

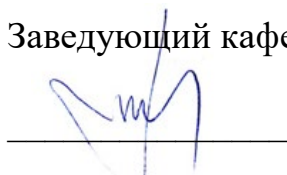
МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тульский государственный университет»

Институт высокоточных систем им. В.П. Грязева  
Кафедра «Тренажерные системы и комплексы»

Утверждено на заседании кафедры  
«Тренажерные системы и комплексы»  
«\_\_» \_\_\_\_\_ 20\_\_ г., протокол №\_\_

Заведующий кафедрой ТСК

 Филиппов В.Н.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ ПО  
ДИСЦИПЛИНЕ  
«Современные IT-технологии»**

**основной профессиональной образовательной программы  
высшего образования – программы бакалавриата**

по направлению подготовки  
**15.03.06 Мехатроника и робототехника**

с направленностью (профилем)  
**Перспективные учебно-тренировочные средства**

Форма обучения: очная

Идентификационный номер образовательной программы: 150306-03-22

Тула 2022 год

**СОДЕРЖАНИЕ**

Введение .....	3
Лабораторная работа № 1 «Организация хранения данных в СУБД MS Access» .....	3
Лабораторная работа № 2 «Создание запросов в СУБД MS Access» ..	12
Лабораторная работа № 3 «Создание форм в СУБД MS Access» .....	18
Лабораторная работа № 4 «Создание отчетов в СУБД MS Access» .....	23
Самостоятельная работа .....	28
Рекомендуемая литература .....	29

## **Введение**

Лабораторный практикум направлен на приобретение навыков разработки баз данных, создания пользовательских запросов и элементов пользовательского интерфейса в среде СУБД MS Access.

Процесс изучения дисциплины направлен на формирование следующих компетенций:

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-4);
- владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных (ПК-2).

В результате изучения дисциплины студент должен:

### ***Знать:***

- историю развития концепции баз данных;
- основные функции современных систем управления базами данных (СУБД);
- методы управления транзакциями;
- классификацию и характеристики моделей данных, лежащих в основе баз данных;
- теорию реляционных баз данных;
- операции реляционной алгебры и реляционное исчисление;
- целостную часть реляционной модели данных;
- методы проектирования реляционных баз данных с использованием нормализации;

### ***Уметь:***

- построить концептуальную информационную модель предметной области в концепции БД;
- реализовать простые информационные технологии с использованием функциональных возможностей современных СУБД (MS Access);
- проектировать реляционную модель данных для выбранной предметной области с использованием нормализации;
- проектировать базу данных для любой предметной области;

- разрабатывать программные объекты для работы с базами данных: экранные формы, отчеты, разрабатывать все виды запросов при помощи построителей запросов;

**Владеть:**

- методикой проектирования баз данных на основе нормализации отношений.
- средствами разработки баз данных и простых элементов пользовательского интерфейса в современных СУБД.

## Лабораторная работа № 1 «Организация хранения данных в СУБД MS Access»

*Тема: Организация хранения данных в СУБД MS Access. Создание таблиц, Построение схемы БД.*

*Раздел дисциплины: Обоснование концепции баз данных.*

*Цель работы: разработать структуру базы данных (БД) для выбранной предметной области, содержащую не менее пяти взаимосвязанных таблиц.*

*Продолжительность: 6 часов.*

### Организация базы данных в среде MS Access

Microsoft Access – это функционально полная реляционная СУБД.

**База данных** в MS Access представляет собой совокупность объектов, хранящихся в одном файле с расширением mdb (рис.1).

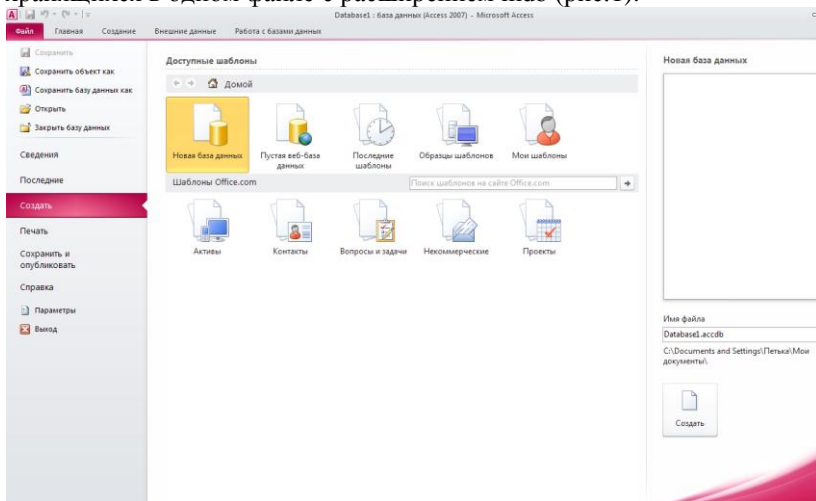


Рис. 1. Окно БД MS Access

Поддерживаются следующие типы объектов: таблицы, формы, запросы, отчеты, макросы, программные модули.

Ниже представлены характеристики БД в СУБД MS Access XP:

- размер файла базы данных Microsoft Access (.mdb) — 2 Гбайт за вычетом места, необходимого системным объектам;
- число объектов в базе данных — 768;
- модули (включая формы и отчеты, свойство Наличие модуля (HasModule) которых имеет значение True) 1 000;
- число знаков в имени объекта — 64;
- число знаков в пароле — 14;
- число знаков в имени пользователя или имени группы — 20;
- число одновременно работающих пользователей — 255;

Основным объектом в БД является **таблица**, хранящая данные о том или ином предмете реального мира. Остальные типы объектов – это различные способы представления информации из таблиц (формы, отчеты, динамические наборы) или действия над таблицами (запросы, макросы, модули).

**Запрос** – это объект, позволяющий как извлекать данные из таблиц с использованием различных критериев, задаваемых пользователем, так и производить различные изменения в таблицах БД. С помощью запроса можно выбрать, изменить или сгруппировать какие-либо данные, содержащиеся в одной или нескольких таблицах. Ответ на запрос также выгладит в виде таблицы и называется **динамическим набором записей**.

**Форма** – это объект, предназначенный для ввода, изменения и просмотра записей в удобном виде на экране. Форма может содержать данные из одной или нескольких взаимосвязанных таблиц, а также не связанные с таблицами данные. Для создания и изменения формы используется методика визуального программирования.

**Отчет** – это объект, предназначенный для печати данных в определенном пользователем виде. Отчет позволяет сгруппировать записи, производить расчеты и выводить как промежуточные, так и полные итоговые значения.


**Макрос** – это набор из одной или нескольких макрокоманд, позволяющих производить различные операции с объектами БД. Например, с помощью макроса при загрузке БД можно автоматически открыть нужные формы или при нажатии кнопки в форме выполнить различные действия (печать формы, открытие другой формы и т.п.) Макрокоманды выбираются из списка стандартных макрокоманд, например.

**Модуль** - это набор процедур и функций на языке Visual Basic. Модули обычно используют для создания достаточно сложных информационных систем. Каждый модуль может быть привязан к объектам форм и отчетам.

Каждый объект имеет структуру, характерную для его типа. Например, таблицы состоят из полей и записей. Формы и отчеты состоят из элементов управления, заголовка и др. Модули состоят из процедур и функций; макросы из макрокоманд. Многие из структурных элементов объектов также считаются объектами.


Все объекты имеют уникальные имена. Имя объекта может состоять из 64 символов, включая пробелы и другие знаки, кроме символов точка (.), восклицательный знак (!), апостроф (‘), квадратные скобки [ ]. Рекомендуется не включать в имена объектов пробелы и избегать слишком длинных имен, что затрудняет программирование приложений.

**Свойство** представляет собой характеристику объекта, например, имя, размер, цвет, тип данных поля и т.п. Свойства текущего объекта сведены в таблицу и доступны для изменения в окне свойств, которое

открывается при нажатии кнопки  на панели инструментов. Набор свойств различен для каждого типа объектов.

Над любым объектом можно выполнить три стандартных действия (им соответствуют три кнопки в окне БД): открыть текущий объект для работы; создать новый объект текущего типа; изменить текущий объект (конструктор).

### **Порядок выполнения лабораторной работы**

Для запуска MS Access выберите иконку  в меню программ MS Windows. Чтобы начать разработку новой базы данных, следует в меню *Файл* выбрать команду *Создать* после чего выбрать пункт *Новая база данных* и присвоить имя новой БД. Затем возможно создание объектов БД “вручную” либо с помощью Мастера, который автоматически генерирует объект в диалоге с пользователем. Независимо от способа создания объекта режим конструктора позволяет в любой момент изменить его структуру и свойства.

#### ***Создание структуры таблиц***

В СУБД MS Access отношение БД называют таблицей, кортежи отношения – записями, атрибуты – полями.

Для создания структуры таблицы в окне База данных необходимо выбрать пункт Таблица и нажать кнопку Создать. В результате откроется диалоговое окно Создание таблицы, в котором следует выбрать

режим Новая таблица. Создание структуры таблицы необходимо производить в режиме конструктора таблиц.

В результате выполнения указанных действий Access выводит на экран окно пустой таблицы в режиме конструктора (рис. 2).

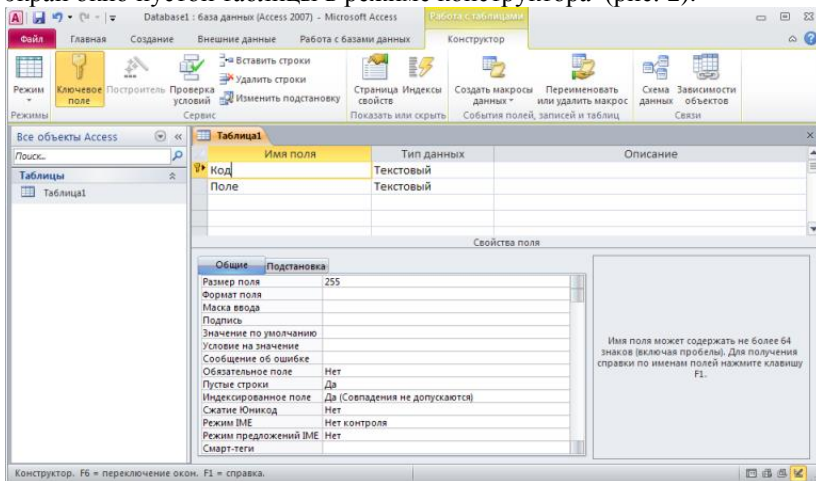


Рис. 2. Новая таблица в режиме конструктора

После того как окно таблицы откроется, активизируется панель инструментов Конструктор таблицы. При определении полей таблицы для каждого поля необходимо ввести имя, тип данных и краткое описание.

Обязательными свойствами каждого поля являются имя, тип и размер. Имя поля задается в столбце *Поле* по тем же правилам, что имена других объектов. Во втором в столбце *Тип данных* открывается список возможных типов данных. Требуемое значение типа данных можно либо выбрать из списка, либо ввести непосредственно с клавиатуры, не прибегая к помощи списка. *Тип данных* определяет, какого вида данные будут храниться в поле – текст, числа, даты и т.д. Важно правильно определить тип поля до того, как начнется ввод данных, в противном случае при изменении типа данные могут быть искажены или утеряны. Ниже приведены типы данных, используемых в СУБД MS Access:

- текстовый (до 255 символов);
- числовой с разной степенью точности;
- дата / время;
- примечания (МЕМО) - до 64000 символов;
- счетчик (для служебных полей, типа КодТовара и т.п.);

- денежный;
- логический (да / нет);
- гиперссылка
- OLE (для хранения данных, сформированных другими прикладными программами - рисунков, схем, звукозаписей, форматированных текстов и т.п.).

Для текстовых и числовых полей пользователь может задать необходимый *размер* поля, при этом следует учитывать специфику хранящихся в конкретном поле данных.

Для каждого поля можно задать *дополнительные свойства* – способ отображения (формат), подпись, используемая в запросах, формах и отчетах, значение поля по умолчанию, правила контроля для ввода данных. Определение этих свойств в ряде случаев позволяет ускорить разработку прикладной программы, описать часть ограничений целостности БД, которые будут проверяться автоматически. Для типов данных *текстовый* и *мемо* может быть задан пользовательский формат ввода значений данных, описание которого приведено в разделе справочной системы Access Форматирование текста.

Описание форматов для различных типов данных представлено в таблице 1.

Описание форматов данных

Таблица 1

Наименование формата	Описание
Для типов данных: <b>Числовой, Денежный</b>	
<i>стандартный формат</i>	устанавливается по умолчанию (разделители и знаки валют отсутствуют)
<i>Денежный</i>	символы валют и два знака после запятой
<i>Евро</i>	Используется денежный формат с символом евро (€) вне зависимости от символа денежной единицы
<i>Фиксированный</i>	выводится, по крайней мере, один разряд
<i>с разделителями разрядов</i>	два знака после запятой и разделители тысяч
<i>Процентный</i>	процент



<i>Экспоненциальный</i>	экспоненциальный формат (например $3.46 \cdot 10^3$ )
Для типа данных <b>Дата/Время</b> существует следующий набор форматов поля:	
<i>длинный формат</i>	Среда, 29 января 2003 г.
<i>средний формат</i>	29 – янв – 03
<b>Наименование формата</b>	<b>Описание</b>
<i>краткий формат</i>	29.01.03
<i>длинный формат времени</i>	10:30:10 PM
<i>средний формат времени</i>	10:30 PM
<i>краткий формат времени</i>	15:30

Для *логического* типа данных используется следующий набор форматов: Да/Нет, Истина/Ложь, Вкл/Выкл.

*Число десятичных знаков* – для числового и денежного типов данных задает число знаков, выводимых после запятой. По умолчанию устанавливается значение Авто, при котором для форматов *денежный*, *фиксированный*, *с разделителем разрядов* и *процентный* выводятся два десятичных знака после запятой. Для формата *стандартный*, число выводимых знаков определяется текущей точностью числовых значений. Можно задать фиксированное число десятичных знаков от 0 до 15.


*Маска ввода* – для *текстового*, *числового*, *денежного* типов данных, а так же для типов *Дата/Время* задается маска ввода, которую пользователь увидит при вводе данных в это поле (например, разделители ( \_ . \_ . \_ ) для поля типа *Дата* ).

Для обеспечения уникальности записей в каждой таблице необходимо наличие первичного ключа – **ключевого поля** таблицы. Общепринятые правила при определении первичного ключа:

- в качестве ключа чаще всего выбирают числовой или символьный код, который используется только для внутренних целей БД и не доступен для изменения пользователем;


- тип ключевого поля – «счетчик» или «числовой».

При необходимости первичный ключ в таблице может состоять из нескольких полей – составной первичный ключ.

Для определения первичного ключа необходимо убедиться, что курсор установлен в поле, которое будет определено как ключ (или выделить несколько полей для составного ключа), затем в меню Прав-  
ка выбрать команду Ключевое поле либо нажать кнопку  на панели инструментов, в результате должен появиться значок ключа слева от имени поля.

### *Связи между таблицами*

Связи между таблицами являются необходимым элементом структуры БД. После определения нескольких таблиц необходимо определить как данные таблицы связаны между собой, для обеспечения полноценной работы с БД – эти связи Access будет использовать в запросах, формах и отчетах.

Для определения связей в БД необходимо в меню *Сервис* выбрать пункт *Схема данных*, либо нажать кнопку  на панели управления. В результате откроется диалоговое окно *Схема данных*, а затем – диалоговое окно *Добавление таблицы*, в котором следует выбрать соединяемые таблицы.

При установлении связи необходимо помнить, что для второй (подчиненной) таблицы должен быть определен внешний ключ – поле, предназначенное для связи с главной таблицей тип данных и размер которого совпадают с полем первичного ключа главной таблицы. Например, для сопоставления сведений о товарах и оплате за проданный товар следует определить связь по полю «Код\_товара» в двух таблицах: «Список товаров» (Код\_товара, Наименование, Единица измерения) и «Оплата» (Код\_товара, Дата\_продажи, Сумма). В первой таблице общее поле является первичным ключом, а во второй – внешним ключом.

Для установления непосредственной связи между двумя выбранными таблицами следует перенести с помощью мыши ключевое поле одной таблицы в другую. В результате откроется диалоговое окно *Связи* (рис. 3)

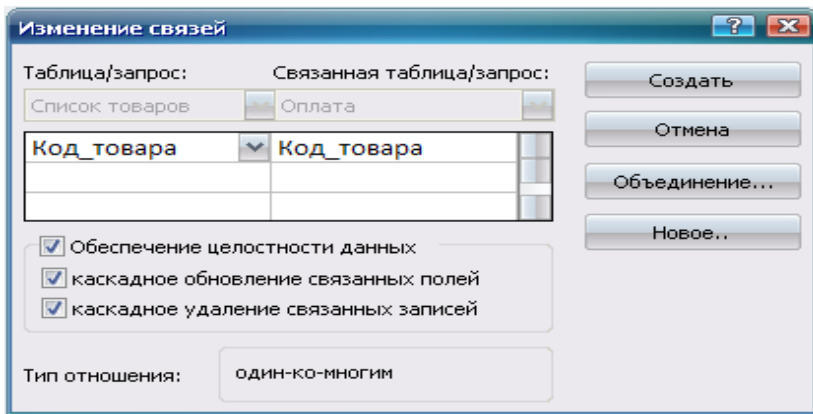


Рис. 3. Окно изменения связей между таблицами

В этом окне следует выбрать опцию Обеспечение целостности данных, после чего выбрать пункты Каскадное обновление связанных полей и Каскадное удаление связанных записей, тем самым, обеспечив требование целостности по ссылкам. Для сохранения связи нажмите кнопку Создать, в результате чего в окне Схема данных будет отображена связь между выбранными таблицами (рис. 4.).

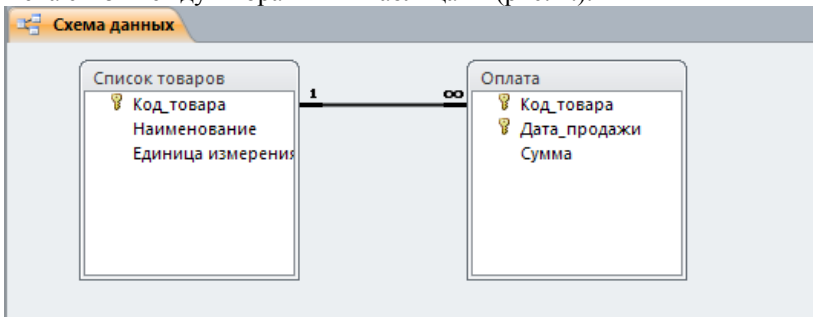


Рис. 4. Определение связей между таблицами

В СУБД MS Access допускаются следующие типы связей:

«один-ко-многим» является наиболее часто используемым типом связи между таблицами. Например, между таблицами «Список Товаров» и «Оплата» существует отношение «один-ко-многим»: товар одного наименования может продаваться различным покупателям, но каждая оплата была произведена за определенный товар.

"многие-ко-многим" реализуется только с помощью третьей таблицы, ключ которой состоит из ключевых полей тех таблиц, которые необходимо связать. Например, между таблицами «Сотрудники» и

«Должности» имеется отношение «многие-ко-многим», поскольку один сотрудник может занимать несколько должностей и на одну должность может занимать несколько сотрудников, такая связь может быть реализована с помощью дополнительной таблицы «Занимаемые должности».

**«один-к-одному».** В этом случае каждая запись в одной таблице может быть связана только с одной записью в другой таблице и наоборот. Этот тип связи используют редко, поскольку такие данные могут быть помещены в одну таблицу. Например, такую связь используют для разделения очень больших по структуре таблиц, для отделения части таблицы по соображениям защиты и т.п.

Следует помнить, что нельзя изменить тип данных для поля, которое связывает таблицу с другой таблицей. Предварительно нужно удалить установленную связь.

После создания схемы базы данных необходимо заполнить созданные таблицы, для чего выберите нужную таблицу и нажмите кнопку Открыть в окне Базы данных.

## **Лабораторная работа № 2 «Создание запросов в СУБД MS Access»**

*Тема: Создание запросов в СУБД MS Access. Создание запросов с помощью визуального средства построителя запросов.*

*Раздел дисциплины: Концепция модели данных*

*Цель работы: создать запросы на выборку, на выборку с параметрами, на обновление записей, на удаление записей в созданных ранее таблицах.*

*Продолжительность: 4 часа.*

### **Типы запросов, создаваемых в Microsoft Access**

В среде MS ACCESS можно создавать следующие типы запросов:

- запросы на выборку;
- запросы с параметрами;
- перекрестные запросы;
- запросы на изменение (запросы на создание таблицы, удаление, обновление, добавление записей);
- запросы SQL (запросы на объединение, запросы к серверу, управляющие запросы, подчиненные запросы).

Наиболее часто используемым запросом является запрос на выборку, возвращающий данные из одной или нескольких таблиц, а также результаты, которые при желании пользователь может изменить.

Запрос на выборку можно использовать для группировки записей, для вычисления сумм, средних значений, пересчета и других действий.

Запрос с параметрами – это запрос, содержащий в себе условие для возвращения записей или значение, которое должно содержаться в поле таблицы. Например, можно создать запрос, в результате которого выводится приглашение на ввод временного интервала. В результате будут возвращены все записи, находящиеся между двумя указанными датами.

Запросы на изменение – это запросы, при запуске которых за одну операцию вносятся изменения в несколько записей. Существует четыре типа запросов на изменение: на удаление, на обновление и добавление записей, а также на создание таблицы.

Запрос на удаление удаляет группу записей из одной или нескольких таблиц. С помощью запроса на удаление можно удалить только всю запись, а не содержимое отдельных полей внутри нее.

Запрос на удаление позволяет удалить записи как из одной таблицы, так и из нескольких таблиц со связями «один-к-одному» или с «один-ко-многим», если при определении связей было установлено каскадное удаление.


Запрос на обновление записей вносит общие изменения в группу записей одной или нескольких таблиц. Например, можно с помощью одного запроса увеличить на 30 процентов стипендию студентов 3-го курса. Запрос на обновление записей позволяет изменять данные только в существующих таблицах.

### **Порядок выполнения работы**

Все запросы, создаваемые в рамках данной лабораторной работы, должны быть реализованы с помощью построителя запросов в режиме конструктора (рис. 5).

Поле:				
Имя таблицы:				
Сортировка:				
Вывод на экран:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора:				
или:				

Рис. 5. Бланк построителя запросов

Для создания нового запроса в окне базы данных (рис. 1) перейдите на вкладку Запросы и нажмите кнопку Создание запроса в режиме конструктора. В появившемся окне (рис. 6) выберите таблицу (таблицы) – источник запроса. Если запрос уже открыт, то для перехода в режим конструктора следует нажать кнопку Вид  на панели инструментов.

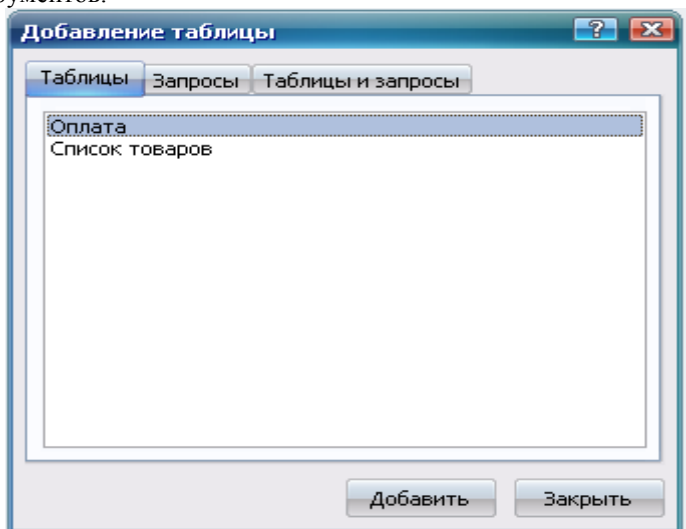




Рис. 6. Добавление таблицы

По умолчанию, любой запрос создаваемый с помощью конструктора является запросом на выборку. Для изменения типа запроса в меню Запрос следует выбрать тип создаваемого запроса либо на панели инструментов нажать кнопку .

Для сохранения запроса необходимо нажать кнопку  на панели инструментов и ввести имя запроса, под которым он будет сохранен.

#### *Создание запроса на выборку*

При создании запроса на выборку, необходимо определить поля, которые будут содержаться в результирующем наборе данных, для этого в строке Имя таблицы (рис.7) выбрать название таблицы, а в строке Поле выбрать названия полей. Для сортировки записей в строке Сортировка можно указать тип сортировки: по возрастанию или по убыванию значений.

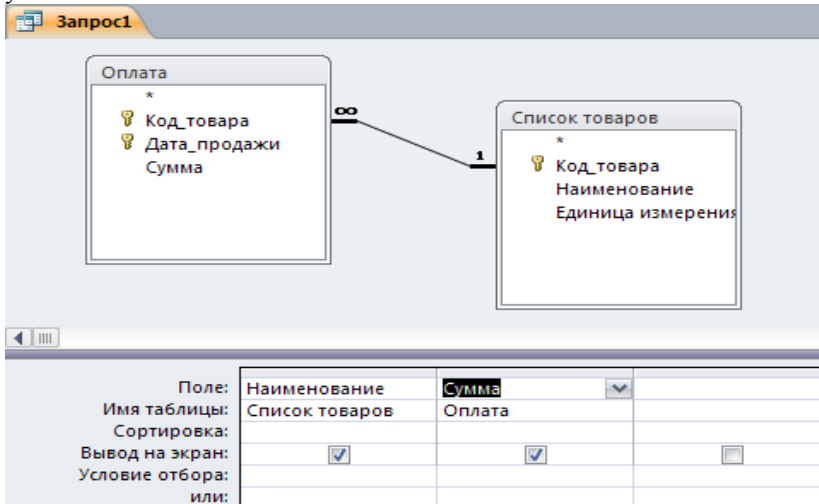



Рис. 7. Запрос на выборку

Связи между таблицами в окне построителя запросов определяются по тому же принципу, что и при определении связей в схеме данных. Для запуска запроса нажмите кнопку Запуск  на панели инструментов.

#### *Создание запроса с параметрами*

Различают два типа запросов с параметрами: с приглашением на ввод условий отбора и с явным указанием условия отбора

Запрос с параметрами отображает одно или несколько определенных диалоговых окон, выводящих приглашение пользователю ввести условия отбора.

Для создания запроса с параметрами создайте новый запрос на выборку, после чего, в режиме конструктора для каждого поля, для которого предполагается использовать параметр, введите в ячейку строки Условие отбора текст приглашения, заключенный в квадратные скобки. Это приглашение будет выводиться при запуске запроса. Текст подсказки должен отличаться от имени поля, но может включать его (рис. 8), введенное в окне приглашения значение будет являться значением параметра.

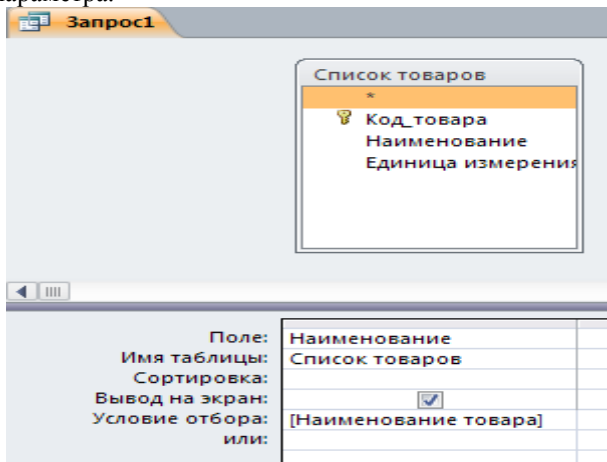



Рис. 8. Запрос на выборку с параметрами

При запуске запроса (рис.8) будет выведена подсказка Наименование товара. Для явного указания условия отбора, в построителе запроса, текстовый параметр необходимо заключить в кавычки: "Сахар", значение числового параметра указывается без дополнительных символов.

#### *Создание запроса на обновление записей*

Для создания запроса на обновление создайте запрос, выбрав таблицу или таблицы, включающие записи, которые необходимо обновить, и поля, которые должны быть использованы в условиях отбора. В режиме конструктора запроса нажмите стрелку рядом с кнопкой Тип запроса  на панели инструментов и выберите команду Обновление.

Выберите поля, значения которых необходимо обновить, после чего в строку Обновление введите выражение или значение, которое должно быть использовано для изменения полей (рис. 9.)



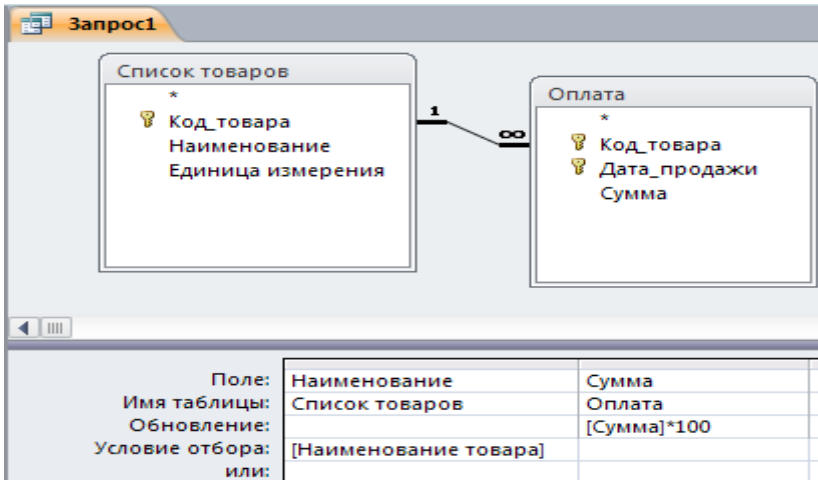





Рис. 9. Запрос на обновление


В строке Условие отбора укажите условие отбора – значения каких записей следует изменить, в противном случае значения выбранных полей будут изменены во всех записях таблицы.


Для просмотра обновляемых записей нажмите кнопку Вид  на панели инструментов. Выводимый список будет содержать значения, которые будут изменены. Для возврата в режим конструктора запроса снова нажмите кнопку Вид  на панели инструментов.

Для запуска запроса нажмите кнопку Запуск  на панели инструментов. Чтобы остановить выполнение запроса, нажмите клавиши Ctrl+Break.

#### *Создание запроса на удаление*

Для создания запроса на удаление создайте запрос, выбрав таблицу, содержащую записи, которые необходимо удалить.

В режиме конструктора запроса нажмите стрелку рядом с кнопкой Тип запроса  на панели инструментов и выберите команду Удаление либо выберите тип запроса в меню Запрос.

Следует помнить, что при удалении записей с помощью запроса на удаление отменить данную операцию невозможно. Следовательно, прежде чем выполнить данный запрос, необходимо просмотреть выбранные для удаления данные. Для этого на панели инструментов нажмите кнопку Вид  и просмотрите в режиме таблицы удаляемые записи.

Для избежания удаления всех записей из таблицы в условии отбора следует указать значение условия для выборки удаляемых записей.

В результате выполнения запроса на удаление будут удалены записи из подчиненных таблиц, для которых установлено Каскадное удаление связанных записей.

### **Лабораторная работа № 3 «Создание форм в СУБД MS Access»**

*Тема: Создание форм в СУБД MS Access. Создание экранных форм и их использование для ввода данных.*

*Раздел дисциплины: Реляционная модель.*

*Цель работы: создать ленточную, табличную и сложную формы в базе данных MS Access, используя в качестве источника записей созданные ранее таблицы и запросы.*

*Продолжительность: 4 часа.*

#### **Формы MS Access**

Формы являются типом объектов базы данных, и используются для отображения и ввода данных в таблицы БД. Форму можно также использовать как кнопочную форму, открывающую другие формы или отчеты БД, а также как пользовательское диалоговое окно.

Обычно для формы выбирается источник записей. В базе данных Microsoft Access источником записей может быть таблица, запрос или инструкция SQL. В проекте Microsoft Access источником записей может быть таблица, представление, инструкция SQL или сохраненная процедура. Другие выводимые в форме сведения, такие как заголовок, дата и номера страниц и др., сохраняются в макете формы.

Связь между формой и ее источником записей создается при помощи графических объектов, которые называют элементами управления – это объекты графического интерфейса пользователя (такие как поле, флажок, полоса прокрутки или кнопка), позволяющий пользователям управлять приложением. Элементы управления используются для отображения данных или параметров, для выполнения действий, либо для упрощения работы с интерфейсом пользователя. Наиболее часто используемым для вывода и ввода данных типом элементов управления является поле.

Существует несколько типов форм: формы можно открывать в виде таблицы, линейной формы и в режиме простой формы. В этих режимах пользователи могут динамически изменять макет формы для изменения способа представления данных. Существует возможность упорядочивать заголовки строк и столбцов, а также применять фильтры к полям. При каждом изменении макета сводная форма немедленно

но выполняет вычисления заново в соответствии с новым расположением данных.

Форму можно создать тремя различными способами:

- При помощи *автоформы* на основе таблицы или запроса. С помощью автоформ можно создавать формы, в которых выводятся все поля и записи базовой таблицы или запроса. Если выбранный источник записей имеет связанные таблицы или запросы, то в форме также будут присутствовать все поля и записи этих источников записей.

- При помощи *мастера* на основе одной или нескольких таблиц или запросов. Мастер задает подробные вопросы об источниках записей, полях, макете, требуемых форматах и создает форму на основании полученных ответов.

- Вручную в режиме *конструктора*. Сначала создается базовая форма, которая затем изменяется в соответствии с требованиями в режиме конструктора.

В ходе выполнения лабораторной работы должны быть созданы *ленточная, табличная и сложная (содержащая подчиненную)* формы.

Ленточная форма – это форма, в которой поля, образующие одну запись, расположены в одной строке; их подписи выводятся один раз в верхней части (заголовке) формы.

Табличная форма – это форма, в которой поля записей расположены в формате таблицы, где каждой записи соответствует одна строка, а каждому полю один столбец. Имена полей служат заголовками столбцов.

*Подчиненной* формой называют форму, вставленную в другую форму. Первичная форма называется *главной* формой, а форма внутри формы называется *подчиненной*. Комбинацию «форма/подчиненная форма» часто называют также иерархической формой или комбинацией «родительской» и «дочерней» форм.

Подчиненные формы особенно удобны для вывода данных из таблиц или запросов, связанных с отношением «один-ко-многим». Например, можно создать форму с подчиненной формой для вывода данных из таблицы «Типы» и из таблицы «Товары». Данные в таблице «Типы» находятся на стороне «один» отношения. Данные в таблице «Товары» находятся на стороне «многие» отношения — каждый тип может иметь несколько товаров. В главной форме отображаются данные на стороне отношения «один». В подчиненной форме отображаются данные на стороне отношения «многие».

Главная форма и подчиненная форма в этом типе форм связаны таким образом, что в подчиненной форме выводятся только те записи, которые связаны с текущей записью в главной форме. Например, когда

главная форма отображает тип «Напитки», подчиненная форма отображает только те товары, которые входят в тип «Напитки».

### Порядок выполнения работы

Создание ленточной табличной и макета сложной формы следует осуществлять с помощью мастера форм. Мастер задает подробные вопросы об источниках записей, полях, макете, требуемых форматах и создает форму на основании полученных ответов.

Для создания ленточной формы с помощью мастера форм в окне базы данных перейдите на вкладку Формы и выберите пункт Создание формы с помощью мастера. В предложенном окне (рис. 10) выберите таблицу или созданный ранее запрос на выборку, который будет использоваться в качестве источника записей формы, выберите поля таблицы/запроса, которые будут доступны для редактирования в создаваемой форме. Затем нажмите кнопку Далее.

Создание форм

Выберите поля для формы.

Допускается выбор нескольких таблиц или запросов.

Таблицы и запросы

Таблица: Оплата

Доступные поля:

Сумма

Выбранные поля:

Код\_товара  
Дата\_продажи

Отмена < Назад Далее > Готово

Рис. 10. Первое окно мастера форм

В следующем окне (рис. 11) выберите тип формы Ленточный, нажмите кнопку Далее и следуйте дальнейшим указаниям мастера форм.

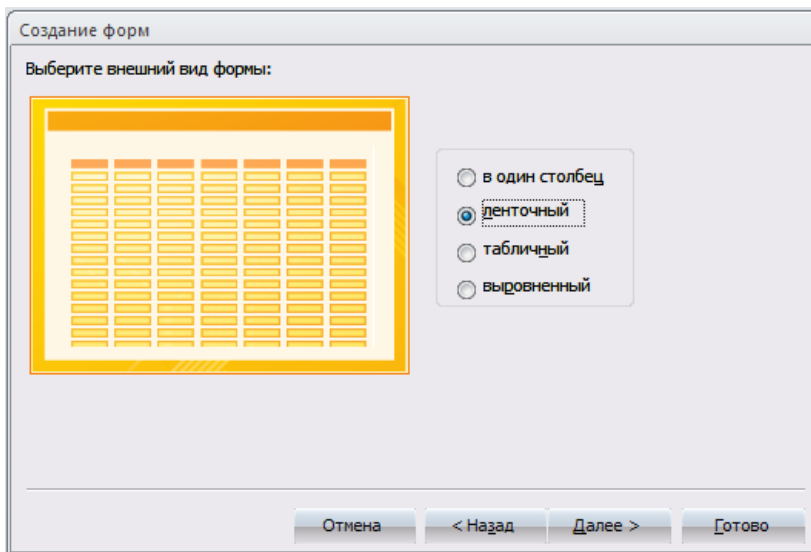


Рис. 11 Второе окно мастера форм

Для создания табличной формы с помощью мастера, следуйте предложенным выше инструкциям, а в окне (рис. 11.) выберите тип формы *табличный*. При этом рекомендуется табличную форму создавать с учетом того, что она может являться подчиненной формой в сложной форме, т.е. источником записей должна быть таблица, содержащая внешний ключ, т.е. связанная с другой таблицей М:1.

Для создания главной формы в окне (рис. 11.) выберите тип формы *в один столбец*, учитывая, что источником записей должна быть таблица, соединенная в схеме данных с подчиненной таблицей с типом связи 1:М.

Созданные формы будут отображены в окне базы данных в разделе *Формы*, их макет можно изменить в режиме конструктора для чего необходимо выбрать нужную форму и нажать кнопку *Конструктор* в окне БД.

В созданной ленточной форме создайте кнопки для навигации по записям и кнопку закрытия формы. Для этого откройте форму в режиме конструктора и в панели инструментов выберите объект *Кнопка* (рис. 12) и поместите её в область примечания формы.

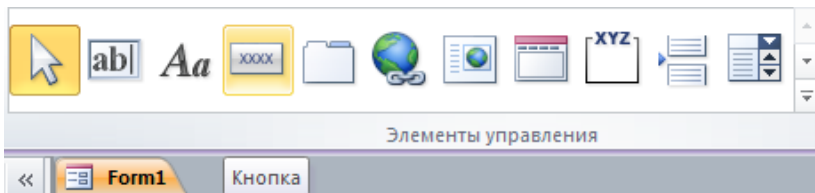


Рис. 12. Панель элементов

После этого откроется окно мастера кнопок (рис. 13) в котором необходимо выбрать категорию *переходы по записям* и в действиях – *первая запись* для перехода к первой записи набора данных, затем нажмите кнопку *Далее* и следуйте дальнейшим инструкциям мастера.

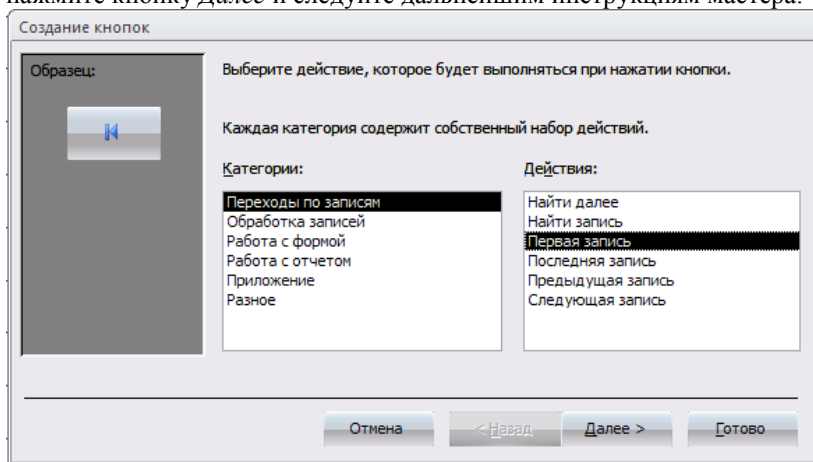


Рис. 13. Мастер кнопок

Аналогично создайте кнопки для перехода на последнюю, предыдущую и следующую запись. При создании кнопки для закрытия формы в мастере кнопок выберите категорию *работа с формой* и действие *закрытие формы*.

Создать подчиненную форму можно либо на основе имеющихся в БД таблиц и запросов либо на основе созданных ранее форм, учитывая при этом, что и главная, и подчиненная формы должны иметь общие поля, необходимые для установления связи между ними.

Для создания подчиненной формы откройте созданную ранее главную форму в режиме конструктора и в панели инструментов выберите объект подчиненная форма и вставьте его в главную форму, поле чего откроется окно мастера подчиненных форм (рис. 14)

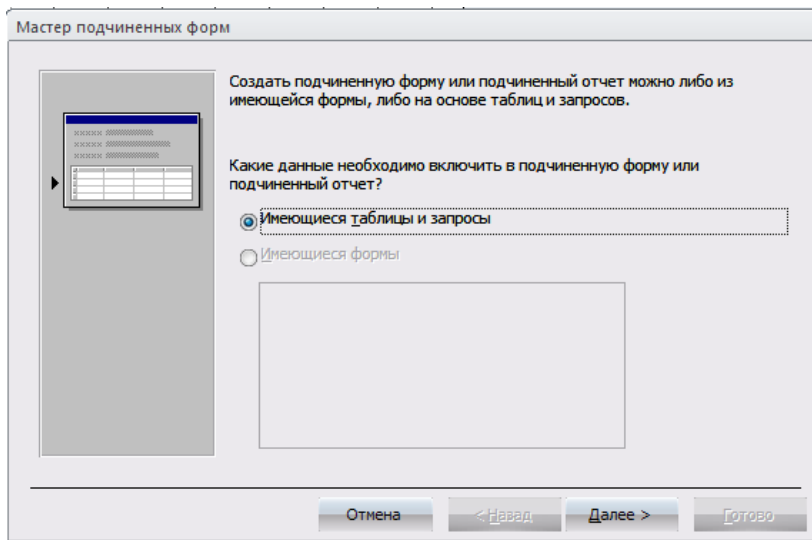


Рис. 14. Мастер подчиненных форм

Выберите из списка предложенных форм созданную ранее форму в табличном виде, при условии, что она удовлетворяет всем требованиям для подчиненных форм, в противном случае создайте подчиненную форму на основе имеющихся таблиц и запросов. Нажмите кнопку *Далее* и следуйте указаниям мастера.

В главной форме создайте кнопки перехода по записям и кнопку закрытия формы, так же как и для ленточной формы, кроме того, создайте кнопку *Добавить новую запись*, по нажатию которой добавлялась бы пустая запись в главную форму, для этого в мастере кнопок выберите категорию *обработка записей* и действие *добавить запись*. Создайте также кнопку, нажатие которой вызовет открытие созданной ранее ленточной формы, для этого в мастере кнопок выберите категорию *работа с формой* и действие *открытие формы*.

#### Лабораторная работа № 4 «Создание отчетов в СУБД MS Access»

*Тема: Создание отчетов в СУБД MS Access. Создание отчетов и их использование для вывода информации.*

*Раздел дисциплины: Реляционная модель.*

*Цель работы: создать отчеты в базе данных MS Access, используя в качестве источника записей созданные ранее таблицы и запросы.*

*Продолжительность: 4 часа.*

### **Отчеты MS Access**

Отчет — это гибкое и эффективное средство для организации данных при выводе на печать. С помощью отчета имеется возможность вывести данные на печать в том виде, в котором они требуются конкретному пользователю.

Основные данные для формирования отчета берется из базовой таблицы, запроса или инструкции SQL, являющихся источниками данных для отчета. Другие сведения (заголовки, примечания отчетов, количество страниц и другая сопроводительная информация) вводятся при разработке отчета.

Пользователь имеет возможность разработать отчет самостоятельно или создать отчет с помощью мастера. Мастер по разработке отчетов MS Access выполняет всю рутинную работу и позволяет быстро разработать макет отчета. Работа мастера отчетов аналогична Работе других мастеров в среде MS Access. После создания основной части отчета разработчик может переключиться в режим конструктора и внести изменения в стандартный макет.

Данные в отчете могут быть сгруппированы по различным полям, например, сведения о продаже товара могут быть сгруппированы по месяцам, либо по наименованию товара. При этом в отчете можно указать способ сортировки выводимых на печать данных. Кроме этого, в отчете, по тому же принципу, как и в форме, можно создать подчиненный отчет.

При включении в главный отчет подчиненного отчета, содержащего данные, относящиеся к данным в главном отчете, необходимо установить связь подчиненного отчета с главным. Связь обеспечивает соответствие записей, выводящихся в подчиненном отчете, записям в главном отчете.

При создании подчиненного отчета с помощью мастера или путем переноса с помощью мыши отчета или таблицы в другой отчет, Microsoft Access автоматически выполняет синхронизацию главного и подчиненного отчета в следующих случаях:

Отчеты базируются на таблицах, между которыми установлены связи в окне Схема данных. При создании отчетов на основе запроса или запросов синхронизация отчета с подчиненным отчетом выполняется автоматически, если связи установлены для базовых таблиц запроса или запросов. Если связи базовых таблиц установлены корректно, Microsoft Access выполнит синхронизацию главного и подчиненного отчетов автоматически.



Главный отчет базируется на таблице, содержащей ключевое поле, а подчиненный отчет базируется на таблице, содержащей поле с тем же именем и с тем же или совместимым типом данных. Это же условие должно выполняться для базовых таблиц запросов, если отчеты базируются на запросах.

Перед созданием отчета убедитесь, что на компьютере настроен используемый по умолчанию принтер.

### Порядок выполнения работы

В ходе выполнения работы необходимо создать отчет, содержащий подчиненный отчет.

Для создания отчета в окне базы данных выберите вкладку Отчеты и нажмите кнопку Создание отчета с помощью мастера.

В первом окне мастера отчетов выберите имя таблицы или запроса, содержащих данные, по которым строится отчет. MS Access по умолчанию использует эту таблицу или запрос как базовый источник данных для отчета. Для удобства используйте тот же источник записей, что и для главной формы в предыдущей лабораторной работе.

В следующем окне (рис. 15) установите уровни группировки данных (если необходимо). Нажмите кнопку *Далее*.

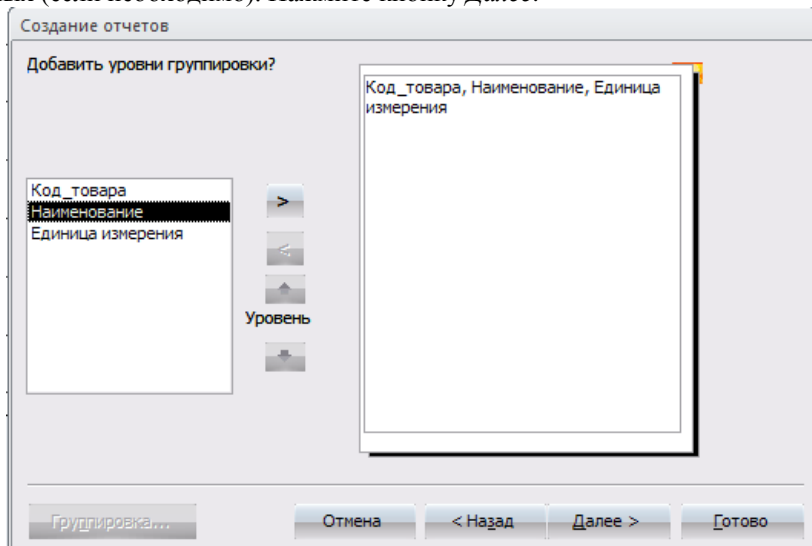


Рис. 15. Мастер отчетов, окно группировки

В следующем окне выберите способ сортировки данных в отчете, нажмите кнопку *Далее* и следуйте дальнейшим инструкциям мастера отчетов.

Для изменения макета отчета откройте его в режиме конструктора (рис. 16).

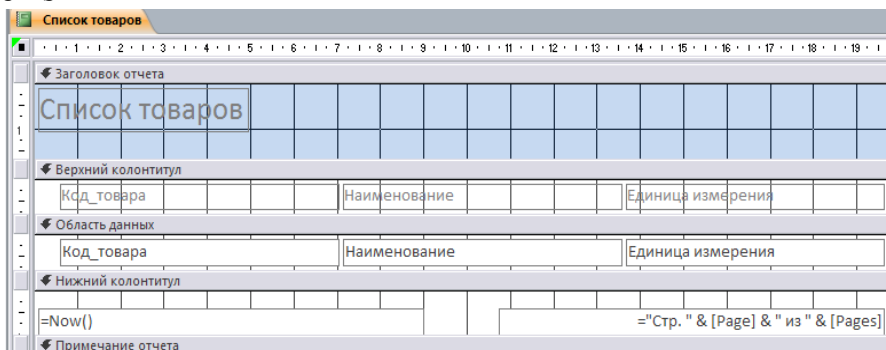







Рис. 16. Отчет, открытый в режиме конструктора


Чтобы изменить внешний вид отчета, нажмите кнопку Автоформат  на панели инструментов и выберите новый внешний вид для отчета.

Для изменения внешнего вида одного элемента управления, например надписи, выделите его, после чего на панели инструментов Форматирование  выберите другой шрифт, размер шрифта или другие параметры.

Для изменения формата отображения данных в элементе управления, например в поле, убедитесь, что данный элемент выделен и нажмите кнопку Свойства  на панели инструментов для вывода окна свойств.

При необходимости добавьте в отчет другие поля из таблицы, являющейся источником данных отчета следующим образом: нажмите кнопку Список полей  на панели инструментов для отображения списка всех полей базовой таблицы и с помощью мыши «перенесите» выбранное поле в отчет.

Создайте заголовок и примечание отчета, куда добавьте надпись, содержащую Ваши ФИО и № группы. Чтобы создать надпись, нажмите кнопку Надпись  на панели элементов. Затем выберите в отчете место, куда ее следует поместить, введите нужный текст и нажмите клавишу Enter.

Поменяйте порядок сортировки и группировки данных в отчете, для этого нажмите кнопку Сортировка и группировка  на панели инструментов для вывода диалогового окна (рис. 17).

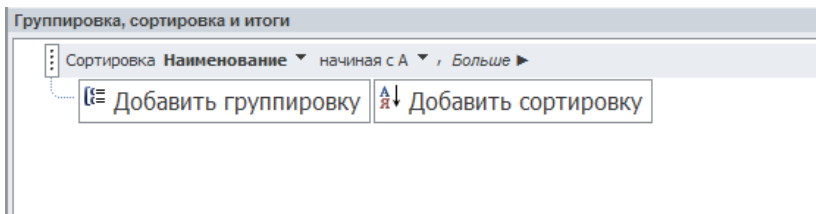



Рис. 17. Окно сортировки и группировки данных

Добавьте новые поля для группировки и сортировки, создайте заголовки и примечание групп.

Для создания подчиненного отчета нажмите кнопку Подчиненная форма/отчет  на панели элементов, затем установите указатель в отчете на том месте, куда требуется поместить подчиненный отчет, и нажмите кнопку мыши. Выполняйте инструкции, выводющиеся в диалоговых окнах мастера.

После нажатия кнопки Готово элемент управления «Подчиненная форма/отчет» будет вставлен в главный отчет. Кроме того, будет создан отдельный отчет, выводющийся как подчиненный отчет.

При включении в главный отчет подчиненного отчета, содержащего данные, относящиеся к данным в главном отчете, необходимо установить связь подчиненного отчета с главным. Связь обеспечивает соответствие записей, выводящихся в подчиненном отчете, записям в главном отчете.

При создании подчиненного отчета с помощью мастера или путем переноса с помощью мыши отчета или таблицы в другой отчет, MS Access автоматически выполняет синхронизацию главного и подчиненного отчета в следующих случаях:

Так же как и при создании подчиненных форм следует учитывать, что отчеты базируются на таблицах, между которыми установлены связи в окне Схема данных. Для подчиненного отчета используйте источник записей такой же, как и для подчиненной формы. Помните, что при создании отчетов на основе запроса или запросов синхронизация отчета с подчиненным отчетом выполняется автоматически, если связи установлены для базовых таблиц запроса или запросов. Если связи базовых таблиц установлены корректно, MS Access выполнит синхронизацию главного и подчиненного отчетов автоматически.

Главный отчет базируется на таблице, содержащей ключевое поле, а подчиненный отчет базируется на таблице, содержащей поле с тем же именем и с тем же или совместимым типом данных. Это же условие должно выполняться для базовых таблиц запросов, если отчеты базируются на запросах.

При связывании главного и подчиненного отчетов Microsoft Access использует свойства Основные поля (LinkMasterFields) и Подчиненные поля (LinkChildFields) элемента управления «Подчиненная форма/отчет». Если по каким-либо причинам Microsoft Access не связывает главный и подчиненный отчет автоматически, пользователь имеет возможность задать значения этих свойств самостоятельно.

Для открытия отчета в сложной форме создайте кнопку открытия отчета для просмотра, используя для этого мастер кнопок, по методике, изложенной в лабораторной работе № 3.

### **Самостоятельная работа**

Согласно рабочей программе отводится следующее количество часов на самостоятельную работу:

- подготовка к контрольным работам – 9 часов.
- подготовка к лабораторным работам – 18 часов.
- индивидуальное задание: Моделирование концептуального представления предметной области – 6 часов.

Форма контроля и проверка достижения заявленных компетенций (ОПК-4, ПК-2): проведение контрольных работ (в том числе тестовых), опрос перед проведением лабораторных работ, проверка отчетов, защита индивидуального задания путем представления презентации – выступление на лекции с демонстрацией примера нормализации.

Для проработки лекционного материала студентам, помимо конспектов лекций, рекомендуются следующие главы учебно-методического пособия [1] по разделам курса:

1. Обоснование концепции баз данных: глава 1.
2. Концепция модели данных: глава 2.
3. Реляционная модель: главы 3-4, глава 5.1.

Для подготовки к лабораторным работам рекомендуется повторить соответствующие тематике разделы учебно-методического пособия [1], а также ознакомиться с порядком выполнения лабораторных работ, по настоящему руководству.

Для изучения тем теоретической части курса, отводимых на самостоятельную проработку, рекомендуется ознакомление со всеми разделами [1]. Кроме того, рекомендуется повторить разделы предложенной литературы [2-7], посвященные проектированию данных и построению пользовательских приложений.

Для выполнения индивидуального задания необходимо:

- выбрать предметную область и провести ее анализ на предмет выявления основных объектов и их характеристик;

- разработать концептуальное представление данных для выбранной предметной области;
- привести глоссарий концептуального представления с описанием основных объектов предметной области

### Рекомендуемая литература

1. Сенченко П. В. Организация баз данных: учеб. пособие / П.В. Сенченко. — Томск: факультет дистанционного обучения ТУ-СУРа, 2015. — 170 с. ил. [Электронный ресурс]. – URL: <https://edu.tusur.ru/training/publications/5179>

2. Сенченко, П.В. Организация баз данных: Учебное пособие/ П. В. Сенченко; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники. – Томск: ТУСУР, 2004. - 184 с.: ил.. - Библиогр.: с. 183-184. - ISBN 5-86889-224-0: УДК 681.3.016(075.8) (гриф СИБРУМЦ) (Наличие в библиотеке ТУСУР: экземпляры всего: 34, из них: англ: 6 , счз1: 1 , счз5: 1, аул: 26.)

3. Дейт К. Дж. Введение в системы баз данных: Пер. с англ./ К. Дж. Дейт. - 6-е изд. - Киев; М.: Диалектика, 1998. - 784 с.: ил. - (Системное программирование). - (в пер.): Б.ц. (наличие в библиотеке ТУСУР: АНЛ – 1 экз.)

4. Саймон, Алан Р. Стратегические технологии баз данных: менеджмент на 2000 год: Пер. с англ./ Алан Р. Саймон; Ред. М. Р. Когаловский, Пер. М. Р. Когаловский, Пер. Н. И. Вьюкова, Пер. Г. Т. Никитина. - М.: Финансы и статистика, 1999. - 480 с.: ил. (наличие в библиотеке ТУСУР: счз1(1), счз5(1))

5. Кузнецов С.Д. Основы современных баз данных. Информационно-аналитические материалы Центра Информационных технологий. М.— режим доступа к сайту <http://citforum.ru/database/osbd/contents.shtml> свободный (дата обращения: 28.02.2012)

6. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие режим доступа к сайту <http://citforum.ru/database/dbguide/index.shtml> свободный (дата обращения: 28.02.2012)

7. Сенченко П.В. Слайды-презентации по дисциплине "Организация баз данных" / "Базы данных", 2012. 270 сл. (учебный сервер каф. АОИ: диск о:\2 курс\Базы данных\Слайды.pdf)

Электронные варианты УМПО находятся в открытом доступе в компьютерных классах.

Для организации работы студентов требуется свободный доступ в компьютерные классы с наличием ОС Windows, MS Office, СУБД MS Access.

Необходимые базы данных, информационно-справочные и поисковые системы: образовательный портал университета (<http://edu.tusur.ru>), электронный каталог библиотеки <http://lib.tusur.ru>; электронные информационно-справочные ресурсы вычислительных залов кафедры АОИ.