы конечного пользователя, но со временем он стал настолько сложным, что превратился в инструмент программиста.

В связи с этим появился и другой способ создания запросов. QBE (Query by example) – способ создания запросов к БД с использованием образцов в виде текстовой строки, названия документа или списка документов. Система QBE преобразует пользовательский ввод в формальный запрос к БД, что позволяет пользователю делать сложные запросы, освобождая от необходимости изучать более сложные языки запросов, таких как SQL. В Access QBE реализован в виде Конструктора запросов (рис. 34).

Рассмотрим основные разновидности запросов.

Запрос на выборку

Позволяет выбрать данные из полей таблиц, формирующие запрос и удовлетворяющие определенному критерию. Например, в случае написания условия *Between 170 And 200* извлекутся все записи, для которых значения указанного поля находятся в указанном интервале.

Если необходимо, чтобы данные, отображенные в результате работы запроса на выборку, были упорядочены по какому-либо полю или сразу по нескольким полям, применяют сортировку по возрастанию или по убыванию.

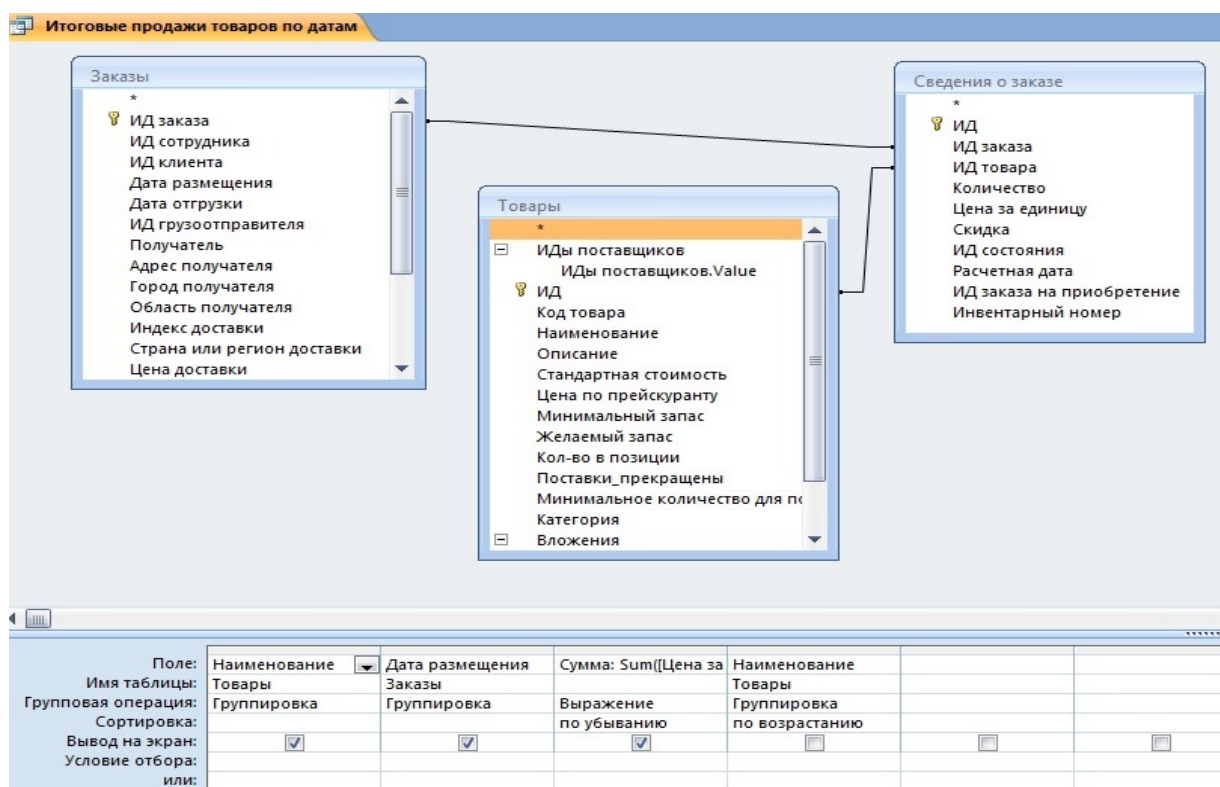


Рис. 34. Конструктор запросов

В нижней части бланка запроса по образцу имеется строка Вывод на экран. По умолчанию предполагается, что все поля, включенные в запрос, должны выводиться на экран, но это не всегда целесообразно. Например, бывают случаи, когда некое поле необходимо включить в запрос потому, что оно является полем сортировки, но в то же время нежелательно, чтобы пользователь видел его содержание. В таких случаях отображение содержимого на экране подавляют сбросом флажка Вывод на экран. Примером может быть запрос на вывод списка сотрудников предприятия, отсортированный по количеству дней, пропущенных по болезни. Он позволит каждому оценить свое положение в этом списке, но не позволит точно узнать, кто и сколько дней болел.

При необходимости в запросе на выборку можно создавать новые вычисляемые поля. Так, если в таблице существуют поля Цена и Количество проданных товаров, то можно определить Стоимость заказа, создав новое поле с выражением Стоимость: [Цена]*[Количество].

Иногда возникает ситуация, когда критерий запроса постоянно нуждается в изменении при сохранении его структуры. Например, необходимо регулярно узнавать успеваемость разных студентов. Поскольку пользователь сам изменять критерий запроса не может, разработчику придется создать запросы на все возможные фамилии, что невыполнимо. В этом случае может помочь запрос с параметрами. При выполнении такого запроса критерий изменяется самим пользователем при каждом запуске. Например, при записи в качестве условия [Введите фамилию студента:] пользователь увидит окно ввода с соответствующей надписью и введет интересующую его фамилию.

При необходимости над результатом выборки можно производить фильтрацию. Например, если необходимо определить пять самых высокооплачиваемых сотрудников, можно произвести в запросе сортировку по убыванию по полю «Зарплата» и указать на необходимость вывода пяти первых значений (заменив на панели инструментов «Набор значений» со слова *Все* на число 5).

Если необходимо определить промежуточные итоги по группам записей, можно создать запрос с группировкой. Для этого используется режим группировки и статистические функции *Sum* (сумма), *Avg* (среднее), *Count* (количество), *Max* (максимум), *Min* (минимум). Например, для того чтобы подсчитать количество лиц женского пола, необходимо сделать группировку по Полу и использовать функцию *Count* по фамилии.

Перекрестные запросы

Перекрестные запросы очень близки сводным таблицам в Excel и позволяют создавать результирующие таблицы на основе результатов расчетов, полученных при анализе группы таблиц. Результаты статистических расчетов группируются по двум наборам данных, один из которых расположен в левом столбце таблицы, а второй – в верхней строке.

В некоторых случаях необходимо сохранить результат запроса в БД для последующей обработки. При этом используются запросы на создание таблицы. При повторном выполнении такого запроса старая таблица будет удалена, а на ее месте появится новая таблица с тем же именем.

Запрос на удаление

Позволяет автоматизировать процесс удаления записей из одной или нескольких таблиц в соответствии с критерием. Запрос можно использовать, например, при отчислении студентов по результатам сессии.

Запрос на добавление

Добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц. Запрос может быть полезен, например, при

переводе студентов, защитивших диплом из таблицы «Студенты» в таблицу «Выпускники».

Запрос на обновление

Данный тип запроса используется в тех случаях, когда необходимо выполнять отбор записей с последующим изменением для них значения определенного поля. Например, если нужно проиндексировать заработную плату сотрудников, можно делать это не вручную, меняя значения в поле Зарплата, а используя запрос на обновление.

В том случае, если необходимо извлечь данные сразу из нескольких таблиц, используют **составные запросы**. Они отличаются от простых только тем, что в качестве источника запроса используется не одна, а несколько таблиц. При этом запросные связи, которые устанавливаются внутри данного запроса, могут отличаться от структурных связей в схеме данных.

При создании составных запросов необходимо правильно определять параметры объединения.

Можно установить один из трех параметров:

1) объединение только тех записей, в которых связанные поля совпадают;

2) левое внешнее соединение (объединение всех записей со стороны «один» и только тех записей со стороны «много», в которых связанные поля совпадают);

3) правое внешнее соединение (объединение всех записей со стороны «много» и только тех записей со стороны «один», в которых связанные поля совпадают).

По умолчанию используется первый тип объединения.

Приведем пример. Пусть запросу необходимо извлечь информацию о фамилии, имени, отчестве и дате рождения абитуриента, а также об оценке, полученной им на экзамене. Добавив все требуемые поля в запрос, получим необходимую информацию. Однако если абитуриент подал документы, а экзамены еще не сдавал, то запрос не даст результата. Это объясняется тем, что в таблице «Экзамены» нет соответствующего внешнего ключа. Изменив параметр объединения на «левое внешнее», увидим всю информацию об абитуриенте, а вместо оценки – пустую ячейку.

Для ускорения выполнения запросов рекомендуется использовать процедуру индексации полей, в которых часто происходит поиск. Записи в таблице представляют собой последовательную структуру и хранятся в том порядке, в каком осуществлялся ввод. В этом случае при выполнении запроса необходимо осуществлять поиск последовательным перебором всех значений, что занимает много времени.

Основной принцип индексации состоит в том, что от базовых таблиц никакой упорядоченности не требуется. Все записи в таблицы вносятся

только в естественном порядке по мере их поступления, т. е. в неупорядоченном виде. А для ускорения поиска используются индексные файлы.

Индексный файл – это специальная служебная таблица. Она содержит значения атрибута, для которого создается индекс, и хранит ссылки на строки, в которых указано данное значение. Таким образом, получаем произвольный доступ ко всем записям.

Часто для ускорения процесса поиска той или иной записи необходимо создавать индекс на основании не одного, а нескольких полей таблицы одновременно, при этом индекс называется **составным**.

9. ИСПОЛЬЗОВАНИЕ СРЕДСТВ АВТОМАТИЗАЦИИ РАБОТЫ БД

В MS Access используются два инструмента для автоматизации работы с БД – макросы и модули. Эти категории объектов предназначены как для автоматизации повторяющихся операций при работе с системой управления базами данных, так и для создания новых функций путем программирования.

Макросы. В СУБД Microsoft Access макросы состоят из последовательности внутренних команд (макрокоманд) СУБД и являются одним из средств автоматизации работы с базой.

Макросы могут быть полезны для автоматизации часто выполняемых задач. Например, при нажатии пользователем кнопки можно запустить макрос, который распечатает отчет или откроет форму. В отличие от Word и Excel, в Access макросы создаются путем выбора последовательности стандартных макрокоманд. Количество возможных макрокоманд невелико, около 50. В основном это открытие, переход, запуск.

Макрос может быть как собственно макросом, состоящим из последовательности макрокоманд, так и группой макросов. Если макросов много, объединение родственных макросов в группы может упростить управление базой данных.

В некоторых случаях требуется выполнять макрокоманду или серию макрокоманд только при выполнении некоторых условий. Например, если в макросе проверяется соответствие данных в форме условиям на значение, то для одних значений может потребоваться вывести одно сообщение, а для других значений – другое сообщение. В подобных случаях условия позволяют определить порядок передачи управления между макрокомандами в макросе.

Модули. Модуль – это набор описаний, инструкций и процедур на языке VBA (Visual Basic for Application), собранных в одну программную единицу и сохраненных под общим именем. Это одно из средств, при помощи которых разработчик базы может заложить в нее нестандартные

функциональные возможности, удовлетворить специфические требования заказчика, повысить быстродействие системы управления, а также уровень ее защищенности.

Существуют два основных типа модулей – модули класса и стандартные модули.

К *модулям класса* относят модули форм и отчетов, которые связаны с определенной формой или отчетом. Они часто содержат процедуры обработки событий, запускаемые в ответ на событие в форме или отчете. Процедуры обработки событий используются для управления поведением формы или отчета и их откликом на события, такие как нажатие кнопки.

В *стандартных модулях* содержатся общие процедуры, не связанные ни с каким объектом, а также часто используемые процедуры, которые могут быть запущены из любого окна БД. Основное различие между стандартным модулем и модулем класса, не связанным с конкретным объектом, заключается в области определения и времени жизни. Значение любой переменной или константы, определенной или существующей в модуле класса, не связанном с конкретным объектом, доступно только во время выполнения этой программы и только из этого объекта.

Таким образом, создание пользовательского интерфейса является одним из важнейших этапов создания БД. Если интерфейс легко осваивается персоналом, прост в использовании, интуитивно понятен и устойчив к ошибкам, то пользователи легко научатся извлекать пользу из представленной в нем информации. В то же время, если интерфейс лишен указанных качеств, то работа с такой системой неизбежно будет сопровождаться теми или иными проблемами.

После создания всех объектов происходит наполнение данными и тестирование БД. Затем она переходит в стадию эксплуатации.

СПИСОК ЛИТЕРАТУРЫ

1. Вольфсон, М. Б. Базы данных: учеб. пособие / М. Б. Вольфсон; СПбГУТ. – СПб., 2008. – 92 с.
2. Советов, Б. Я. Базы данных: теория и практика: учебник для вузов / Б. Я. Советов. – М. : Высш. шк., 2005. – 463 с.
3. Базы данных: учебное пособие для вузов. – СПб. : Политехника, 2008. – 172 с.
4. Советов, Б. Я. Информационные технологии : учебник для вузов / Б. Я. Советов. – М. : Высш. шк., 2003. – 262 с.
5. Информатика : учебник / Б. В. Соболев, А. Б. Галин, Ю. В. Панов и др. – Ростов н/Д : Феникс, 2006. – 446 с.
6. Каганов, В. И. Компьютерные вычисления в средах Excel и Mathcad / В. И. Каганов – М. : Горячая линия – Телеком, 2003. – 328 с.
7. Уокенбах, Дж. Microsoft Office Excel 2007. Библия пользователя / Дж. Уокенбах ; пер. с англ. – М. : «И. Д. Вильямс», 2009. – 816 с.
8. Абдулазар, Л. Лучшие методики применения Excel в бизнесе / Л. Абдулазар ; пер. с англ. – М. : «И. Д. Вильямс», 2006. – 464 с.

Вольфсон Михаил Борисович
Стригина Елена Владимировна

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ

Учебное пособие

Редактор *С. Д. Щербакова*
Компьютерная верстка *Е. А. Головинской*

План 2014 г., п. 70

Подписано к печати 25.06.2014 г.
Объем 3,5 усл.-печ. л. Тираж 30 экз. Заказ 486

Редакционно-издательский центр СПбГУТ
191186 СПб., наб. р. Мойки, 61
Отпечатано в СПбГУТ

М. Б. ВОЛЬФСОН, Е. В. СТРИГИНА

***ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ
БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ***

Учебное пособие

**Санкт-Петербург
2014**