

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА  
И ПРОДОВОЛЬСТВИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
АГРАРНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной техники

**А.И. Шакирин, О.М. Львова**

## **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

### **Разработка СУБД в Microsoft Office Access**

*Лабораторный практикум  
по дисциплине «Информационные технологии»,  
разделу «Системы управления базами данных»*

Минск  
БГАТУ  
2010

УДК 004.65 (07)  
ББК 32.81я7  
Ш 17

Рекомендовано научно-методическим советом агроэнергетического факультета

Протокол № 3 от 12 ноября 2009 г.

Рецензенты:

доктор физико-математических наук, профессор кафедры  
«Вычислительные методы и программирование»

БГУИР *С.В. Колосов*;

кандидат технических наук, профессор кафедры  
«Информационные технологии автоматизированных систем»

БГУИР *М.П. Ревотюк*

**Шакирин, А.И.**

Ш 17 Информационные технологии. Разработка СУБД в Microsoft Office Access : лабораторный практикум / А.И. Шакирин, О.М. Львова. – Минск : БГАТУ, 2010. – 124 с.  
ISBN 978-985-519-206-1.

ISBN 978-985-519-206-1

© БГАТУ, 2010

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ . . . . .	4
Лабораторная работа № 1. . . . .	5
Лабораторная работа № 2. . . . .	30
Лабораторная работа № 3. . . . .	38
Лабораторная работа № 4. . . . .	66
Лабораторная работа № 5. . . . .	83
Лабораторная работа № 6. . . . .	93
Лабораторная работа № 7. . . . .	111
ЛИТЕРАТУРА . . . . .	117

## ВВЕДЕНИЕ

Цель дисциплины «Информационные технологии» — подготовка специалистов к использованию современных информационных технологий как инструмента для решения на высоком уровне научных и практических задач в своей предметной области, в том числе:

- обеспечение создания и ведения баз данных по технологической и эксплуатационной наследственности деталей машин и их соединений;
- овладение навыками работы с профессиональными базами данных.

В настоящее время существует большое количество различных программных средств, предназначенных для разработки систем управления базами данных (СУБД). Разработка СУБД в *Microsoft Office Access* связана прежде всего с тем, что СУБД *MS Access* тесно интегрирована с *OC Windows* и другими приложениями. СУБД *MS Access* содержит такие сервисные средства, как проверка орфографии и автозамена, анализ таблиц и быстроедействие, возможность коллективной работы, защита информации. Отмеченные свойства делают СУБД *MS Access* удобным инструментом в работе с базами данных на уровне офиса или небольшого предприятия.

Лабораторный практикум предназначен для приобретения и закрепления навыков разработки СУБД в *Microsoft Office Access* при выполнении лабораторных работ по разделу «Системы управления базами данных» дисциплины «Информационные технологии» студентами агроинженерного, агроэнергетического факультетов и факультета «Технический сервис в АПК».

Практикум содержит методический материал для выполнения 7 лабораторных работ по следующим темам:

- 1) создание структуры таблиц и связей;
- 2) создание простых форм и заполнение таблиц;
- 3) создание запросов на выборку;
- 4) создание запросов на изменение;
- 5) создание простых отчетов;
- 6) разработка сложной формы;
- 7) создание макросов и главной кнопочной формы.

Для закрепления полученных навыков по каждой теме студентам предлагаются 26 вариантов индивидуальных заданий.

## Лабораторная работа №1

### СОЗДАНИЕ СТРУКТУРЫ ТАБЛИЦ И СВЯЗЕЙ

**Цель лабораторной работы:** приобрести навыки создания структуры таблиц базы данных и межтабличных связей.

#### Порядок выполнения лабораторной работы

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания структуры таблиц и установить связи между таблицами. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.

#### 1. Краткие теоретические сведения

**База данных (БД)** – это поименованная совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, не зависящая от прикладных программ.

Кроме данных, база содержит **методы и средства**, позволяющие каждому из пользователей оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным пользователям, образуется информация, которой они пользуются и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

**Система управления базами данных (СУБД)** – это совокупность программ и языковых средств, предназначенных для создания, редактирования и использования БД.

##### 1.1. Общая характеристика СУБД *Microsoft Office Access*

СУБД *Access* поддерживает реляционную модель данных, обеспечивающую логическую связь разных информационных таблиц по значениям однотипных полей, что позволяет корректировать данные, не нарушая их целостность.

**Таблицы** – основные объекты базы данных. В таблицах хранятся все данные, имеющиеся в БД, а также таблицы содержат структуру самой базы данных: поля, их типы и свойства.

**Запросы** – предназначены для извлечения данных из таблиц и предоставления их пользователю в удобном виде.

**Формы** – средства для ввода и просмотра данных. С помощью форм можно закрыть некоторые поля для несанкционированного ввода, можно разместить специальные элементы управления (раскрывающиеся списки, переключатели, флажки и др.) для автоматизации ввода.

**Отчеты** определяют вид документа, предназначенного для вывода на печать.

**Макросы** позволяют связывать отдельные действия, реализуемые с помощью форм, запросов, отчетов, и организовать их выполнение через меню.

Для реализации нестандартных процедур обработки данных пользователь может создать программы на языке *Visual Basic*, объединив их в **модули**.

Таблицы и запросы, формы и отчеты, макросы и модули составляют объектную модель БД. Многочисленные конструкторы, шаблоны и мастера упрощают работу пользователя по созданию новых объектов, сводя ее к ответам на типовые вопросы.

##### 1.2. Таблицы

Все данные представлены в виде простых таблиц, разбитых на строки и столбцы, на пересечении которых находятся данные. В таблицах данные распределяются по столбцам, которые называют **полями**, и строкам, которые называют **записями**. Запись является наименьшей единицей обмена данными между оперативной и внешней памятью, поле – наименьшей единицей обработки данных. Все таблицы называются соответствующими именами.

Создание новой таблицы происходит в два этапа:

- 1) определение структуры (имена, типы и размеры полей);
- 2) создание записей таблицы и заполнение их данными.

Готовую структуру таблицы можно позднее модифицировать, например, добавить или удалить поля, поменять их порядок и свойства. Однако лучше все исправления вносить до начала заполнения таблицы, поскольку изменения в структуре могут повлечь за собой потерю или искажение данных.

Перечислим **основные свойства полей таблиц СУБД Access**.

- **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.

- **Тип поля** — определяет тип данных, которые могут содержаться в данном поле.

- **Размер поля** — определяет максимальный размер (в символах) данных, которые могут размещаться в данном поле.

- **Формат поля** — определяет способ форматирования данных в ячейках, принадлежащих полю.

- **Маска ввода** — определяет форму, в которой вводятся данные в поле (средство автоматизации ввода данных).

• **Подпись** — определяет заголовок столбца таблицы для данного поля. Если подпись не указана, то в качестве заголовка столбца используется свойство **Имя поля**.

• **Значение по умолчанию** — то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).

• **Условие на значение** — ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).

• **Сообщение об ошибке** — текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных, (проверка ошибочности 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 строчковых данных. От свойства **Обязательное поле** отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например, к текстовым.

• **Индексированное поле** — если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значения в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Поскольку в разных полях могут содержаться данные разного типа, то и свойства у полей могут различаться в зависимости от типа данных. Так, например, список вышеуказанных свойств полей относится в основном к полям текстового типа. Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним и свои. Например, для данных, представляющих действительные числа, важным свойством является количество знаков после десятичной запятой. С другой стороны, для полей, используемых для хранения рисунков, звукозаписей, видеоклипов и других объектов большинство вышеуказанных свойств не имеют смысла.

После создания всех полей таблицы и определения их свойств можно перейти к вводу и редактированию данных.

СУБД Access работает со следующими **типами данных**.

**Текстовый** — тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).

**Поле Мемо** — специальный тип данных для хранения больших объемов текста (до 65 535 символов).

**Числовой** — тип данных для хранения действительных чисел.

**Дата/время** — тип данных для хранения календарных дат и текущего времени.

**Денежный** — тип данных для хранения денежных сумм.

**Счетчик** — специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование — для порядковой нумерации записей.

**Логический** — тип для хранения логических данных (могут принимать только два значения, например Да или Нет).

**Поле объекта** — специальный тип данных, предназначенный для хранения объектов, например мультимедийных.

**Гиперссылка** — специальное поле для хранения адресов *URL Web*-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.

**Мастер подстановок** — это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод в данных поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

### 1.3. Связи между таблицами

С помощью ключевых полей таблицы связывают между собой в единую структуру. Для связывания сведений, хранящихся в разных таблицах, каждая таблица базы данных должна содержать поля или набор полей, однозначно определяющих каждую запись. Такое поле или набор полей называют первичным ключом.

**Первичный ключ (ключевое поле)** — одно или несколько полей (столбцов), комбинация значений которых однозначно определяет каждую запись в таблице. Значение первичного ключа должно быть уникальным, то есть в таблице не должно быть двух или более записей с одинаковым значением первичного ключа. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах.

В таблице «Преподаватель» (рисунок 1.1) в качестве первичного ключа может служить только поле «Таб.№», т.к. значения в этом поле не повторяются. Значения других полей могут повторяться внутри данной таблицы.

Таб.№	ФИО	Уч.степень	Уч.звание	Код кафедры
301	Иванов П.П.	Д-р тех. наук	Профессор	01
302	Иванов Ф.П.	Канд. техн. наук	Доцент	01
303	Петров В.В.	Канд. техн. наук	Доцент	01
304	Тимофеев Т.Т.	Канд. техн. наук	Доцент	02
305	Павлов П.П.	Д-р тех. наук	Доцент	02

Рисунок 1.1 – Данные таблицы «Преподаватель»

**Внешний ключ** – одно или несколько полей (столбцов) в таблице, содержащих ссылку на поле или поля первичного ключа в другой таблице. Поле внешнего ключа определяет способ объединения таблиц.

Если в таблице вообще нет никаких полей, которые можно было бы использовать как ключевые, всегда можно ввести дополнительное поле типа **Счетчик** — оно не может содержать повторяющихся данных по определению.

При формировании нескольких связанных таблиц используется **схема данных**, наглядно отображающая таблицы и связи между ними. Общее поле связывает две таблицы таким образом, что *Access* может объединить данные из двух таблиц для просмотра, редактирования или печати.

Различают три вида связей между таблицами:

- один-к-одному;
- один-ко-многим;
- многие-ко-многим.

Связь между таблицами организуется на основе общего поля, причем в одной из таблиц оно обязательно должно быть ключевым, то есть на стороне “один” должно выступать ключевое поле, содержащее уникальные, неповторяющиеся значения (первичный ключ). Значения на стороне “многие” могут повторяться.

**Связь один-к-одному** связывает две таблицы по их ключевым полям. Это может применяться в тех случаях, когда для одного ключевого поля существует два больших блока информации, отличающиеся по смыслу. Например, паспортные данные студента (первый блок) и его успеваемость (второй блок). В этом случае создаются две таблицы и затем они связываются по ключевому полю, предположим, по номеру зачетной книжки (рисунок 1.2). При этом одной записи в таблице-потомке соответствует одна запись в таблице-предке.

№ зачет.книжки	ФИО	Серия и номер паспорта	№ зачет.книжки	Физика	Матем.	КИТ
74060301	Иванов И.И.	MP1234567	74060301	5	8	7
74060302	Петров П.П.	MH0987655	74060302	6	9	9
74060302	Сергеев С.С.	MP8974021	74060302	7	10	8

Рисунок 1.2 – Связь один-к-одному

**Связь один-ко-многим** является самой распространенной в реляционных базах данных.

На схеме данных эти поля соединены линией связи. С одной стороны эта линия маркирована значком **1** (один), с другой стороны — значком  $\infty$  (бесконечность). Это графический метод изображения связи **один-ко-многим** (рисунок 1.3).

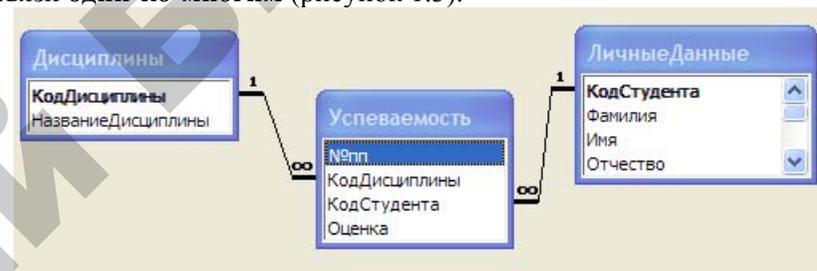


Рисунок 1.3 – Связь один-ко-многим

**Связь многие-ко-многим** реализуется при помощи промежуточной таблицы со связями **один-ко-многим**.

## 2. Пример создания структуры таблиц БД

*Создать базу данных Деканат, содержащую сведения о студентах и их успеваемости по изучаемым дисциплинам. БД должна содержать четыре таблицы.*

Таблица 1 – **Дисциплины** содержит сведения об изучаемых дисциплинах. Таблица должна иметь следующие поля: код дисциплины; наименование дисциплины.

Таблица 2 – **Личные данные** содержит информацию о студенте: шифр студента; ФИО; дату и место рождения; номер телефона, если такой имеется; наличие общежития.

Таблица 3 – **Успеваемость** представляется собой сводную ведомость результатов сдачи сессии каждым студентом. Таблица должна иметь следующие поля: № п/п; код дисциплины; код студента; оценка.

Таблица 4 – **Стипендия** будет использоваться для назначения стипендии студентам в зависимости от результатов сдачи сессии. Таблица должна иметь следующие поля: средний балл – минимальное значение; средний балл – максимальное значение; вид стипендии; размер.

## 2.1. Создание базы данных в СУБД Microsoft Access

Для запуска Access используется последовательность команд: **Пуск** → **Все программы** → **Microsoft Office** → **Microsoft Office Access**. После запуска Access на экране появляется окно, показанное на рисунке 1.4.

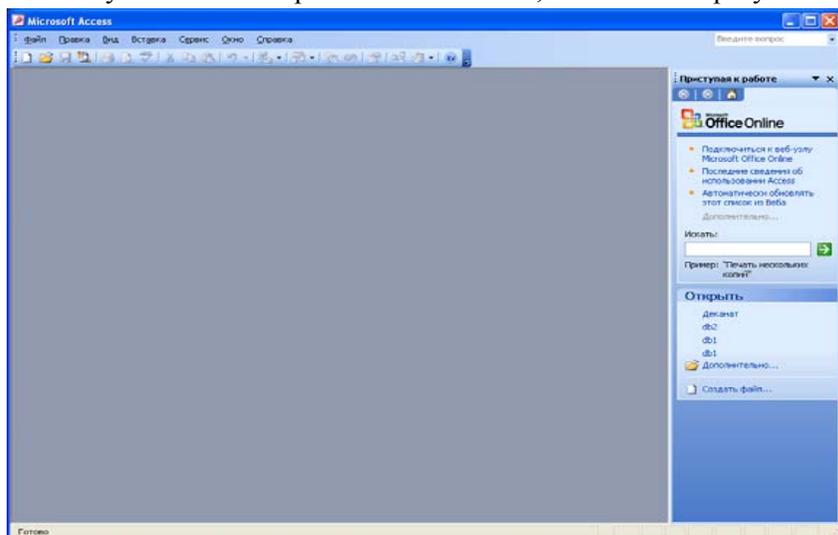


Рисунок 1.4 – Панель интерфейса Microsoft Access

В меню **Файл** укажите команду **Создать...** и затем **Новая база данных...** В открывшемся окне найдите свою папку и сохраните в ней файл под именем **Деканат**. В панели интерфейса Access откроется окно пустой БД **Деканат**, показанное на рисунке 1.5.

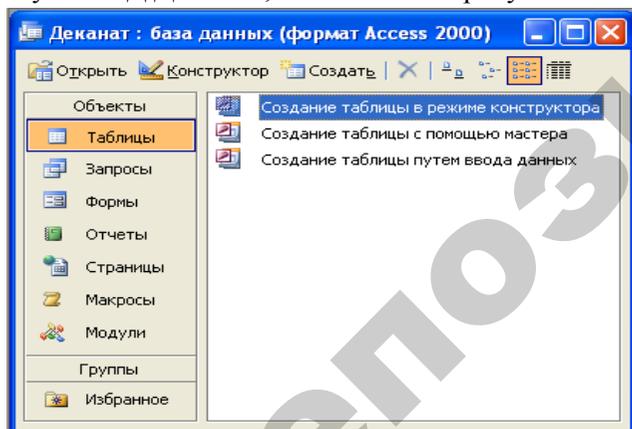


Рисунок 1.5 – Окно пустой базы данных Деканат

В окне БД **Деканат** имеются кнопки, позволяющие перейти к различным объектам БД: **Таблицы**, **Формы**, **Запросы**, **Отчеты** и т. д. Для создания таблицы откройте вкладку **Таблицы**. На экране появится окно, в котором Access предлагает один из трех способов создания таблицы. Выберем наиболее универсальный способ **Создание таблицы в режиме конструктора** и укажем пункт меню **Открыть** или **Конструктор**. На экране появится окно **Таблица1:таблица**, в котором разработчик в столбец **Имя поля** заносит названия полей, а в столбце **Тип данных** выбирает тип данных из раскрывающегося списка, как показано на рисунке 1.6. Нижняя часть бланка содержит список свойств поля, выделенного в верхней части. Некоторые свойства заданы по умолчанию. При настройке свойств каждого поля можно установить маску для ввода поля, границы изменения значений поля, значение по умолчанию, сообщение при вводе ошибочных данных.

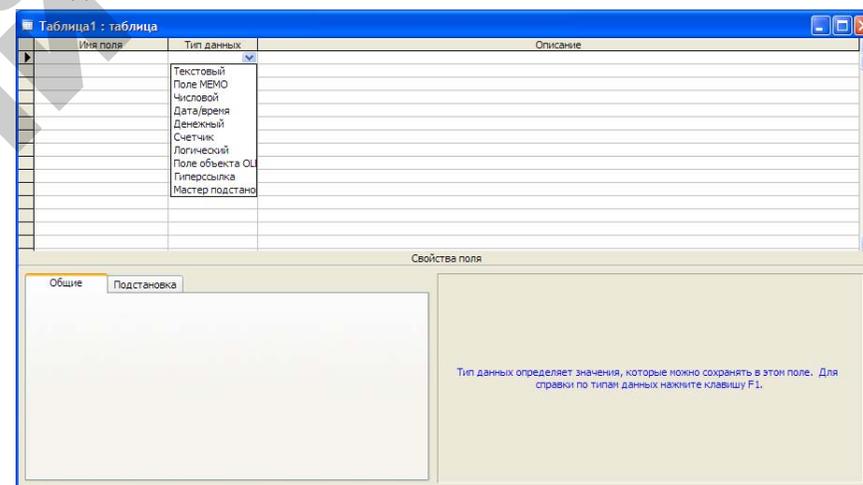


Рисунок 1.6 – Выбор типов данных

## 2.2. Создание структуры таблиц

1. В окне базы данных откройте вкладку **Таблицы**.
2. Нажмите кнопку **Создать** и в диалогом окне **Новая таблица** выберите опцию **Конструктор**. В результате откроется бланк создания структуры таблицы (рисунок 1.5).
3. Создайте структуру таблицы **Дисциплины**. Внесите в окно конструктора таблиц названия полей, типы данных и свойства в соот-

ветствии с таблицей 1.1. Свойства полей, не указанные в таблице, оставьте без изменения.

Таблица 1.1 – Дисциплины

Имя поля	Тип данных	Размер поля	Значение по умолчанию
<i>КодДисциплины</i>	Числовой	Длинное целое	0
<i>НазваниеДисциплины</i>	Текстовый	50	

4. В таблице **Дисциплины** определите ключевое поле (первичный ключ). Очевидно, что таким полем является поле **КодДисциплины**. Выделите поле **КодДисциплины**, щелчком правой кнопки мыши откройте контекстное меню и выберите в нем пункт **Ключевое поле**.

5. Сохраните созданную таблицу. Для этого нажмите кнопку  на Панели инструментов и дайте таблице имя **Дисциплины**. Вид таблицы **Дисциплины** в режиме конструктора будет иметь вид, представленный на рисунке 1.7.

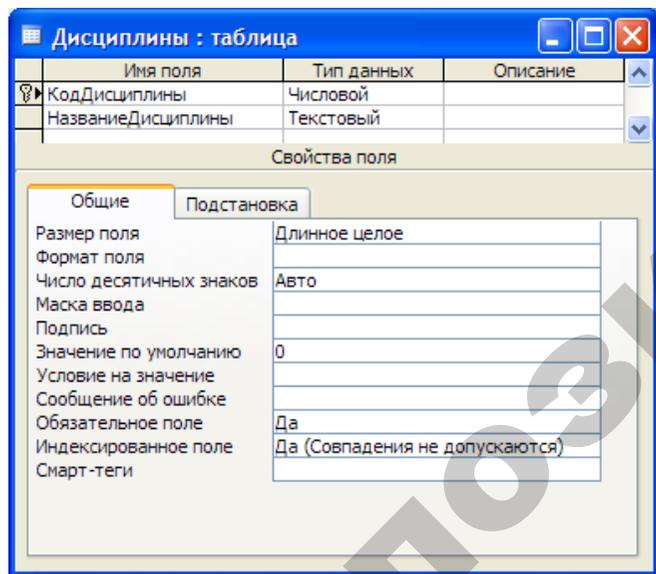


Рисунок 1.7 – Структура таблицы **Дисциплины**

6. Аналогичным образом создайте таблицы **ЛичныеДанные**, **Успеваемость** и **Стипендия**. Структура создаваемых таблиц указана в таблицах 1.2-1.4 соответственно. Свойства полей, не указанные в таблицах, оставьте без изменения.

Таблица 1.2 – ЛичныеДанные

Имя поля	Тип данных	Размер поля (Формат поля)	Ключевое поле
<i>КодСтудента</i>	Числовой	Длинное целое	*
<i>Фамилия</i>	Текстовый	20	
<i>Имя</i>	Текстовый	20	
<i>Отчество</i>	Текстовый	20	
<i>ДатаРождения</i>	Дата/время	Краткий формат даты	
<i>МестоРождения</i>	Текстовый	50	
<i>Рост</i>	Числовой	Длинное целое	
<i>Вес</i>	Числовой	Длинное целое	
<i>Телефон</i>	Текстовый	20	
<i>Общежитие</i>	Логический	Да/Нет	

Таблица 1.3 - Успеваемость

Имя поля	Тип данных	Обязательное поле	Условие на значение	Сообщение об ошибке
<i>№пп</i>	Счетчик			
<i>КодДисциплины</i>	Числовой	Да		
<i>КодСтудента</i>	Числовой	Да		
<i>Оценка</i>	Числовой	Да	>=1 And <=10	Введенное значение – не оценка

При вводе в поле **Оценка** значения, которое не удовлетворяет **Условию на значение**, на экран будет выведено сообщение об ошибке «Введенное значение – не оценка»

Таблица 1.4 - Стипендия

Имя поля	Тип данных	Размер поля	Формат поля
<i>СрБаллМин</i>	Числовой	Действительное	Основной
<i>СрБаллМакс</i>	Числовой	Действительное	Основной
<i>Вид</i>	Текстовый	20	
<i>Размер</i>	Числовой	Длинное целое	Денежный (число десятичных знаков - 0)

### 2.3. Создание связей между таблицами

Перед созданием связей необходимо предварительно закрыть все открытые таблицы. Не допускается создание или удаление связей между открытыми таблицами. Для установления связей между таблицами используется окно **Схема данных**.

1. В меню **Сервис** выберите пункт **Схема данных...**, либо в панели инструментов нажмите пиктограмму . Одновременно с открытием окна **Схема данных** открывается диалоговое окно **Добавление таблицы** (рисунок 1.8)
2. В открывшемся окне **Добавление таблицы** выделите название таблицы и нажмите кнопку **Добавить**. Повторите это для всех созданных таблиц и закройте окно **Добавление таблицы**.

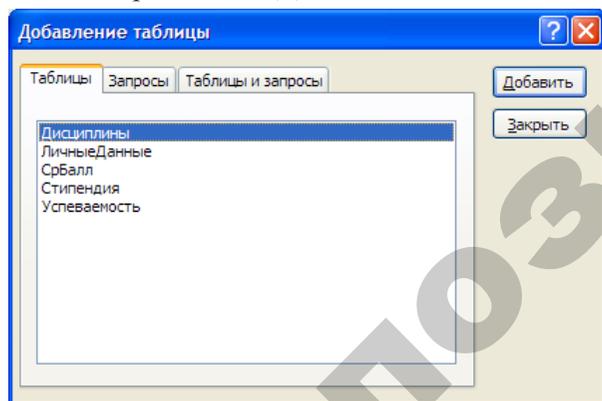


Рисунок 1.8 – Диалоговое окно **Добавление таблицы**

3. Откорректируйте размер и положение таблиц в окне **Схема данных** так, как показано на рисунке 1.9. Обратите внимание на то, что ключевые поля **КодДисциплины** и **КодСтудента**, обозначающие поле **первичного ключа**, в таблицах выделены **жирным шрифтом**.

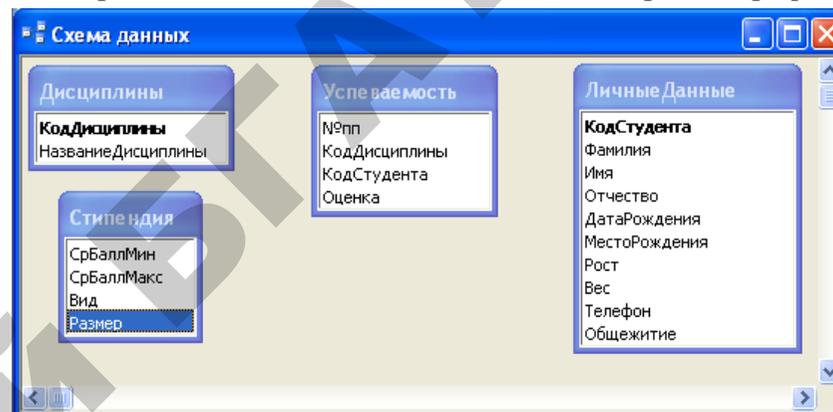


Рисунок 1.9 - Окно **Схема данных** с таблицами СУБД Деканат

4. Установим связь между таблицами **Дисциплины** и **Успеваемость**. Для этого нужно выбрать ключевое поле **КодДисциплины** в таблице **Дисциплины** и, не отпуская левую клавишу мыши, переместить его на поле **КодДисциплины** таблицы **Успеваемость**. На экране откроется окно **Изменение связей**. В этом окне необходимо проверить правильность имен связываемых полей, включить флажки **Обеспечение целостности данных**, **каскадное обновление связанных полей** и **каскадное удаление связанных записей**.

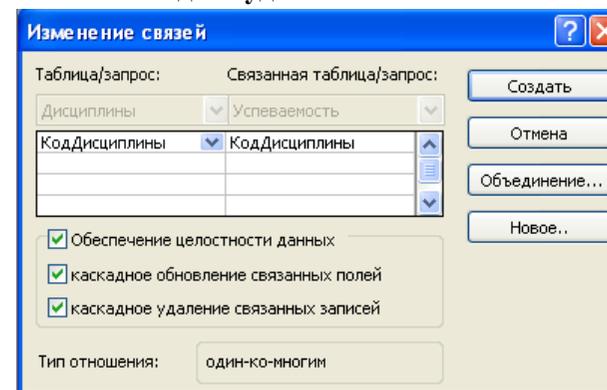


Рисунок 1.10 - Окно **Изменение связей**

При установлении связи между таблицами **Дисциплины** и **Успеваемость** окно **Изменение связей** должно выглядеть так, как показано на рисунке 1.10. Убедитесь в том, что между таблицами установлен тип отношения **один-ко-многим** и нажмите кнопку **Создать**.

5. Пользуясь этой методикой, установите связь между таблицами **Личные Данные** и **Успеваемость**. Окончательный вид окна **Схема данных** показан на рисунке 1.11.

При создании связи в окне **Схема данных** может возникнуть необходимость в изменении структуры таблицы, не покидая окна. Для этого необходимо установить указатель мыши на модифицируемую таблицу и нажать правую кнопку мыши. Затем из контекстного меню выбрать команду **Конструктор таблиц**. Для удаления связи необходимо установить указатель мыши на линию связи, выделить ее левой кнопкой мыши и нажать клавишу **Delete**.

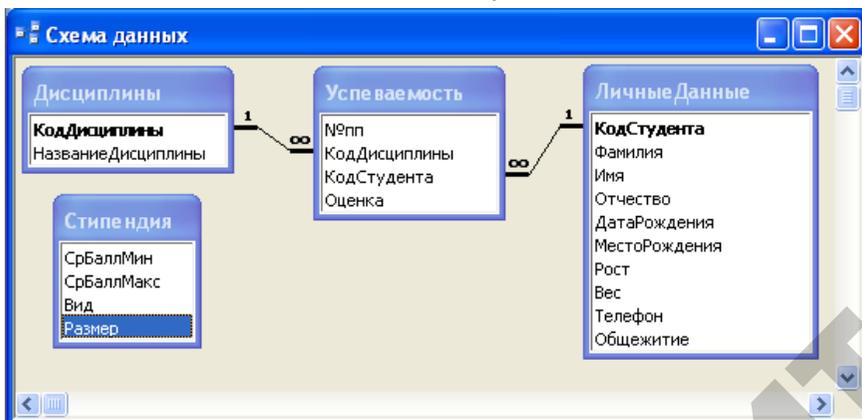


Рисунок 1.11 – Окончательный вид окна **Схема данных**

### 3. Вопросы для самоконтроля

1. Перечислите основные объекты БД.
2. Перечислите основные свойства полей БД.
3. Какие типы данных используются в БД?
4. Для чего используется ключевое поле?
5. В чем отличие первичного ключа от вторичного?
6. Какие типы отношений реализуют связи между таблицами?
7. Как изменить структуру таблицы в окне **Схема данных**?
8. Как изменить существующую связь?

### 4. Варианты индивидуальных заданий

Создать структуры таблиц (типы полей и ключевые поля задать самостоятельно). Установить связи между таблицами.

#### Вариант 1. БД Факультет

##### 1. Таблица 1 – **Сотрудники**.

Таблица должна иметь следующие поля:

- порядковый номер;
- ФИО сотрудника;
- дата рождения;
- ученая степень;
- стаж работы;
- код кафедры;
- номер телефона.

##### 2. Таблица 2 – **Кафедры**.

Таблица должна иметь следующие поля:

- код кафедры;
- наименование кафедры.

#### Вариант 2. БД Программы

##### 1. Таблица 1 – **Программные продукты**.

Таблица должна иметь следующие поля:

- порядковый номер;
- шифр программного продукта;
- код фирмы-производителя программного продукта;
- наименование продукта;
- тип программного продукта (*например, операционные системы, антивирусные программы, графические редакторы, СУБД и т.д.*);
- цена одного экземпляра;
- количество экземпляров.

##### 2. Таблица 2 – **Фирмы**.

Таблица должна иметь следующие поля:

- код фирмы-производителя программного продукта;
- название фирмы (*например, Microsoft, Borland,, Лаборатория Касперского*).

#### Вариант 3. БД Склад

##### 1. Таблица 1 – **Склад**.

Таблица должна иметь следующие поля:

- порядковый номер товара;
- код товара;
- цена товара;
- код фирмы-поставщика;
- количество экземпляров товара.

2. Таблица 2 – **Товары.**

Таблица должна иметь следующие поля:

- код товара;
- название товара;
- производитель товара.

3. Таблица 3 – **Фирмы.**

Таблица должна иметь следующие поля:

- код фирмы-поставщика;
- название фирмы-поставщика.

**Вариант 4. БД Платежи.**

1. Таблица 1 – **Платежи.**

Таблица должна иметь следующие поля:

- порядковый номер платежа;
- дата платежа;
- номер счета;
- код банка;
- сумма платежа;
- пеня.

2. Таблица 2 – **Счета.**

Таблица должна иметь следующие поля:

- номер счета;
- название организации.

3. Таблица 3 – **Банки.**

Таблица должна иметь следующие поля:

- код банка;
- название банка.

**Вариант 5. БД Комплектующие к ПК**

1. Таблица 1 – **Комплектующие**

Таблица должна иметь следующие поля:

- порядковый номер;
- код изделия;
- цена;
- количество.

2. Таблица 2 – **Изделия.**

Таблица должна иметь следующие поля:

- код изделия;
- наименование изделия;
- код фирмы-изготовителя.

3. Таблица 3 – **Фирмы.**

Таблица должна иметь следующие поля:

- код фирмы;
- наименование фирмы.

**Вариант 6. БД Музыка**

1. Таблица 1 – **Диски**

Таблица должна иметь следующие поля:

- порядковый номер компакт-диска;
- название компакт-диска;
- цена.

2. Таблица 2 – **Содержание**

Таблица должна иметь следующие поля:

- порядковый номер компакт-диска;
- название произведения;
- код исполнителя;
- длительность воспроизведения в минутах.

3. Таблица 3 – **Исполнители.**

Таблица должна иметь следующие поля:

- код исполнителя;
- фамилия исполнителя.

**Вариант 7. БД Абитуриент.**

1. Таблица 1 – **Результаты.**

Таблица должна иметь следующие поля:

- порядковый номер;
- шифр абитуриента;
- код предмета;
- полученная оценка.

2. Таблица 2 – **Абитуриенты**

Таблица должна иметь следующие поля:

- шифр абитуриента;
- фамилия, имя, отчество абитуриента;
- год рождения;
- средний балл аттестата.

3. Таблица 3 – **Предметы**

Таблица должна иметь следующие поля:

- код предмета;
- наименование предмета.

**Вариант 8. БД Зарплата**

1. Таблица 1 – **Отдел кадров**

Таблица должна иметь следующие поля:

- ФИО сотрудника;
- домашний адрес;
- телефон;
- паспортные данные.

2. Таблица 2 – **Ведомость.**

Таблица должна иметь следующие поля:

- название цеха;
- ФИО сотрудника;
- больничные дни (значение по умолчанию – 0);
- начисляемый заработок;
- удержано;
- дата выдачи.

**Вариант 9. БД Телефон.**

1. Таблица 1 – **Разговоры.**

Таблица должна иметь следующие поля:

- порядковый номер разговора;
- номер телефона;
- дата разговора;
- время начала разговора;
- время окончания разговора;
- код города вызываемого абонента.

2. Таблица 2 – **Телефоны.**

Таблица должна иметь следующие поля:

- номер телефона;
- фамилия владельца;
- адрес владельца.

3. Таблица 3 – **Города.**

Таблица должна иметь следующие поля:

- код города;
- название города.

**Вариант 10. БД Отдел.**

1. Таблица 1 – **Сотрудники.**

Таблица должна иметь следующие поля:

- порядковый номер;
- фамилия сотрудника;
- дата рождения;
- код области рождения;
- месячный оклад;
- премия в процентах.

2. Таблица 2 – **Области.**

Таблица должна иметь следующие поля:

- код области;
- наименование области;
- населенный пункт.

**Вариант 11. БД Квартира**

1. Таблица 1 – **Квартиры**

Таблица должна иметь следующие поля:

- порядковый номер;
- дата выставления на продажу;
- код района;
- количество комнат;
- общая площадь в кв. метрах;
- наличие телефона (значение по умолчанию «да»);
- оценочная стоимость квартиры.

2. Таблица 2 – **Районы**

Таблица должна иметь следующие поля:

- код района;
- название района.

**Вариант 12. БД Справочник врача.**

1. Таблица 1 – **Даты.**

Таблица должна иметь следующие поля:

- порядковый номер пациента;
- код пациента;
- дата обращения к врачу.

2. Таблица 2 – **Пациенты.**

Таблица должна иметь следующие поля:

- код пациента;
- фамилия и инициалы;

- год рождения;
- домашний адрес;
- код диагноза.

### 3. Таблица 3 – **Диагнозы.**

Таблица должна иметь следующие поля:

- код диагноза;
- диагноз.

### **Вариант 13. БД Почта.**

#### 1. Таблица 1 – **Отправления.**

Таблица должна иметь следующие поля:

- порядковый номер;
- дата отправки;
- код города назначения;
- адрес получателя;
- фамилия получателя;
- адрес отправителя;
- фамилия отправителя;
- вес почтового отправления в граммах.

#### 2. Таблица 2 – **Города.**

Таблица должна иметь следующие поля:

- код города;
- название города.

### **Вариант 14. БД Автомобили.**

#### 1. Таблица 1 – **Цены.**

Таблица должна иметь следующие поля:

- порядковый номер;
- код марки автомобиля;
- год выпуска;
- пробег в километрах;
- стартовая цена.

#### 2. Таблица 2 – **Марки.**

Таблица должна иметь следующие поля:

- код марки автомобиля;
- марка автомобиля;
- код страны изготовителя.

#### 3. Таблица 3 – **Страны.**

Таблица должна иметь следующие поля:

- код страны;
- название страны.

### **Вариант 15. БД Учет компьютерных средств.**

#### 1. Таблица 1 – **Компьютеры.**

Таблица должна иметь следующие поля:

- код подразделения университета;
- инвентарный № компьютера;
- наименование компьютера;
- место расположения (№ ауд.);
- год ввода в эксплуатацию;
- стоимость.

#### 2. Таблица 2 – **Подразделения.**

Таблица должна иметь следующие поля:

- код подразделения университета;
- наименование подразделения.

### **Вариант 16. БД Успеваемость.**

#### 1. Таблица 1 – **Оценки.**

Таблица должна иметь следующие поля:

- код предмета;
- код студента;
- оценка;
- дата сдачи экзамена.

#### 2. Таблица 2 – **Студенты.**

Таблица должна иметь следующие поля:

- код студента;
- Ф.И.О.;
- дата рождения;
- год поступления;
- группа;
- стипендия (по умолчанию «да»).

#### 3. Таблица 3 – **Предметы.**

Таблица должна иметь следующие поля:

- код предмета;
- наименование предмета.

### **Вариант 17. БД Заказы.**

#### 1. Таблица 1 – **Работы.**

Таблица должна иметь следующие поля:

- № заказа;
- код клиента;
- код сотрудника;

- дата размещения заказа;
- срок выполнения заказа (*по умолчанию – 14 дней*);
- дата завершения работ (*если заказ не выполнен – в поле пусто*);
- стоимость работ;
- описание работы.

2. Таблица 2 – **Клиенты.**

Таблица должна иметь следующие поля:

- код клиента;
- название фирмы;
- форма оплаты (*по умолчанию - безналичная*);
- номер телефона.

3. Таблица 3 – **Сотрудники.**

Таблица должна иметь следующие поля:

- код сотрудника;
- Ф.И.О. сотрудника;
- вознаграждение (*% от стоимости работ*);
- контактный телефон.

**Вариант 18. БД Реализация готовой продукции**

1. Таблица 1 – **Товары.**

Таблица должна иметь следующие поля:

- код товара;
- наименование;
- оптовая цена;
- описание.

2. Таблица 2 – **Покупатели.**

Таблица должна иметь следующие поля:

- код покупателя;
- наименование покупателя;
- телефон;
- район;
- адрес.

3. Таблица 3 – **Сделки.**

Таблица должна иметь следующие поля:

- код сделки;
- дата сделки;
- код товара;
- количество;
- код покупателя.

**Вариант 19. БД Транспорт.**

1. Таблица 1 – **Перевозки.**

Таблица должна иметь следующие поля:

- номер рейса;
- дата рейса;
- код водителя;
- код автомобиля;
- вес груза в тоннах;
- пробег в километрах.

2. Таблица 2 – **Водители.**

Таблица должна иметь следующие поля:

- код водителя;
- фамилия водителя.

3. Таблица 3 – **Автомобили.**

Таблица должна иметь следующие поля:

- код автомобиля;
- номер автомобиля;
- марка автомобиля.

**Вариант 20. БД Расписание.**

1. Таблица 1 – **Ведомость.**

Таблица должна иметь следующие поля:

- № поезда;
- категория;
- Ф.И.О. начальника поезда;
- количество вагонов в составе;
- наличие вагона-ресторана.

2. Таблица 2 – **Маршрутный лист.**

Таблица должна иметь следующие поля:

- № поезда;
- пункт отправления (*значение по умолчанию «Минск»*);
- пункт назначения;
- время отправления;
- время прибытия;

**Вариант 21. БД Соревнования.**

1. Таблица 1 – **Протокол.**

Таблица должна иметь следующие поля:

- место проведения (*значение по умолчанию – стадион «Динамо»*);
- дата проведения;

- Ф.И.О. участника;
- вид спорта;
- результат;
- занятое место.

2. Таблица 2 – **Карточка участника.**

Таблица должна иметь следующие поля:

- Ф.И.О. участника;
- дата рождения;
- спортивный разряд;
- рост;
- вес.

**Вариант 22. БД Праздничное бюро.**

1. Таблица 1 – **Мероприятия.**

Таблица должна иметь следующие поля:

- код мероприятия;
- код сотрудника;
- место проведения (*по умолчанию – ресторан «Журавинка»*);
- дата проведения;
- время начала;
- время окончания;
- количество мест.

2. Таблица 2 – **Сотрудники.**

Таблица должна иметь следующие поля:

- код сотрудника;
- Ф.И.О.;
- комиссионные (%).

3. Таблица 3 – **Типы мероприятий.**

Таблица должна иметь следующие поля:

- код мероприятия;
- название мероприятия;
- цена за 1 место.

**Вариант 23. БД Составление учебного плана.**

1. Таблица 1 – **Преподаватели.**

Таблица должна иметь следующие поля:

- код преподавателя;
- Фамилия, Имя, Отчество;
- ученая степень.

2. Таблица 2 – **Предметы.**

Таблица должна иметь следующие поля:

- код предмета;
- название;
- количество часов;
- итоговый контроль (*по умолчанию «зачет»*).

3. Таблица 3 – **Группа.**

Таблица должна иметь следующие поля:

- код преподавателя;
- код предмета;
- номер группы.

**Вариант 24. БД Туристическая фирма.**

1. Таблица 1 – **Предложения.**

Таблица должна иметь следующие поля:

- код путевки;
- страна;
- климат;
- продолжительность;
- цена;
- категория отеля (*по умолчанию \*\*\**);
- название отеля.

2. Таблица 2 – **Путевки.**

Таблица должна иметь следующие поля:

- код путевки;
- код клиента;
- дата отправления
- скидка (*по умолчанию – 0 %*).

3. Таблица 2 – **Клиенты.**

Таблица должна иметь следующие поля:

- код клиента;
- Фамилия, Имя, Отчество;
- адрес (город);
- телефон.

**Вариант 25. БД Грузовые перевозки.**

1. Таблица 1 – **Маршруты.**

Таблица должна иметь следующие поля:

- код маршрута;
- пункт назначения;
- дальность (*км*);

- оплата.
2. Таблица 2 – **Водители**.  
Таблица должна иметь следующие поля:
    - код водителя;
    - фамилия, имя, отчество;
    - дата рождения;
    - стаж.
  3. Таблица 3 – **Маршрутный лист**.  
Таблица должна иметь следующие поля:
    - код маршрута;
    - код водителя;
    - дата отправки;
    - дата возвращения;
    - премия в %

#### **Вариант 26. БД Прокат автомобилей.**

1. Таблица 1 – **Список автомобилей**.  
Таблица должна иметь следующие поля:
  - код автомобиля;
  - марка;
  - стоимость автомобиля;
  - стоимость проката (за 1 день);
  - год выпуска.
2. Таблица 2 – **Клиенты**.  
Таблица должна иметь следующие поля:
  - код клиента;
  - фамилия;
  - имя;
  - отчество;
  - адрес;
  - телефон.
3. Таблица 2 – **Выданные автомобили** (в текущем году).  
Таблица должна иметь следующие поля:
  - код автомобиля;
  - код клиента;
  - дата выдачи;
  - дата возврата.

## **Лабораторная работа №2**

### **СОЗДАНИЕ ПРОСТЫХ ФОРМ И ЗАПОЛНЕНИЕ ТАБЛИЦ**

**Цель лабораторной работы:** приобрести навыки создания простых форм и заполнения таблиц БД.

#### **Порядок выполнения лабораторной работы**

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания простых форм и заполнения таблиц БД **Деканат**. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.

#### **1. Краткие теоретические сведения**

Для ввода данных в базу данных удобно использовать формы. Данные в таблицы можно заносить и без помощи форм, но существуют, по крайней мере, четыре причины, которые делают формы незаменимым средством ввода данных в БД.

1. Пользователям СУБД нельзя предоставлять доступ к таблицам (самому ценному из того, что есть в БД).
2. Разные пользователи могут иметь разные права доступа к информации, содержащейся в таблицах. Формы предоставляют пользователю средства для заполнения только тех полей, которые ему заполнять положено. Например, один сотрудник банка имеет право вводить только имена и адреса клиентов, другой – только номера их расчетных счетов, а третий – только денежные суммы, хранящиеся на этих счетах. Для ввода данных им предоставляют разные формы, хотя данные из этих форм могут поступать в одну таблицу.
3. Ввод данных в таблицу – чрезвычайно утомительное занятие. Уже после нескольких часов работы люди делают ошибки. Ввод данных в форму проще, т.к. в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и прочие) для автоматизации ввода. К тому же элементы управления в формах настраивают таким образом, чтобы при вводе данных выполнялась их первичная проверка.
4. Преимущества форм раскрываются особенно наглядно, когда информация для БД вводится вручную из заполненных бланков (анкеты, накладные, счета, описи, ведомости и т. п.). В этом случае форму делают графическими средствами так, чтобы она была точной копией бланка – это заметно упрощает работу персонала по вводу данных, снижает его утомление и предотвращает появление ошибок.

## 2. Пример создания простых форм и заполнения таблиц БД Деканат

Формы, предназначенные для ввода данных в таблицы, удобно создавать с помощью средств автоматизации, называемые **автоформы**.

Существуют три вида автоформ:

- *в столбец* – отображает все поля одной записи, удобна для ввода и редактирования данных;
- *ленточная* – отображает группу записей, удобна для вывода данных;
- *табличная* – по внешнему виду ничем не отличается от таблицы, на которой она основана.

### 2.1. Создание Автоформы вида Ленточная

Для ввода данных в таблицу *Дисциплины* создать автоформу вида *Ленточная*.

1. В окне базы данных *Деканат* откройте вкладку **Формы**.
2. Нажмите кнопку **Создать**, в результате откроется диалоговое окно **Новая форма**.
3. В диалоговом окне **Новая форма** выберите опцию **Автоформа: ленточная**, в раскрывающемся списке укажите в качестве источника данных таблицу *Дисциплины*, как показано на рисунке 2.1.

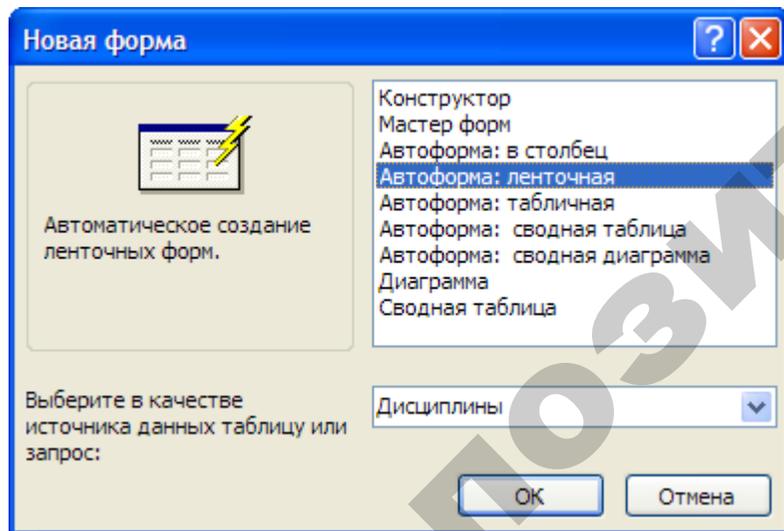


Рисунок 2.1 – Окно **Новая форма**

4. Названия некоторых дисциплин могут превышать размер поля, который по умолчанию устанавливает *Access*. Поэтому, прежде чем заносить данные в поле, необходимо откорректировать его размер. Для этого перейдите в *Конструктор* (**Вид→Конструктор**), выделите поле **НазваниеДисциплины** и, наведя курсор мыши на маркеры, установите необходимый размер поля, как показано на рисунке 2.2.

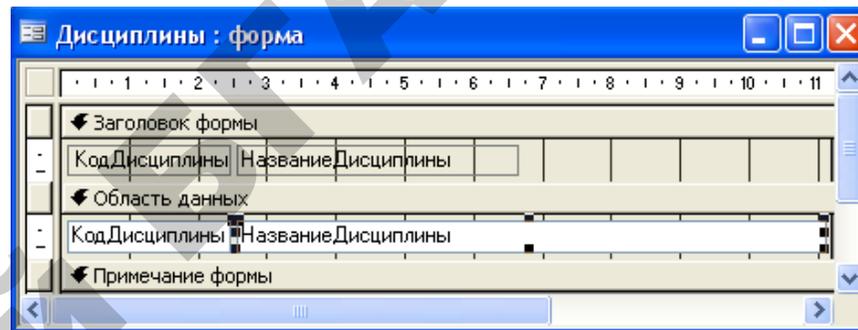


Рисунок 2.2 – Форма *Дисциплины* в режиме конструктора

5. После этого перейдите в режим формы (**Вид→Режим формы**) и занесите данные в форму *Дисциплины*, как показано на рисунке 2.3. В окне БД *Деканат* нажмите кнопку **Таблицы** и убедитесь, что в таблицу *Дисциплины* данные занесены правильно.

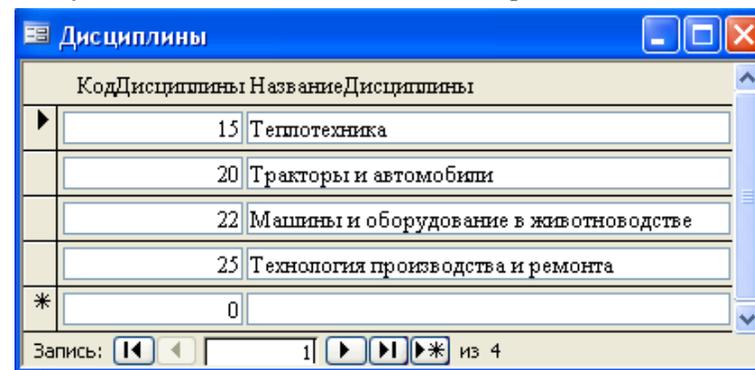


Рисунок 2.3 – Форма *Дисциплины*

### 2.2. Создание Автоформы: в столбец

Для заполнения таблицы *ЛичныеДанные*, в которой могут содержаться конфиденциальные сведения, создайте две формы вида *Автоформа: в столбец* - *ЛичныеДанные1* и *ЛичныеДанные2*, по-

казанные на рисунках 2.4 и 2.5, соответственно. Для создания этих форм используйте методику создания формы **Дисциплины**. С помощью созданных форм занесите информацию так, чтобы таблица **ЛичныеДанные** имела вид, показанный на рисунке 2.6.

Личные Данные

КодСтудента: 74060301  
 ДатаРождения: 01.01.1988  
 МестоРождения: Брест  
 Рост: 171  
 Вес: 71  
 Телефон: 2010101

Запись: 1 из 10

Рисунок 2.4 – Форма ЛичныеДанные1

Личные Данные

КодСтудента: 74060301  
 Фамилия: Первый  
 Имя: Петр  
 Отчество: Петрович  
 Общежитие:

Запись: 1 из 10

Рисунок 2.5 – Форма ЛичныеДанные2

	КодСтудента	Фамилия	Имя	Отчество	ДатаРождения	МестоРождения	Рост	Вес	Телефон	Общежитие
+	74060301	Первый	Петр	Петрович	01.01.1988	Брест	171	71	2010101	<input type="checkbox"/>
+	74060302	Второй	Виктор	Викторович	02.02.1988	Минск	172	72	2020202	<input type="checkbox"/>
+	74060303	Третий	Тимофей	Тимофеевич	03.03.1988	Гомель	173	73	2030303	<input type="checkbox"/>
+	74060304	Четвертый	Чеслав	Чеславович	04.04.1988	Гродно	174	74	2040404	<input checked="" type="checkbox"/>
+	74060305	Пятый	Павел	Павлович	05.05.1988	Витебск	175	75	2050505	<input type="checkbox"/>
+	74060306	Шестой	Шалва	Шалвович	06.06.1989	Витебск	176	76	2060606	<input checked="" type="checkbox"/>
+	74060307	Седьмой	Сергей	Сергеевич	07.07.1989	Могилев	177	77	2070707	<input type="checkbox"/>
+	74060308	Восьмой	Владимир	Владимирович	08.08.1989	Гродно	178	78	2080808	<input checked="" type="checkbox"/>
+	74060309	Девятый	Денис	Денисович	09.09.1989	Гомель	179	79	2090909	<input type="checkbox"/>
+	74060310	Десятый	Дмитрий	Дмитриевич	10.10.1989	Брест	180	80	2101010	<input checked="" type="checkbox"/>
							0	0		<input type="checkbox"/>

Запись: 11 из 11

Рисунок 2.6 – Таблица ЛичныеДанные

### 2.3. Создание Автоформы вида табличная

Создайте форму **Стипендия** вида *Автоформа: табличная*, используя описанную выше методику. С помощью созданной формы занесите информацию, чтобы таблица **Стипендия** имела вид, как показано на рисунке 2.7.

	СрБаллМин	СрБаллМакс	Вид	Размер
▶	1	5	Социальная	78574
	5	6	Учебная	93006
	6	8	Учебная	111607
	8	9	Учебная	130208
	9	10	Учебная	148810
*	0	0		0

Запись: 1 из 5

Рисунок 2.7 – Таблица Стипендия

### 2.4. Добавление в форму элемента управления «Поле со списком»

1. Создайте форму **Успеваемость** вида *Автоформа: в столбец* для таблицы **Успеваемость**.

2. В поля *КодДисциплины* и *КодСтудента* формы **Успеваемость** данные вносятся из соответствующих полей таблиц *Дисциплины* и *ЛичныеДанные*, что делает заполнение формы неудобным для пользователя. Оптимальное решение – использовать для ввода данных элемент управления, называемый *поле со списком*. Для преобразования обычных полей в «поля со списком» выполните следующие действия:

а) откройте форму **Успеваемость** в режиме *Конструктор (Вид→Конструктор)*;

б) щелкните правой кнопкой мыши по полю **КодДисциплины** и в появившемся контекстном меню выберите опцию **Преобразовать элемент в... Поле со списком**, как показано на рисунке 2.8;

в) выделите поле **КодДисциплины** и выполните команду **Вид → Свойства**;

г) в появившемся окне **Поле со списком: КодДисциплины** перейдите на вкладку **Все** и установите в полях следующие значения (поля, не указанные ниже, оставьте без изменения):

Тип источника строк – Таблица или запрос  
 Источник строк – Дисциплины  
 Число столбцов – 2  
 Ширина столбцов – 2см;3см  
 Присоединенный столбец – 1  
 Ширина списка – 5см

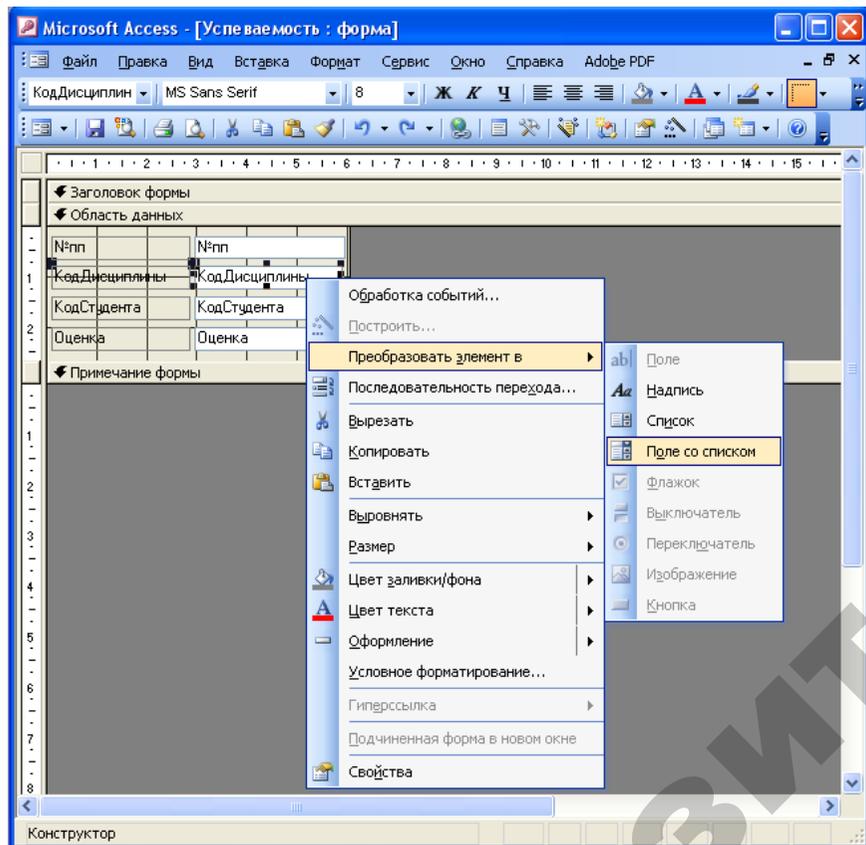


Рисунок 2.8 – Преобразование элемента в поле со списком

д) самостоятельно преобразуйте поле **КодСтудента** в элемент *поле со списком*; при редактировании свойств этого поля укажите в качестве источника строк таблицу **ЛичныеДанные**.

е) внешний вид формы **Успеваемость** представлен на рисунке 2.9.

3. Сохраните форму **Успеваемость** и перейдите в режим формы (**Вид→Режим формы**).

4. Заполните форму **Успеваемость**, используя данные таблиц, представленных на рисунках 2.10 – 2.11. Убедитесь, что данные в таблицу **Успеваемость** занесены правильно.

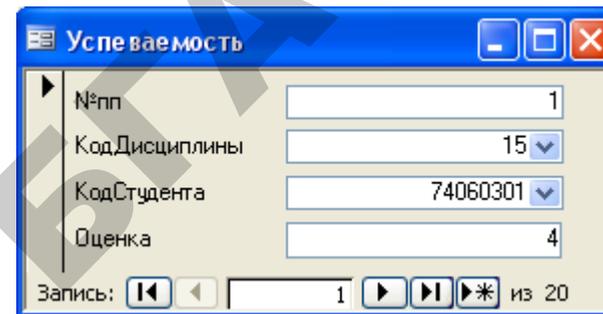


Рисунок 2.9 – Форма **Успеваемость**

№пп	КодДисциплины	КодСтудента	Оценка
1	15	74060301	4
2	20	74060301	4
3	22	74060301	4
4	25	74060301	4
5	15	74060302	4
6	20	74060302	4
7	22	74060302	5
8	25	74060302	5
9	15	74060303	5
10	20	74060303	5
11	22	74060303	5
12	25	74060303	5
13	15	74060304	5
14	20	74060304	5
15	22	74060304	6
16	25	74060304	6
17	15	74060305	6
18	20	74060305	6
19	22	74060305	6
20	25	74060305	6
21	15	74060306	7

Рисунок 2.10 – Таблица **Успеваемость** (начало таблицы)

№пп	КодДисциплины	КодСтудента	Оценка
21	15	74060306	7
22	20	74060306	7
23	22	74060306	7
24	25	74060306	7
25	15	74060307	8
26	20	74060307	8
27	22	74060307	8
28	25	74060307	8
29	15	74060308	8
30	20	74060308	8
31	22	74060308	9
32	25	74060308	9
33	15	74060309	9
34	20	74060309	9
35	22	74060309	9
36	25	74060309	9
37	15	74060310	9
38	20	74060310	9
39	22	74060310	10
40	25	74060310	10
* (етчик)	0	0	0

Запись: 14 из 40

Рисунок 2.10 – Таблица **Успеваемость** (конец таблицы)

### 3. Вопросы для самоконтроля

1. Для чего предназначены формы?
2. Почему пользователям СУБД нельзя предоставлять доступ к таблицам?
3. Когда для ввода данных в одну таблицу необходимо использовать несколько форм?
4. Каким образом можно автоматизировать ввод данных с помощью формы?
5. Почему при использовании форм упрощается работа персонала по вводу данных, снижается его утомление и предотвращается появление ошибок?

### 4. Варианты индивидуальных заданий

Создать формы для заполнения таблиц, структуры которых разработаны в соответствии с заданными вариантами индивидуальных заданий лабораторной работы №1. С помощью созданных форм заполнить таблицы модельными данными, значения которых подготовить самостоятельно.

## Лабораторная работа №3 СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОРКУ

**Цель лабораторной работы:** приобрести навыки создания запросов на выборку.

### Порядок выполнения лабораторной работы

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания запросов БД **Деканат**. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.

### 1. Краткие теоретические сведения

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. С помощью запросов можно выполнять преобразование данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое.

Все записи в основные таблицы вносятся только в *естественном* порядке по мере их поступления, то есть в неупорядоченном виде. Если же пользователю надо видеть данные, отсортированные или отфильтрованные по тому или иному принципу, он просто использует соответствующий запрос.

Особенность запросов состоит в том, что они получают данные из базовых таблиц и создают на их основе временную *результующую таблицу* в оперативной памяти ПК.

Запросы также можно использовать в качестве источников записей для форм, отчетов и страниц доступа к данным. В *Microsoft Access* можно создать несколько типов запросов.

**Запросы на выборку.** Запросы на выборку позволяют выбрать необходимые данные из одной или нескольких взаимосвязанных таблиц, произвести вычисления и получить результат в виде новой таблицы. Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

**Запросы с параметрами.** Запрос с параметрами — это запрос, который при выполнении отображает в собственном диалоговом

окне приглашение ввести данные, например, условие для возвращения записей или значение, которое требуется вставить в поле.

Запросы с параметрами также удобно использовать в качестве основы для форм, отчетов и страниц доступа к данным. Например, на основе запроса с параметрами можно создать месячный отчет о доходах. При печати данного отчета *Microsoft Access* выводит на экран приглашение ввести месяц, доходы за который должны быть приведены в отчете. После ввода месяца *Microsoft Access* выполняет печать соответствующего отчета.

**Перекрестные запросы.** Перекрестные запросы используют для расчетов и представления данных в структуре, облегчающей их анализ. Перекрестный запрос подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк.

## 2. Пример создания запросов БД Деканат

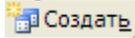
Для создания запросов в СУБД *Access* имеется удобное графическое средство формирования **запроса по образцу** – *QBE (Query By Example)*, с помощью которого легко можно построить сложный запрос. При конструировании запроса достаточно выделить и перетащить необходимые поля из таблиц в бланк запроса и ввести условия отбора записей.

Последовательное выполнение ряда запросов позволяет решать достаточно сложные задачи, не прибегая к программированию, что очень важно для большинства пользователей.

### 2.1. Создание запроса на выборку

*Создать Запрос1 на выборку следующей информации о студентах: Ф.И.О., дата рождения, рост и вес. Информацию вывести только о студентах, рост которых не менее 176 см, вес не превышает 79 кг, Ф.И.О. отсортировать в алфавитном порядке.*

Для создания запроса выполните следующие действия.

1. Перейдите к объектам **Запросы** в левой части окна БД Деканат и нажмите кнопку **Создать**  в окне БД.
2. В окне **Новый запрос** выберите **Конструктор** и нажмите кнопку **ОК**. В результате откроется диалоговое окно конструктора запросов и окно **Добавление таблицы**.

3. В диалоговое окно конструктора запроса добавьте таблицу **ЛичныеДанные**, в которой находятся запрашиваемые поля.

4. Поместите в бланк запроса следующие поля: **Фамилия**, **Имя**, **Отчество**, **ДатаРождения**, **Рост** и **Вес**. Размещение полей может выполняться либо перетаскиванием мышью необходимого поля, либо двойным щелчком на имени поля в таблице.

5. Для поля **Фамилия** установите сортировку по возрастанию (в поле *Сортировка*:).

6. В бланке запроса для поля **Рост** установите условие отбора  $\geq 176$ , а для поля **Вес**, соответственно,  $< 79$ . Окончательно сформированное окно конструктора запроса должно выглядеть так, как показано на рисунке 3.1.

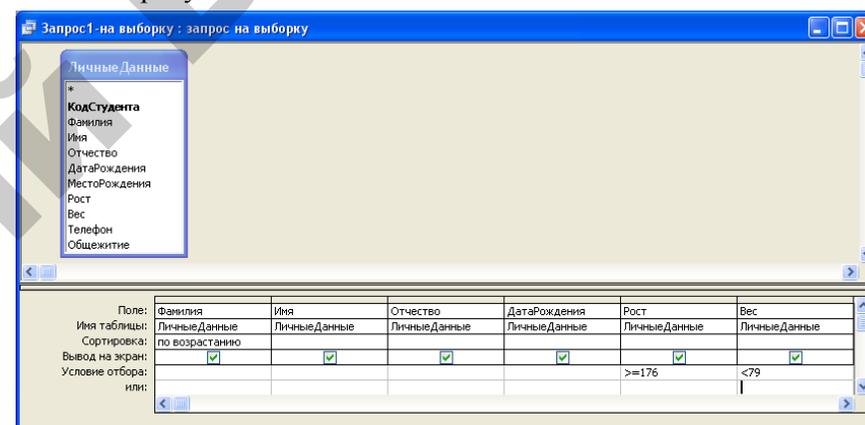


Рисунок 3.1 – Диалоговое окно конструктора запросов

7. Выполнить запрос нажатием клавиши меню  – **Запуск**. Результат выполнения запроса приведен на рисунке 3.2.

8. Сохраните запрос под именем **Запрос1** и убедитесь, что он появился в объектах **Запросы**.

	Фамилия	Имя	Отчество	ДатаРождения	Рост	Вес
	Восьмой	Владимир	Владимирович	08.08.1989	178	78
	Седьмой	Сергей	Сергеевич	07.07.1989	177	77
	Шестой	Шалва	Шалвович	06.06.1989	176	76
*					0	0

Рисунок 3.2 – Результат выполнения запроса на выборку

## 2.2. Создание запроса с параметром

Создать *Запрос2* на выборку следующей информации о студенте: фамилия, название дисциплины, оценка. Фамилия студента должна вводиться как параметр в диалоговом окне.

Для создания запроса выполните следующие действия.

1. Перейдите к объектам **Запросы** в левой части окна БД **Деканат** и нажмите кнопку **Создать** в окне БД.
2. В окне **Новый запрос** выберите **Конструктор** и нажмите кнопку **ОК**.
3. В диалоговое окно конструктора запроса добавьте таблицы **Дисциплины**, **Успеваемость** и **ЛичныеДанные**, в которых находятся запрашиваемые поля.
4. Поместите в бланк запроса следующие поля: **Фамилия**, **НазваниеДисциплины**, **Оценка**.
5. Для поля **Фамилия** в строке **Условие отбора**: наберите  
[Введите фамилию студента]

Текст необходимо заключить в квадратные скобки. Окно конструктора запроса должно выглядеть так, как показано на рисунке 3.3.

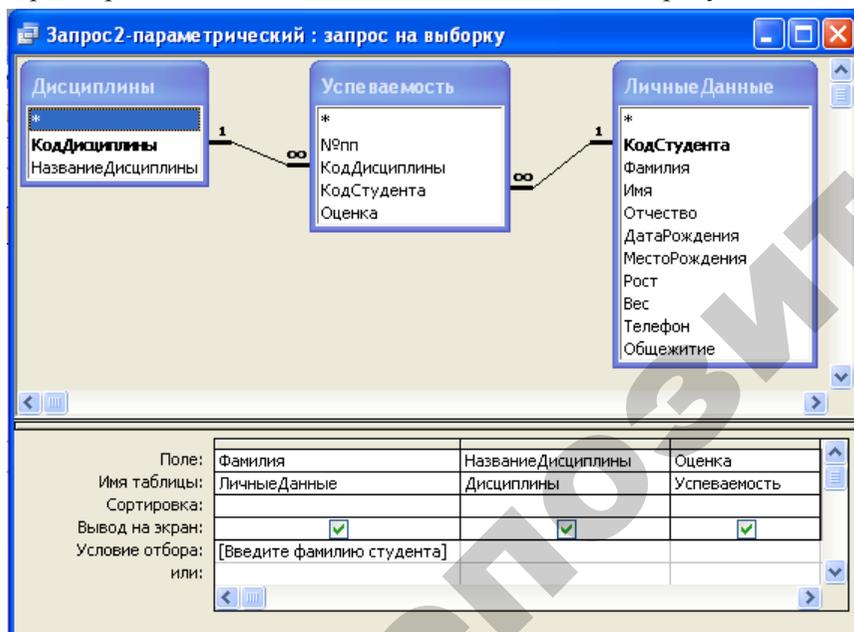


Рисунок 3.3 – Диалоговое окно конструктора запроса с параметром

6. Выполните запрос нажатием клавиши меню  – **Запуск**. На экране появится диалоговое окно **Введите значение параметра** (рисунок 3.4), которое содержит приглашение ввести условие отбора для поля **Фамилия**.

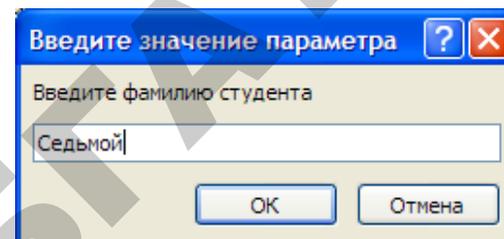


Рисунок 3.4 – Диалоговое окно для ввода параметра запроса

7. В диалоговом окне в качестве параметра введите фамилию студента **Седьмой** и нажмите кнопку **ОК**. Результат выполнения запроса приведен на рисунке 3.5.
8. Сохраните запрос под именем **Запрос2** и убедитесь, что он появился в объектах **Запросы**.
9. Выполните несколько раз **Запрос2**, используя в качестве параметра фамилии других студентов.

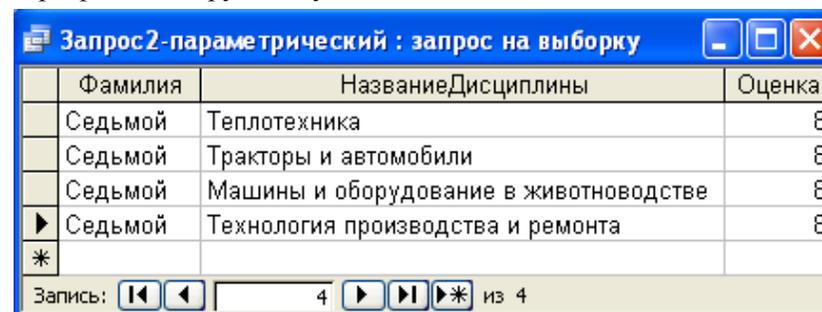


Рисунок 3.5 – Результат выполнения запроса с параметром

## 2.3. Создание запроса с вычисляемым полем

Создать *Запрос3* на выборку следующей информации о студенте: *Фамилия*, *место рождения*, *дата рождения*, *возраст*.

В БД отсутствует поле **Возраст**. Поле **Возраст** является вычисляемым полем, т.е. его значение можно вычислить по значениям других полей, имеющимся в БД. Так, это поле будет вычисляться по формуле:

$$\text{Возраст} = \text{Текущий год} - \text{Год рождения студента}$$

Для создания запроса выполните следующие действия:

1. Перейдите к объектам **Запросы** в левой части окна БД **Деканат** и нажмите кнопку **Создать** в окне БД. В окне **Новый запрос** выберите **Конструктор** и нажмите кнопку **ОК**.
2. В диалоговое окно конструктора запроса добавьте таблицу **ЛичныеДанные**. Поместите в бланк запроса поля: **Фамилия**, **МестоРождения**, **ДатаРождения**.
3. Сформируйте в пустой ячейке бланка запроса вычисляемое поле **Возраст**: (двоеточие после названия поля обязательно!). Для упрощения формирования подобных выражений лучше воспользоваться построителем выражений, который вызывается нажатием кнопки меню . Диалоговое окно **Построитель выражений** представлено на рисунке 3.6. Пользуясь панелями диалогового окна и кнопкой **Вставить**, выделяя необходимые поля, постройте следующую формулу.

Возраст: Year(Date())-Year([ЛичныеДанные].[ДатаРождения])

Здесь с помощью встроенной функции **Year** из поля типа Дата/время извлекается год. Встроенная функция **Date()** используется для получения текущей даты. [ЛичныеДанные] – название таблицы, содержащей поле [ДатаРождения].

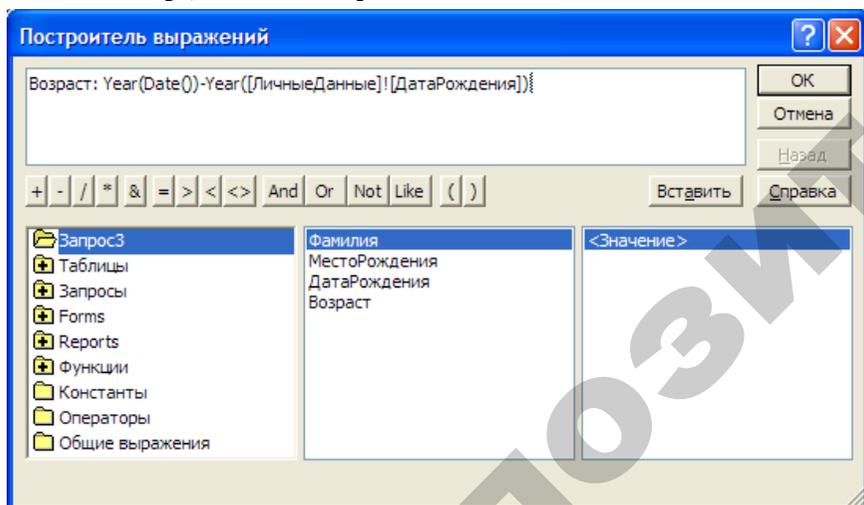


Рисунок 3.6 – Диалоговое окно **Построитель выражений** для создания вычисляемого поля

4. Для возвращения в бланк запроса нажмите кнопку **ОК**. Окончательный вид бланка запроса представлен на рисунке 3.7.

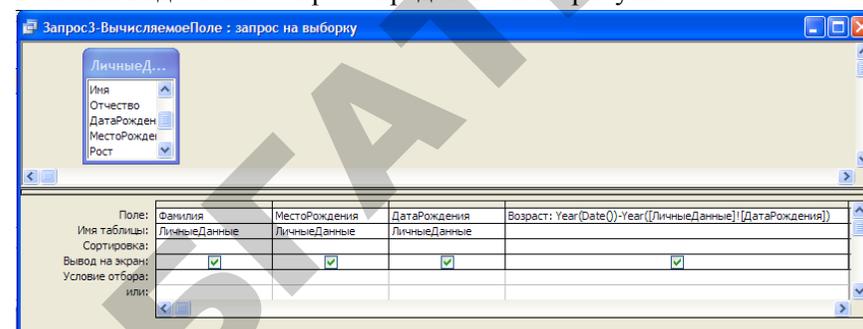


Рисунок 3.7 – Бланк запроса с вычисляемым полем

5. Сохраните запрос под именем **Запрос3**. Выполните запрос нажатием клавиши меню – **Запуск**. Результат выполнения запроса приведен на рисунке 3.8.

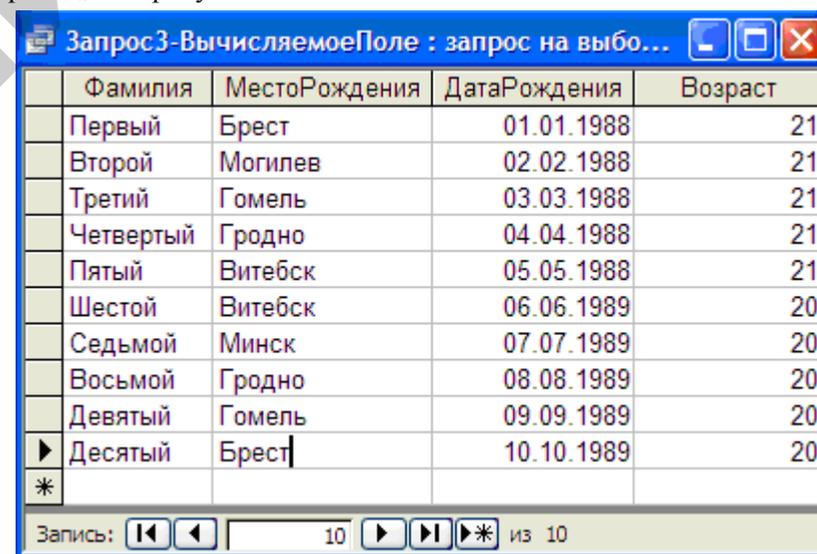


Рисунок 3.8 – Результат выполнения запроса с вычисляемым полем

Перечень основных операторов, используемых при создании выражений, приведен в Приложении 1.

## 2.4. Групповые операции в запросах

Создать **Запрос4** на подсчет количества студентов, проживающих в общежитии.

Для создания запросов с использованием групповых операций в бланк запроса включаются поля, по которым надо произвести группировку и поля, по которым надо произвести итоговые вычисления. Для расчета итоговых вычислений используют стандартные итоговые функции. К итоговым функциям относятся:

- **Sum** – вычисляет сумму всех значений заданного поля в каждой группе;
- **Avg** – вычисляет среднее арифметическое всех значений данного поля в каждой группе;
- **Min/Max** – возвращает наименьшее (наибольшее) значение, найденное в данном поле внутри каждой группы;
- **Count** – возвращает число непустых записей в данном поле для каждой группы;
- **First / Last** – возвращает первое (последнее) значение данного поля в группе.

Для создания запроса выполните следующие действия.

1. Перейдите к объектам **Запросы** в левой части окна БД **Деканат** и нажмите кнопку **Создать** в окне БД. В окне **Новый запрос** выберите **Конструктор** и нажмите кнопку **ОК**.
2. В диалоговое окно конструктора запроса добавьте таблицу **ЛичныеДанные**, в которой находятся запрашиваемые поля.
3. Поместите в бланк запроса поля: **КодСтудента** и **Общежитие**.
4. В соответствии с заданием измените название поля **КодСтудента** на **Количество**, для чего перед названием поля **КодСтудента** необходимо набрать текст **Количество:**, причем, наличие символа : (двоеточие) обязательно.
5. Для добавления поля **Групповые операции** в бланк запроса выберите в меню **Вид** команду **Групповая операция**. Тот же результат можно получить нажатием кнопки  на панели инструментов **Конструктора запросов**.
6. Установить для поля **Групповая операция** в бланке запроса следующие значения для соответствующих полей:
  - Количество:КодСтудента – **Count**;
  - Общежитие – **Условие**.

7. Установите в поле **Условия отбора** значения в соответствии с требованием запроса. Окно конструктора запроса должно выглядеть так, как показано на рисунке 3.9.

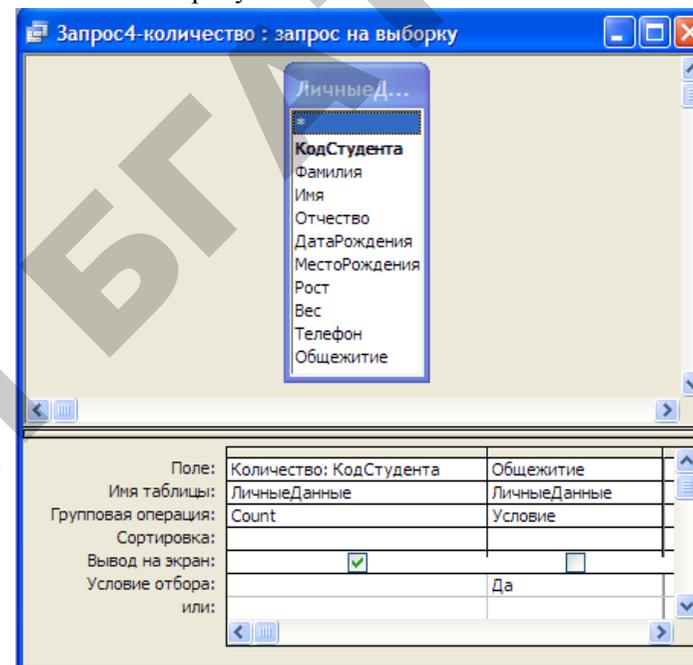


Рисунок 3.9 – Диалоговое окно конструктора запроса

8. Сохраните запрос под именем **Запрос4** и убедитесь, что он появился в объектах **Запросы**. Выполнить запрос. Результат выполнения запроса на рисунке 3.10.

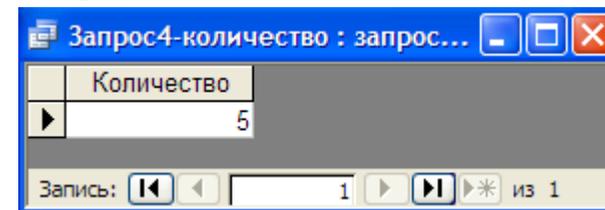


Рисунок 3.10 – Результат выполнения Запрос4-количество

Перечень основных операторов, используемых при групповых операциях в запросах, приведен в Приложении 2.

## 2.5. Создание перекрестного запроса

Создать **Запрос5-перекрестный** на получение сводной таблицы, в которой указаны Фамилии студентов и их оценки по четырем дисциплинам, а также средний балл каждого студента. Фамилии должны быть отсортированы в алфавитном порядке.

Перекрестный запрос создается в два этапа. Сначала надо создать вспомогательный запрос на выборку полей, которые участвуют в формировании перекрестного запроса. Затем с помощью Мастера создать перекрестный запрос.

1. Для создания вспомогательного запроса на выборку в диалоговое окно конструктора запроса добавьте таблицы **ЛичныеДанные**, **Успеваемость** и **Дисциплины**, в которых находятся запрашиваемые поля.
2. Поместите в бланк запроса следующие поля: **Фамилия**, **НазваниеДисциплины**, **Оценка**.
3. Выполните запрос и сохраните его под именем **ЗпрДляПерекрестного**.
4. Для создания перекрестного запроса с помощью Мастера на вкладке **Запросы** нажмите кнопку **Создать**. В окне **Новый запрос** выберите **Перекрестный запрос**, нажмите ОК.
5. В диалоговом окне **Создание перекрестных таблиц** щелкните по переключателю **Запросы**, выберите запрос **ЗпрДляПерекрестного** и нажмите кнопку **Далее** (рисунок 3.11).

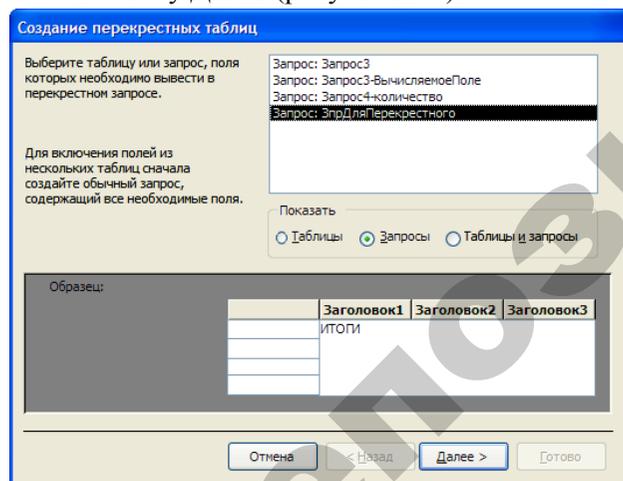


Рисунок 3.11 – Создание перекрестного запроса: выбор исходных данных

6. В следующем окне выберите поле **Фамилия**, которое будет использоваться в качестве заголовков строк, и щелкните по кнопке **Далее** (рисунок 3.12).

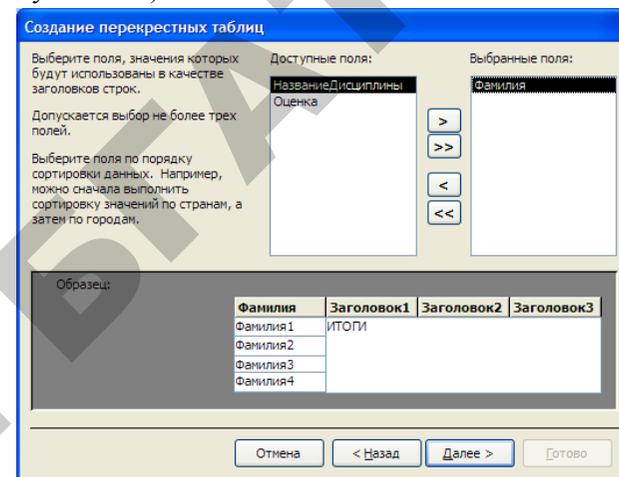


Рисунок 3.12 – Создание перекрестного запроса: выбор поля для заголовков строк

7. Выберите поле **НазваниеДисциплины**, которое будет использоваться в качестве заголовков столбцов (рисунок 3.13), и щелкните по кнопке **Далее**.

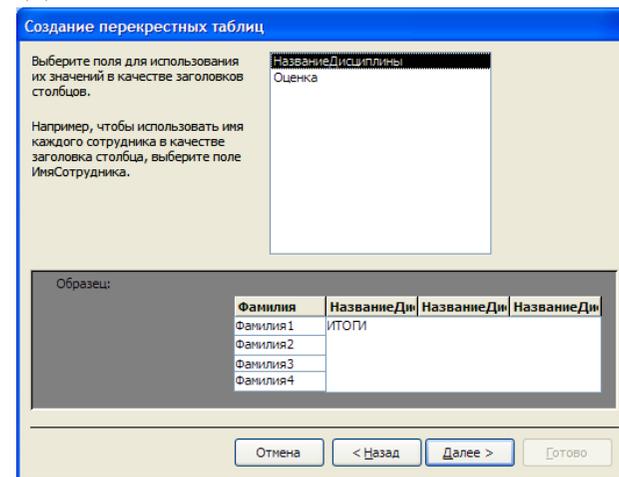


Рисунок 3.13 – Создание перекрестного запроса: выбор поля для заголовков столбцов

8. Выберите поле **Оценка** и функцию **Среднее**, щелкните по кнопке **Далее** (рисунок 3.14).

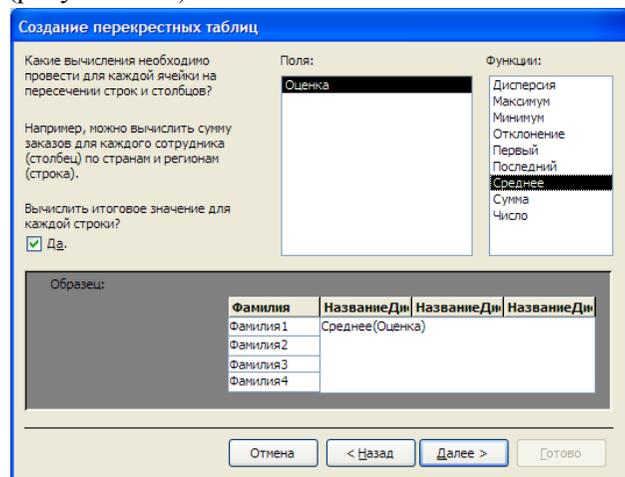


Рисунок 3.14 – Создание перекрестного запроса: выбор функции

9. Введите название запроса **Запрос5-перекрестный** и щелкните по кнопке **Готово** (рисунок 3.15). Обратите внимание на то, что Access создает еще итоговое значение среднего балла каждого студента.

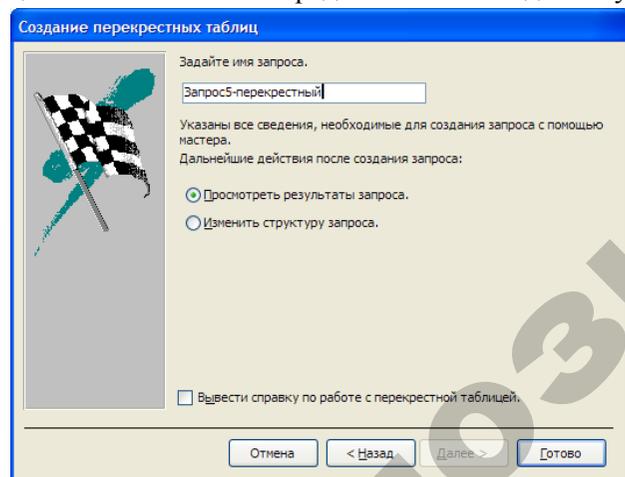


Рисунок 3.15 – Создание перекрестного запроса: имя запроса

10. Выполните **Запрос5-перекрестный**. Результат выполнения запроса на рисунке 3.16.

	Итоговое значение	Машины и оборудование	Теплотехника	Технология произв	Тракторы и автомобили
Восьмой	8,5	9	8	9	8
Второй	4,5	5	4	5	4
Девятый	9	9	9	9	9
Десятый	9,5	10	9	10	9
Первый	4	4	4	4	4
Пятый	6	6	6	6	6
Седьмой	8	8	8	8	8
Третий	5	5	5	5	5
Четвертый	5,5	6	5	6	5
Шестой	7	7	7	7	7

Рисунок 3.16 – Результат выполнения Запроса5

## 2.6. Задания для самостоятельного выполнения

1. Создайте запрос на выборку следующей информации о студентах: фамилия, место рождения, рост, рост в футах. Информацию вывести только о студентах, проживающих в общежитии. Информацию отсортировать в порядке уменьшения роста студентов. Сохраните запрос под именем **Запрос\_Рост\_в\_Футах**.

*РостФут – рост студента в футах. 1 фут ≈ 0,3048м.*

Результат выполнения запроса – на рисунке 3.17.

	Фамилия	Общежитие	Рост	РостФут
▶	Десятый	☑	180	5,9055
	Восьмой	☑	178	5,8399
	Шестой	☑	176	5,7743
	Четвертый	☑	174	5,7087
	Третий	☑	173	5,6759
*		☐	0	

Рисунок 3.17 – Результат выполнения запроса.

2. Создайте параметрический на выборку информации о результатах сдачи экзамена, название которого необходимо указать в диалогом окне как параметр. Запрос должен содержать следующие поля: название дисциплины, фамилия студента, оценка. Сохраните запрос под именем **Запрос\_Экзамен**.

Информацию отсортировать в алфавитном порядке фамилий студентов.

3. Создать запрос для определения среднего балла по каждой дисциплине. Информацию отсортировать в порядке убывания среднего балла. Сохраните запрос под именем **Запрос\_СреднийБалл**.  
Результат выполнения запроса на рисунке 3.18.

НазваниеДисциплины	Средний балл
Машины и оборудование в животноводстве	6,9
Теплотехника	6,5
Технология производства и ремонта	6,9
Тракторы и автомобили	6,5

Рисунок 3.18 – Результат выполнения **Запроса\_СреднийБалл**

Функции для работы с данными типа дата/время приведены в Приложении 3. Функции для работы со строковыми данными приведены в Приложении 4.

### 3. Вопросы для самоконтроля

1. Для чего служат запросы?
2. В чем состоит особенность запросов?
3. Какие типы запросов можно создать в *Microsoft Access*?
4. Для чего используются запросы на выборку?
5. Что такое запрос с параметром и когда его удобно использовать?
6. Для чего используются перекрестные запросы?

### 4. Варианты индивидуальных заданий

В каждом варианте создать различные виды запросов. Результатом выполнения каждого запроса должно быть не менее трех записей, что, скорее всего, потребует корректировки информации в таблицах.

Курсивом отмечены вычисляемые поля в запросах.

#### Вариант 1. БД Факультет.

1. **Запрос1** на условную выборку следующей информации о сотрудниках кафедры, имеющих стаж работы более 10 лет (информацию отсортировать в порядке уменьшения стажа работы):

- фамилия сотрудника;
- дата рождения;
- стаж работы;
- наименование кафедры;
- номер телефона;
- *возраст сотрудника.*

2. Параметрический **Запрос2** на выборку всей информации о сотрудниках кафедры, наименование которой вводит пользователь.

3. **Запрос3** для подсчета количества сотрудников факультета, старше 50 лет.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список кафедр факультета, для каждой кафедры указать количество сотрудников каждой ученой степени и общее количество сотрудников.

#### Вариант 2. БД Программы.

1. Создать **Запрос1** на условную выборку следующей информации о программных продуктах, имеющих цену менее 50000 рублей (информацию отсортировать в порядке возрастания цены):

- шифр программного продукта;
- название фирмы изготовителя;
- наименование программного продукта;
- цена одного экземпляра;
- количество экземпляров;
- *общая стоимость программного обеспечения.*

2. **Запрос2** на выборку всей информации о программных продуктах фирмы изготовителя, название которой вводит пользователь.

3. **Запрос3** для подсчета общей стоимости всей программных продуктов.

4. **Запрос4** – перекрестный запрос, в котором сводная таблица должна выводить список фирм-изготовителей программного обеспечения, для каждой фирмы указать кол-во экземпляров каждого типа программного продукта и общее количество представленных программных продуктов.

#### Вариант 3. БД Склад.

1. **Запрос1** на условную выборку следующей информации о товарах, имеющих цену более 50000 рублей (информация должна быть отсортирована в порядке увеличения количества экземпляров товара на складе):

- название товара;
- цена товара;
- название фирмы-поставщика;
- количество экземпляров товара на складе;
- *общая стоимость товара.*

2. **Запрос2** на выборку всей информации о товарах, название фирмы-поставщика которого вводит пользователь.

3. **Запрос3** для подсчета общей стоимости всех товаров на складе.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список фирм-поставщиков, для каждого поставщика указать количество экземпляров каждого товара и общее количество представленного товара.

#### Вариант 4. БД Платежи.

1. **Запрос1** на условную выборку следующей информации о платежах, которые превысили 100000 рублей (информация должна быть отсортирована в порядке уменьшения сумм платежей):

- дата платежа;
- название организации;
- название банка;
- сумма платежа;
- пеня;
- *к оплате.*

2. **Запрос2** на выборку всей информации о платежах, проведенных организацией, название которой вводит пользователь.

3. **Запрос3** для подсчета общей суммы платежей.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список организаций, для каждой организации указать количество платежей, проведенных каждым банком, а также общее количество платежей.

#### Вариант 5. БД Комплекующие к ПК

1. **Запрос1** на условную выборку следующей информации об изделиях, цена которых меньше 100 у.е. (наименование фирм-изготовителей должно быть отсортировано в алфавитном порядке):

- порядковый номер;
- наименование изделия;
- наименование фирмы-изготовителя;
- цена;
- количество экземпляров;
- *общая стоимость.*

2. **Запрос2** на выборку всей информации об изделиях фирмы-изготовителя, наименование которой задает пользователь.

3. **Запрос3** для подсчета общей стоимости изделий.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список фирм – изготовителей, для каждой фирмы указать количество экземпляров каждого изделия, а также общее количество изделий.

#### Вариант 6. БД Музыка.

1. **Запрос1** на условную выборку следующей информации о компакт-дисках, цена которых не превышает 5000 рублей (фамилии исполнителей отсортировать в алфавитном порядке):

- название компакт-диска;
- название произведения;
- фамилия исполнителя;
- длительность воспроизведения в минутах;
- *длительность воспроизведения в часах;*
- цена.

2. **Запрос2** на выборку компакт-дисков с исполнителем, фамилию которого вводит пользователь.

3. **Запрос3** для подсчета общего времени воспроизведения.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список исполнителей, для каждого исполнителя указать длительность воспроизведения музыкального произведения для каждого компакт-диска, а также общее время звучания.

#### Вариант 7. БД Абитуриент.

1. **Запрос1** на условную выборку следующей информации об абитуриентах младше 18 лет (фамилии должны быть отсортированы в алфавитном порядке):

- фамилия, имя, отчество;
- год рождения;
- наименование предмета;
- полученная оценка;
- средний балл;
- *возраст абитуриента.*

2. **Запрос2** на выборку информации о результатах сдачи экзамена, название которого вводит пользователь.

3. **Запрос3** для вывода значения среднего балла, полученного на вступительных экзаменах.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить информацию в следующем виде.

	Средний балл	Предмет1	...	Предмет N
Абитуриент1	Значение1	оценка1	...	оценка1
...	...	...	...	...
Абитуриентk	Значениеk	оценкаk	...	оценкаk

#### Вариант 8. БД Зарплата.

1. **Запрос1** на условную выборку следующей информации о работниках, которые были на больничном более трех дней, и у которых сумма удержания превышает 20000 рублей (информация должна быть отсортирована в порядке уменьшения суммы заработка):

- название цеха;
- Ф.И.О.;
- домашний адрес;
- домашний телефон;
- начисляемый заработок;
- удержано
- *к выплате;*
- дата выдачи.

2. **Запрос2** на выборку всей информации о работниках цеха, название которого вводит пользователь.

3. **Запрос3** для вывода значения фонда заработной платы.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список цехов, для каждого цеха подсчитать количество работников, которые провели на больничном 0, 3, 5 и т.д. дней, а также общее число работников в каждом цехе.

#### Вариант 9. БД Телефон.

1. **Запрос1** на условную выборку следующей информации об абонентах, продолжительность разговора которых превышает 10 минут (информацию отсортировать в порядке увеличения продолжительности разговора):

- номер телефона;
- дата разговора;
- название города;
- фамилия владельца телефона;
- адрес владельца телефона;
- *продолжительность разговора.*

2. **Запрос2** на выборку всей информации об абонентах, звонивших в заданный город, название которого вводит пользователь. Фамилии абонентов отсортировать в алфавитном порядке.

3. **Запрос3** для подсчета общего времени разговоров по каждому городу.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить для каждого абонента время разговоров по каждому городу, а также общее время разговоров каждого абонента.

#### Вариант 10. БД Отдел.

1. **Запрос1** на условную выборку информации о сотрудниках отдела, имеющих зарплату не ниже 350 000 руб. (фамилии должны быть отсортированы в алфавитном порядке). Заработную плату рассчитать как сумму месячного оклада и премии.

- фамилия сотрудника;
- дата рождения;
- наименование области;
- населенный пункт;
- месячный оклад;
- премия в процентах;
- *зарплата.*

2. **Запрос2** на выборку всей информации о сотрудниках отдела, родившихся в той области Беларуси, наименование которой вводит пользователь.

3. **Запрос3** для подсчета месячного фонда заработной платы сотрудников отдела.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список областей, для каждой области определить количество сотрудников, родившихся в этой области по годам рождения и общее количество сотрудников.

### Вариант 11. БД Квартира

1. **Запрос1** на условную выборку следующей информации о квартирах, общая площадь которых больше 80 кв. метров (информация должна быть отсортирована в порядке уменьшения общей площади):

- дата выставления на продажу;
- название района;
- количество комнат;
- общая площадь в кв. метрах;
- наличие телефона;
- оценочная стоимость квартиры;
- *цена 1 кв. метра.*

2. **Запрос2** на выборку всей информации о квартирах района, который задает пользователь.

3. **Запрос3** для подсчета общего количества квартир, выставленных для продажи для каждого района города;

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить для каждого района города количество 1-комнатных, 2-комнатных и т.д. квартир, а также общее количество квартир в каждом районе.

### Вариант 12. БД Справочник врача

1. **Запрос1** на условную выборку следующей информации о пациентах, возраст которых старше 60 лет (фамилии пациентов должны быть отсортированы в алфавитном порядке):

- фамилия и инициалы;
- *возраст;*
- дата обращения к врачу;
- домашний адрес;
- диагноз.

2. **Запрос2** на выборку всей информации о пациентах с диагнозом, название которого вводит пользователь.

3. **Запрос3** для вывода общего количества пациентов

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна иметь вид

	Общее кол-во пациентов	Дата обращения 1	...	Дата обращения N
Диагноз 1	...	...	...	...
...	...	...	...	...
Диагноzk	...	...	...	...

### Вариант 13. БД Почта

1. **Запрос1** на условную выборку следующей информации о почтовых отправлениях, вес которых больше 1500 граммов (информация должна быть отсортирована в порядке уменьшения веса):

- дата отправки;
- город;
- фамилия получателя;
- адрес отправителя;
- фамилия отправителя;
- *вес почтового отправления в килограммах.*

2. **Запрос2** на выборку всей информации о почтовых отправлениях в город, название которого вводит пользователь.

3. **Запрос3** для подсчета общего веса всех почтовых отправлений.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна содержать информацию о количестве почтовых отправлений в каждый город по датам, а также общее количество отправлений в каждый город.

### Вариант 14. БД Автомобили

1. **Запрос1** на условную выборку следующей информации об автомобилях, выпущенных после 2008 года (информация должна быть отсортирована в порядке уменьшения пробега):

- марка автомобиля;
- название страны изготовителя;
- год выпуска;
- пробег в километрах;
- стартовая цена;
- *возраст автомобиля.*

2. **Запрос2** на выборку всей информации об автомобилях той марки, которую задает пользователь.

3. **Запрос3** для вывода предполагаемой выручки от продажи всех автомобилей.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна содержать информацию для вывода информации об общем количестве выставленных на продажу автомобилей каждой марки, а также количество автомобилей каждого года выпуска. Примерный вид сводной таблицы представлен ниже.

	Общее кол-во автомобилей	Год выпуска 1	...	Год выпуска n
Марка автомобиля 1	...	...	...	...
...	...	...	...	...
Марка автомобиля k	...	...	...	...

#### Вариант 15. БД Учет компьютерных средств

1. **Запрос1** на условную выборку информации о тех компьютерах, расположенных на 2 этаже (№ ауд. начинается с «2»). Информацию отсортировать в порядке возрастания № ауд.

- наименование подразделения;
- инвентарный № компьютера;
- наименование компьютера;
- место расположения;
- год ввода в эксплуатацию;
- стоимость;
- *стоимость после переоценки.*

«Стоимость после переоценки» рассчитывается следующим образом. Если возраст компьютера более 5 лет – уценка 20% от первоначальной стоимости.

2. **Запрос2** на выборку всей информации о компьютерах, закрепленных за подразделением университета, наименование которого вводит пользователь.

3. **Запрос3** для подсчета количества компьютеров в каждом подразделении университета.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить информацию об общем количестве компьютеров в каждом подразделении университета, а также количество компьютеров каждого года выпуска.

#### Вариант 16. БД Успеваемость

1. **Запрос1** на условную выборку информации о результатах сдачи сессии студентами группы (номер группы указать самостоятельно). (ФИО отсортировать в алфавитном порядке).

- название предмета;
- Ф.И.О. студентов;
- оценка;
- дата сдачи экзамена;
- группа.

2. **Запрос2** на выборку *всей* информации о студенте, ФИО которого вводит пользователь. Вычислить возраст студента в текущем году.

3. **Запрос3** для определения среднего балла по каждому предмету.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список студентов, для каждого студента указать оценки, полученные по предметам, а также средний балл за сессию.

#### Вариант 17. БД Заказы

1. **Запрос1** на условную выборку информации о заказах на работы, стоимость которых больше 2 000 000 руб. (информацию отсортировать по полю «Название фирмы» в алфавитном порядке).

- № заказа;
- фирма-клиент;
- Ф.И.О. сотрудника;
- дата завершения работ;
- стоимость работ;
- форма оплаты.

2. **Запрос2** на выборку *всей* информации о сотруднике, ФИО которого вводит пользователь. Рассчитать *вознаграждение (в рублях)* сотрудника по каждому заказу.

3. **Запрос3** для определения общей суммы вознаграждения, полученной каждым сотрудником.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить информацию о сумме вознаграждения, полученной каждым сотрудником по каждому клиенту, а также общей суммы вознаграждения каждого сотрудника.

#### Вариант 18. БД Реализация готовой продукции

1. **Запрос1** на условную выборку информации о сделках, сумма которых превышает 1 000 000 рублей (записи отсортировать в порядке возрастания суммы сделки).

- код сделки;
- дата сделки;
- наименование товара;
- наименование покупателя;
- оптовая цена;
- количество;
- *сумма сделки.*

2. **Запрос2** на выборку *всей* информации о товаре, наименование которого вводит пользователь в диалоговом режиме.

3. **Запрос3 (итоговый)** - для каждого покупателя рассчитать общую сумму проведенных сделок.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить информацию в виде

	Общая сумма сделки	Наименование товара 1	...	Наименование товара N
Покупатель 1	...	...	...	...
...	...	...	...	...
Покупатель k	...	...	...	...

### Вариант 19. БД Транспорт

1. **Запрос1** на условную выборку следующей информации о перевозках, в которых пробег превысил 1000 километров (информацию отсортировать в порядке убывания пробега):

- номер рейса;
- дата рейса;
- фамилия водителя;
- номер автомобиля;
- марка автомобиля;
- вес груза в тоннах;
- пробег в километрах.

2. **Запрос2** на выборку всей информации о перевозках в определенный день, который вводит пользователь.

3. **Запрос3** для подсчета общего числа *тонно-километров*. Тонно-километр рассчитывается как произведение количества тонн груза и расстояния в километрах.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список водителей, для каждого водителя указать количество рейсов, совершенных на различных автомобилях, а также общее количество рейсов для каждого водителя.

### Вариант 20. БД Расписание.

1. **Запрос1** на условную выборку следующей информации о всех скорых поездах (№ поезда от 1 до 99). Информация должна быть отсортирована в порядке увеличения времени в пути:

- № поезда;
- категория;
- пункт отправления (значение по умолчанию – Минск);
- пункт назначения;

- время отправления; время прибытия;
- *время в пути*;
- количество вагонов в составе;
- наличие вагона-ресторана.

2. **Запрос2** на выборку *всей* информации о пассажирских поездах (№ поезда от 100 и больше), прибывающих в Минск в промежуток времени, который задает пользователь.

3. **Запрос3** (итоговый запрос), который выводит список пунктов назначения и определяет количество поездов, следующих в каждый из этих городов.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна содержать список пунктов назначения, для каждого города указать количество скорых и пассажирских поездов (категория поезда).

### Вариант 21. БД Соревнования

1. **Запрос1** на условную выборку следующей информации об участниках, моложе 18 лет и занявших призовые места (фамилии должны быть отсортированы в алфавитном порядке):

- Ф.И.О. участника;
- год рождения;
- вид спорта;
- результат;
- занятое место;
- *возраст спортсмена*.

2. **Запрос2** на выборку всей информации об участниках, занявших первые места в том виде спорта, название которого вводит пользователь.

3. **Запрос3** для вывода общего количества участников соревнований по каждому виду спорта.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить информацию о количестве спортсменов по видам спорта для каждого стадиона, а также общее количество участников, выступавших на каждом стадионе. Примерный вид сводной таблицы представлен ниже.

	Общее кол-во участников	Стадион 1	...	Стадион N
Вид спорта 1	...	....	....	...
...	...	...	...	...
Вид спорта k	....	...	...	...

### Вариант 22. БД Праздничное бюро

1. **Запрос1** на условную выборку информации о мероприятиях, проведенных во Дворце Республики (ФИО сотрудников отсортировать в алфавитном порядке):

- название мероприятия;
- Ф.И.О. сотрудника;
- место проведения;
- дата проведения;
- время начала;
- количество мест;
- цена за 1 место;
- *общая стоимость мероприятия.*

2. **Запрос2** на выборку всей информации о мероприятиях, проведенных сотрудником, ФИО которого вводит пользователь в диалоговом режиме.

3. **Запрос3** (итоговый запрос) для каждого сотрудника общую сумму полученных комиссионных (в рублях).

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна выводить список мест проведения, для каждого ресторана указать количество мероприятий каждого уровня (*название мероприятия*), а также общее количество проведенных мероприятий.

### Вариант 23. БД Составление учебного плана

1. **Запрос1** на условную выборку информации о предметах с формой контроля – экзамен. Сортировку провести в алфавитном порядке ФИО преподавателей.

- название предмета;
- номер группы;
- Ф.И.О. преподавателя;
- количество часов;
- *количество часов в неделю;*
- итоговый контроль.

2. **Запрос2** на выборку *всей* информации о нагрузке преподавателя, ФИО которого вводит пользователь.

3. **Запрос3** (итоговый запрос), который выводит список преподавателей и суммарное количество часов нагрузки каждого.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна содержать для каждой группы количество часов по каждому предмету и суммарное количество часов.

### Вариант 24. БД Туристическая фирма

1. **Запрос1** на условную выборку следующей информации о путевках, предлагающих отели \*\*\*. Записи отсортировать в порядке убывания цены путевки.

- Ф.И.О. клиента;
- страна;
- отель;
- дата отправления;
- *дата возвращения;*
- *итоговая цена путевки.*

2. **Запрос2** на выборку *всей* информации о путевках в страну, название которой задает пользователь.

3. **Запрос3** (итоговый запрос), который определяет количество предложений в каждую страну.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна содержать список стран, для каждой страны указать количество отелей каждой звездности, а также общее количество отелей.

### Вариант 25. БД Грузовые перевозки

1. **Запрос1** на условную выборку следующей информации о грузовых перевозках, длительность которых превышает 5 дней. Записи отсортировать в порядке возрастания даты отправки.

- код маршрута;
- Ф.И.О. водителя;
- пункт назначения;
- дата отправки;
- *длительность рейса;*
- оплата;
- премия в %;
- *вознаграждение водителю.*

2. **Запрос2** на выборку *всей* информации о перевозках в **пункт назначения**, название которого задает пользователь.

3. **Запрос3** (итоговый запрос), который выводит список водителей и для каждого водителя определяет общую сумму вознаграждения.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна содержать список водителей, для каждого водителя указать количество рейсов в каждый пункт назначения и общее количество рейсов.

### Вариант 26. БД Прокат автомобилей.

1. **Запрос1** на условную выборку следующей информации о прокате автомобилей стоимость которых не превышает 20 000 \$. Записи отсортировать в алфавитном порядке Ф.И.О. клиентов.

- Ф.И.О. клиента;
- марка автомобиля;
- стоимость автомобиля;
- дата выдачи;
- дата возврата;
- *длительность проката*;
- стоимость проката (за 1 день);
- *к оплате*.

2. **Запрос2** на выборку *всей* информации о прокате автомобилей, марку которых задает пользователь.

3. **Запрос3** (итоговый запрос), который выводит список автомобилей и для каждого автомобиля определяет количество выданных.

4. **Запрос4** (перекрестный запрос), в котором сводная таблица должна содержать список автомобилей, для каждого автомобиля указать стоимость проката по месяцам года и общую сумму, полученную за прокат каждого автомобиля.

### Лабораторная работа №4

#### СОЗДАНИЕ ЗАПРОСОВ НА ИЗМЕНЕНИЕ

**Цель лабораторной работы:** приобрести навыки создания запросов на изменение (модифицирующих запросов).

#### **Порядок выполнения лабораторной работы**

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания запросов БД **Деканат**. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.

#### **1. Краткие теоретические сведения**

**Запросы на изменение.** Запросом на изменение называют запрос, который за одну операцию изменяет или перемещает несколько записей. Существует четыре типа запросов на изменение.

- **На создание таблицы.** Запрос на создание таблицы создает новую таблицу на основе всех или части данных из одной или нескольких таблиц. Запрос на создание таблицы полезен при создании таблицы для экспорта в другие БД *Microsoft Access* или при создании архивной таблицы, содержащей старые записи.

- **На обновление записей.** Запрос на обновление вносит общие изменения в группу записей одной или нескольких таблиц. Например, на 10 процентов поднимаются цены на все молочные продукты или на 5 процентов увеличивается зарплата сотрудников определенной категории. Запрос на обновление записей позволяет изменять данные в существующих таблицах.

- **На удаление записей.** Запрос на удаление удаляет группу записей из одной или нескольких таблиц. Например, запрос на удаление позволяет удалить записи о товарах, поставки которых прекращены или на которые нет заказов. С помощью запроса на удаление можно удалять только всю запись, а не отдельные поля внутри нее.

- **На добавление записей.** Запрос на добавление добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц. Например, появилось несколько новых клиентов, а также база данных, содержащая сведения о них. Чтобы не вводить все данные вручную, их можно добавить в таблицу «Клиенты».

## 2. Модифицирующие запросы

### 2.1. Запрос на создание таблицы из двух таблиц

Разработать **Запрос6**, который создает таблицу **Тбл1-СрБалл** включающую информацию о студентах: **Ф.И.О.**, **средний балл по дисциплинам, общежитие**. Значения среднего балла должно быть отсортировано по возрастанию.

Для создания запроса выполните следующие действия.

1. В диалоговое окно конструктора запроса добавьте таблицы **ЛичныеДанные** и **Успеваемость** и определите условия выполнения в бланке запроса, как показано на рисунке 4.1.

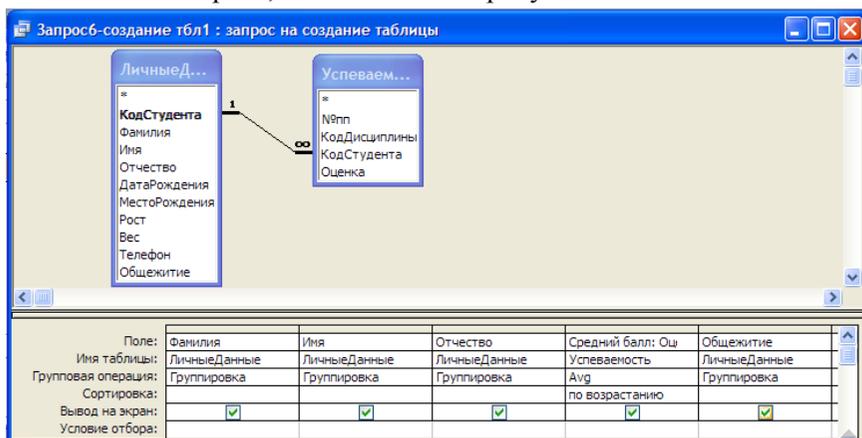


Рисунок 4.1 – Окно конструктора запроса на создание таблицы **Тбл1-СрБалл**

2. Выполните команду **Запрос → Создание таблицы** либо нажмите кнопку  (тип запроса) на панели инструментов **Конструктор запросов**. В диалоговое окно **Создание таблицы** (рисунок 4.2) введите имя новой таблицы – **Тбл1-СрБалл**.

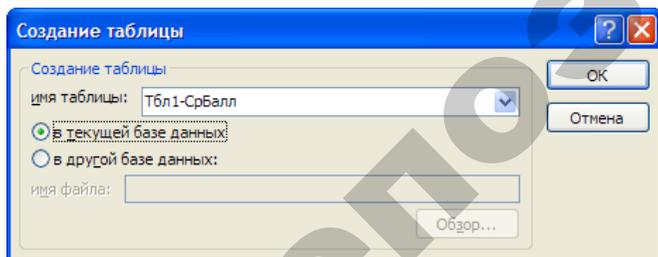


Рисунок 4.2 – Диалоговое окно **Создание таблицы**

3. Сохраните запрос под именем **Запрос6** и убедитесь, что он появился в объектах **Запросы**.

4. Выполните **Запрос6**. Выводится два приглашения. Первое приглашение требует подтвердить выполнение запроса на изменение, второе - на добавление 10 записей в новую таблицу. На оба приглашения ответьте «Да» (рисунок 4.3 и рисунок 4.4).

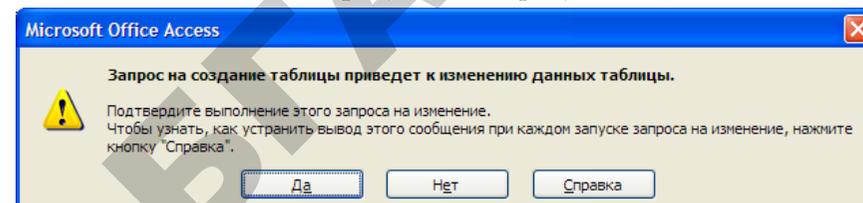


Рисунок 4.3 – Приглашение подтвердить выполнение запроса на изменение

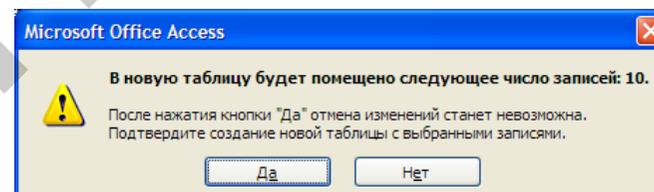


Рисунок 4.4 – Приглашение подтвердить создание таблицы

5. После выполнения запроса в БД создается новая таблица **Тбл1-СрБалл**, которую можно просмотреть на вкладке **Таблицы**. Результат выполнения запроса показан на рисунке 4.5.

	Фамилия	Имя	Отчество	Средний балл	Общежитие
▶	Первый	Петр	Петрович	4	0
	Второй	Виктор	Викторович	4,5	0
	Третий	Тимофей	Тимофеевич	5	-1
	Четвертый	Чеслав	Чеславович	5,5	-1
	Пятый	Павел	Павлович	6	0
	Шестой	Шалва	Шалвович	7	-1
	Седьмой	Сергей	Сергеевич	8	0
	Восьмой	Владимир	Владимирович	8,5	-1
	Девятый	Денис	Денисович	9	0
*	Десятый	Дмитрий	Дмитриевич	9,5	-1

Рисунок 4.5 – Результат выполнения запроса на создание таблицы **Тбл1-СрБалл**

## 2.2. Запрос на создание таблицы из трех таблиц

Разработать *Запрос7*, который создает таблицу **Тбл2-НазначениеСтипендии**. В таблицу должна быть включена следующая информация: **Ф.И.О.**, **средний балл по дисциплинам**, **вид стипендии**, **ее размер**, **место рождения студента**, **код студента**.

Для создания запроса выполните следующие действия.

1. В диалоговое окно конструктора запроса добавьте три таблицы: **ЛичныеДанные**, **Тбл1-СрБалл** и **Стипендия**. В данном примере необходимо провести выборку из таблиц, у которых возможно совпадение значений в ячейках, и вывести эти совпадения в виде результата. Для этого требуется создать внутреннее соединение между таблицами по интересующим полям.

2. Соедините таблицы **ЛичныеДанные** и **Тбл1-СрБалл** по полю **Фамилия**. Для этого поле **Фамилия** в таблице **ЛичныеДанные** «перетащите» на аналогичное поле в таблице **Тбл1-СрБалл**. При этом получается линия с точками по обеим сторонам, означающая, что соединение выполнено между полями, связь которых в схеме данных не задана, и они не являются ключевыми полями. Поместить в бланк запроса следующие поля: **Ф.И.О.**, **СрБалл**, **Вид**, **Размер**, **МестоРождения**.

3. Как показано на рисунке 4.6, для поля **СрБалл** определить условие отбора:

`>=[Стипендия]![СрБаллМин] And <[Стипендия]![СрБаллМакс]`

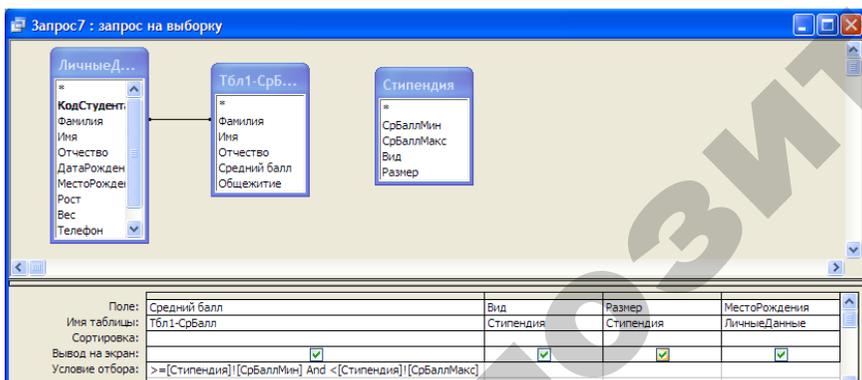


Рисунок 4.6 – Диалоговое окно конструктора запроса на создание таблицы **Тбл2-НазначениеСтипендии**

4. Выполните команду **Запрос** → **Создание таблицы** либо нажмите кнопку (тип запроса) на панели инструментов **Конструктор запросов**. В диалоговое окно **Создание таблицы** введите имя новой таблицы – **Тбл2-НазначениеСтипендии**.

5. Сохраните запрос под именем **Запрос7** и убедитесь, что он появился в объектах **Запросы**.

7. Выполните запрос. После выполнения запроса в БД создается новая таблица **Тбл2-НазначениеСтипендии**, которую можно просмотреть в объектах **Таблицы**. Результат выполнения запроса показан на рисунке 4.7.

Фамилия	Имя	Отчество	СрБалл	Вид	Размер	МестоРождения
Первый	Петр	Петрович	4	Социальная	78574	Брест
Второй	Виктор	Викторович	4,5	Социальная	78574	Могилев
Третий	Тимофей	Тимофеевич	5	Учебная	93006	Гомель
Четвертый	Чеслав	Чеславович	5,5	Учебная	93006	Гродно
Пятый	Павел	Павлович	6	Учебная	111607	Витебск
Шестой	Шалва	Шалвович	7	Учебная	111607	Витебск
Седьмой	Сергей	Сергеевич	8	Учебная	130208	Минск
Восьмой	Владимир	Владимирович	8,5	Учебная	130208	Гродно
Девятый	Денис	Денисович	9	Учебная	148810	Гомель
Десятый	Дмитрий	Дмитриевич	9,5	Учебная	148810	Брест

Рисунок 4.7 – Таблица **Тбл2-НазначениеСтипендии**, созданная с помощью *Запроса7*.

## 2.3. Запрос на обновление таблицы

В соответствии с Указом Президента Республики Беларусь № 711 от 29 декабря 2008 г. «Об увеличении всех видов государственных стипендий учащейся молодежи» Совету Министров РБ дано указание увеличить в 1,05 раза размеры всех видов государственных стипендий. Создать *Запрос8* на обновление таблицы **Стипендия**.

Для создания запроса выполните следующие действия.

1. В диалоговое окно конструктора запроса добавьте таблицу **Стипендия**.

2. Выполните команду **Запрос** → **Обновление** либо нажмите кнопку (тип запроса) на панели инструментов **Конструктор запросов**.

3. При переходе к запросу на обновление в бланке запроса исчезают строки **Сортировка** и **Вывод на экран** и появляется строка **Обновление**.

4. Заполните бланк запроса, как показано на рисунке 4.8.

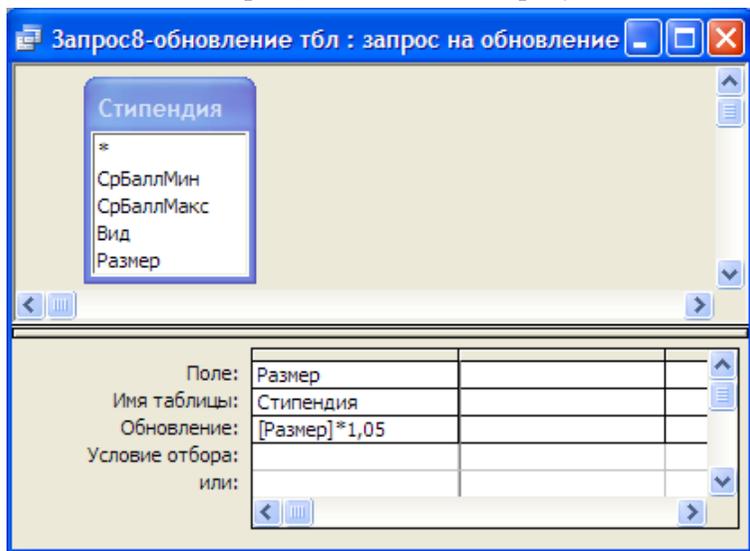


Рисунок 4.8 – Диалоговое окно конструктора запроса на обновление таблицы **Стипендия**

5. Сохраните запрос под именем **Запрос8** и убедитесь, что он появился в объектах **Запросы**.

6. Создайте копию таблицы **Стипендия**.

7. Запустите **Запрос8** на обновление. Выводится сообщение, в котором отображается число обновляемых записей (рисунок 4.9).

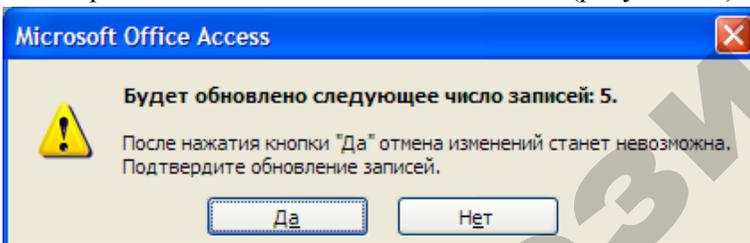


Рисунок 4.9 – Приглашение подтвердить обновление указанного числа записей

8. Перейдите на вкладку **Таблицы** и убедитесь в том, что записи в таблице **Стипендия** обновляются в соответствии с требованиями запроса.

## 2.4. Запрос на удаление записей из таблицы

Создать **Запрос9** на удаление из таблицы **Тбл2-НазначениеСтипендии** записей, которые удовлетворяют следующим критериям: название места рождения студента содержит первую букву «Б» или букву «и» в середине названия.

Перед удалением записей рекомендуется вывести их, используя запрос на выборку, а затем этот запрос преобразовать в запрос на удаление.

Для создания запроса выполните следующие действия.

1. В диалоговое окно конструктора запроса добавьте таблицу **Тбл2-НазначениеСтипендии**.
2. Выполните команду **Запрос** → **Удаление** либо нажмите кнопку  (тип запроса) на панели инструментов **Конструктор запросов**.
3. При переходе к запросу на удаление, в бланке запроса исчезают строки **Сортировка** и **Вывод на экран** и появляется строка **Удаление**.
4. Заполните бланк запроса, как показано на рисунке 4.10.

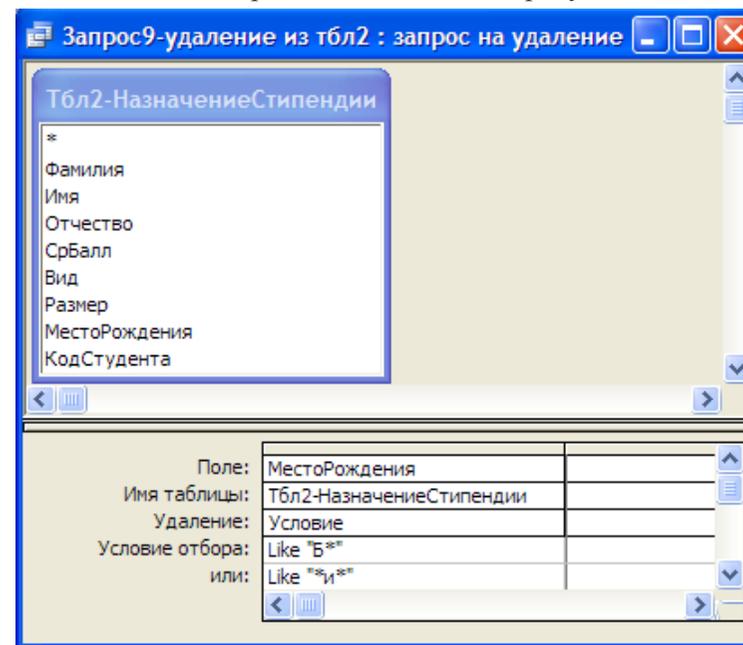


Рисунок 4.10 – Диалоговое окно конструктора запроса на удаление записей из таблицы **Тбл2-НазначениеСтипендии**

- Сохраните запрос под именем **Запрос9** и убедитесь, что он появился в объектах **Запросы**.
- Выполните **Запрос9**. Выводится приглашение подтвердить уничтожение записей (рисунок 4.11).
- Перейдите на вкладку **Таблицы** и убедитесь в том, что записи в таблице **Тбл2-НазначениеСтипендии** удаляются в соответствии с требованиями запроса.

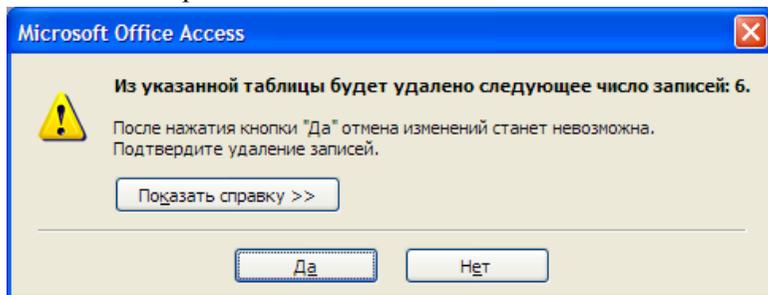


Рисунок 4.11 – Приглашение подтвердить удаление записей

### 2.5. Запрос на добавление записей в таблицу

Создать **Запрос10** на добавление в таблицу **Тбл2-НазначениеСтипендии** записей о студентах, которые родились в Витебске.

Для создания запроса выполните следующие действия.

- Создайте новый запрос и поместите в него таблицы **ЛичныеДанные**, **Тбл1-СрБалл** и **Стипендия**. Эти таблицы являются источниками для добавления записей в таблицу **Тбл2-НазначениеСтипендии**.
- Из таблиц-источников в бланк запроса переместите поля, которые входят в состав таблицы **Тбл2-НазначениеСтипендии**: **Фамилия**, **Имя**, **Отчество**, **СрБалл**, **Вид**, **Размер**, **МестоРождения**.
- Для полей **СрБалл** и **МестоРождения** определите условия отбора, как показано на рисунке 4.12.
- Выполните команду **Запрос** → **Добавление** либо нажмите кнопку  (тип запроса) на панели инструментов **Конструктор запросов**.
- В открывшемся диалоговом окне **Добавление** нужно указать таблицу, к которой будут добавлены записи – **Тбл2-НазначениеСтипендии** (рисунок 4.13).

- Сохраните запрос под именем **Запрос8** и убедитесь, что он появился в объектах **Запросы**.
- После запуска запроса в таблицу **Тбл2-НазначениеСтипендии** добавляются новые данные в соответствии с требованиями запроса.

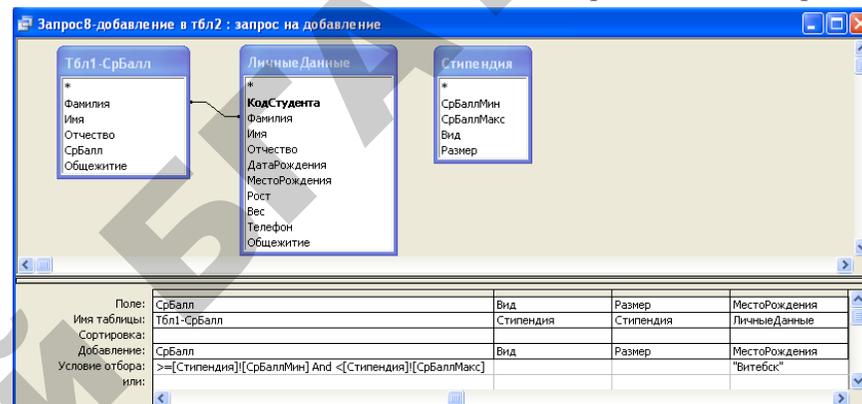


Рисунок 4.12 – Диалоговое окно конструктора запроса на добавление записей в таблицу **Тбл2-НазначениеСтипендии**

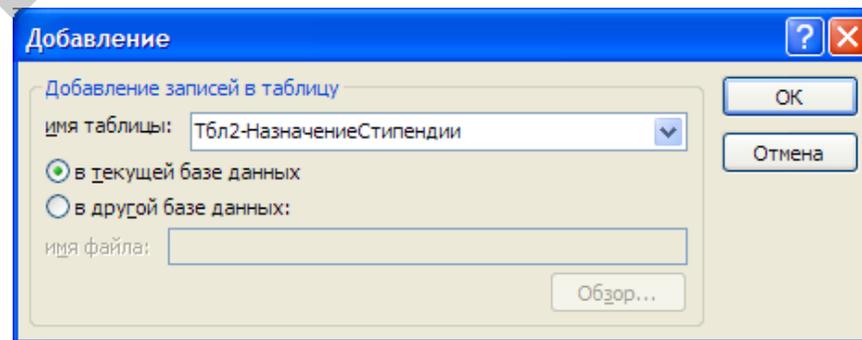


Рисунок 4.13 – Диалоговое окно **Добавление**

### 3. Вопросы для самоконтроля

- Какой запрос называют запросом на изменение?
- Какие типы запросов на изменение можно создать в *Microsoft Access*?
- Что такое запрос на создание таблицы и когда его удобно использовать?
- Что делает запрос на обновление записей?
- Что выполняет запрос на удаление записей?
- Что реализует запрос на добавление записей?

#### 4. Варианты индивидуальных заданий

В каждом варианте создать различные виды запросов. Результатом выполнения каждого запроса должно быть не менее трех записей, что, скорее всего, потребует корректировки информации в таблицах.

##### Вариант 1. БД Факультет

1. **Запрос5** на обновление таблицы1 – в связи с введением новой АТС первые три цифры номеров телефонов, начинающихся «236», заменить на «290». (!!!!Сначала создай копию таблицы1!!!!).
2. **Запрос6** на создание таблицы3 с полями *Фамилия сотрудника, Дата рождения, Стаж работы*. Включить в создаваемую таблицу записи о сотрудниках, имеющих стаж работы более 10 лет.
3. **Запрос7** на добавление в таблицу3 записей из таблицы1 о сотрудниках, стаж работы которых менее 5 лет.
4. **Запрос8** на удаление записей из таблицы3, касающихся сотрудников, старше 60 лет.

##### Вариант 2. БД Программы

1. **Запрос5** на обновление таблицы1. Если «количество экземпляров» превышает 15, то «цену одного экземпляра» снизить на 5%. (!!!!создай копию таблицы1!!!!)
2. **Запрос6** на создание таблицы3 с полями *Название фирмы-изготовителя, Наименование продукта, Цена одного экземпляра*. В таблицу3 включить записи о программном обеспечении, разработанном фирмой Microsoft.
3. **Запрос7** на добавление в таблицу3 записей о программном обеспечении фирмы Sumantec.
4. **Запрос8** на удаление из таблицы3 записей с ценой одного экземпляра более 30\$.

##### Вариант 3. БД Склад

1. **Запрос5** на создание таблицы4 с полями *Название товара, Цена товара, Название фирмы-поставщика, Количество экземпляров товара на складе, Общая стоимость товара*. Таблица должна содержать записи о товарах, произведенных фирмой «Adidas».
2. **Запрос6** на добавление в таблицу4 товаров другой фирмы (на усмотрение разработчика).
3. **Запрос7** на обновление таблицы4. Если «количество экземпляров товара» более 100, «цена» уменьшается на 5%.

4. **Запрос8** на удаление из таблицы4 записей о товарах, общая стоимость которых более 2 000 000 руб.

##### Вариант 4. БД Платежи

1. **Запрос5** на создание таблицы4 с полями *Дата платежа, Название организации, Название банка, Сумма платежа, Пеня, К оплате*. Таблица должна содержать список проводок платежей *Беларусь* Банком.
2. **Запрос6** на добавление в таблицу4 платежей *Prior* Банка.
3. **Запрос7** на удаление из таблицы4 платежей, совершенных до 31.01.2006г.
4. **Запрос8** на обновление таблицы4 – если «сумма платежа» больше 200 000 руб., то уменьшить ее в 2 раза.

##### Вариант 5. БД Комплектующие к ПК

1. **Запрос5** на создание таблицы4 с полями *Порядковый номер; Наименование изделия; Наименование фирмы-изготовителя; Цена; Количество экземпляров; Общая стоимость*. Таблица должна содержать список комплектующих изделий для ПК фирмы INTEL.
2. **Запрос6** на добавление в таблицу4 комплектующих изделий фирмы AMD.
3. **Запрос7** на обновление таблицы4. Если «цена» ниже 40 у.е., увеличить «количество экземпляров» в 2 раза.
4. **Запрос8** на удаление из таблицы4 комплектующих, общая стоимость которых превышает 500у.е.

##### Вариант 6. БД Музыка

1. **Запрос5** на создание таблицы4 с полями *Название компакт-диска; Название произведения; Фамилия исполнителя; Длительность воспроизведения в минутах; Цена*. Таблица должна содержать список компакт-дисков исполнителя N, заданного разработчиком.
2. **Запрос6** на добавление в таблицу4 компакт-дисков другого исполнителя.
3. **Запрос7** на удаление из таблицы4 информации о компакт-дисках, цена которых выше определенной суммы.
4. **Запрос8** на обновление таблицы4. Произвести уценку всех компакт-дисков, уменьшив цену на 15%.

#### Вариант 7. БД Абитуриент

1. **Запрос5** на создание таблицы4 с полями *Фамилия, имя, отчество; Год рождения; Наименование предмета; Полученная оценка; Средний балл; Возраст абитуриента*. Таблица должна содержать результаты сдачи экзамена по Русскому языку.
2. **Запрос6** на добавление в таблицу4 информации о результатах экзамена по Математике.
3. **Запрос7** на удаление из таблицы4 информации об абитуриентах, имеющих средний балл меньше проходного балла. Проходной балл задает пользователь.
4. **Запрос8** на обновление таблицы4. Абитуриентам младше 18 лет, к *Среднему баллу аттестату* прибавить 10.

#### Вариант 8. БД Зарплата

1. **Запрос5** на создание таблицы3 с полями *название цеха, ФИО, домашний адрес, больничные дни, к выплате*. В таблицу занести записи по цеху, выбранному разработчиком.
2. **Запрос6** на добавление в таблицу3 сведений о работниках другого цеха.
3. **Запрос7** на удаление из таблицы3 записей о работниках, которые были на больничном менее 5 дней.
4. **Запрос8** на обновление таблицы3. Если работник проболел более 20 дней, назначить материальную помощь в размере 20% от начисляемого заработка и прибавить ее к графе *к выплате*.

#### Вариант 9. БД Телефон

1. **Запрос5** на создание таблицы4 с полями *Номер телефона, Фамилия владельца, Дата разговора, Название города, Код города*. Включить в создаваемую таблицу записи о разговорах с городом Москва.
2. **Запрос6** на добавление в таблицу4 записей о разговорах с Брестом.
3. **Запрос6** на удаление из таблицы4 записей о разговорах до 01.01.2007г.
4. **Запрос8** на обновление таблицы4. Код Москвы заменить на код 495.

#### Вариант 10. БД Отдел

1. **Запрос5** на создание таблицы3 с полями *Фамилия сотрудника, Дата рождения, Наименование Области, Месячный оклад*. Включить в создаваемую таблицу записи о сотрудниках, родившихся в Минской области.

2. **Запрос6** на добавление в таблицу 3 записей из о сотрудниках, родившихся в Брестской области.
3. **Запрос7** на удаление записей из таблицы3, касающихся сотрудников, старше 65 лет.
4. **Запрос8** на обновление таблицы 3 – увеличить месячный оклад на 5% сотрудникам отдела, месячный оклад которых менее 300 000 рублей.

#### Вариант 11. БД Квартира

1. **Запрос5** на создание таблицы4 с полями *Дата выставления на продажу; Название района; Количество комнат; Общая площадь в кв. метрах; Наличие телефона; Оценочная стоимость квартиры; Цена 1 кв. метра*. Таблица должна содержать записи о 2-комнатных квартирах.
2. **Запрос6** на добавление в таблицу4 информации о 3-комнатных квартирах.
3. **Запрос7** на обновление таблицы4. Если квартира выставлена на продажу более чем 2 года назад, «оценочную стоимость» снизить на 10%.
4. **Запрос8** на удаление из таблицы4 квартир без телефона.

#### Вариант 12. БД Справочник врача

1. **Запрос5** на создание таблицы4 с полями *Фамилия и инициалы; Возраст; Дата обращения к врачу; Домашний адрес; Диагноз*. Таблица должна содержать список пациентов с диагнозом «ОРЗ».
2. **Запрос6** на добавление в таблицу4 информации о пациентах с диагнозом «бронхит».
3. **Запрос7** на удаление из таблицы4 информации о пациентах, обратившихся к врачу до 31.12.06.
4. **Запрос8** на обновление таблицы2. Пациенту N изменить адрес по усмотрению разработчика.

#### Вариант 13. БД Почта

1. **Запрос5** на создание таблицы3 с полями *Дата отправки; Город; Фамилия получателя; Адрес отправителя; Фамилия отправителя; Вес почтового отправления в килограммах*. Таблица должна содержать список почтовых отправлений в г. Гродно.
2. **Запрос6** на добавление в таблицу3 информации о почтовых отправлениях в г. Брест.
3. **Запрос7** на удаление из таблицы3 информации об отправлениях до 1.01.2007.

4. **Запрос8** на обновление таблицы3. Если вес почтового отправления больше 15 кг, к «ФИО отправителя» добавить требование «ДОПЛАТИТЬ».

#### **Вариант 14. БД Автомобили**

1. **Запрос5** на создание таблицы4 с полями *Марка автомобиля; Название страны изготовителя; Год выпуска; Пробег в километрах; Стартовая цена; Возраст автомобиля*. Таблица должна содержать список автомобилей марки *BMW*.

2. **Запрос6** на добавление в таблицу4 информации об автомобилях марки, определенной пользователем.

3. **Запрос7** на удаление из таблицы4 информации об автомобилях, пробег в км которых более 30000 км.

4. **Запрос8** на обновление таблицы4. Если «возраст» автомобиля более 10 лет, *стартовая цена* снижается на 10%.

#### **Вариант 15. БД Учет компьютерных средств**

1. **Запрос5** на создание таблицы3 с полями *Наименование подразделения, Инвентарный № компьютера, Место расположения, Год ввода в эксплуатацию, Стоимость после переоценки*. Включить в создаваемую таблицу сведения о компьютерах, находящихся на балансе кафедры «Вычислительная техника».

2. **Запрос6** на добавление в таблицу3 сведений о компьютерах, закрепленных за отделом ЦИТ.

3. **Запрос7** на обновление таблицы3. Компьютеры старше 7 лет должны быть списаны, для этого в графу «Стоимость после переоценки» записать 0.

4. **Запрос8** на удаление из таблицы3 сведений о компьютерах, подлежащих списанию.

#### **Вариант 16. БД Успеваемость**

1. **Запрос5** на создание таблицы4 с полями *ФИО студента, Название предмета, Оценка, дата сдачи экзамена, Стипендия, Группа*. Включить в создаваемую таблицу, сведения об успеваемости по одному из предметов на выбор разработчика.

2. **Запрос6** на добавление в таблицу4 сведений об успеваемости по другому предмету.

3. **Запрос7** на обновление таблицы4. Если экзамен сдан после окончания сессии, в поле «Стипендия» занести «нет».

4. **Запрос8** на удаление из таблицы3 записей о студентах группы (номер группы указать самостоятельно).

#### **Вариант 17. БД Заказы**

1. **Запрос5** на создание таблицы4 с полями *№ заказа, Фирма-клиент, ФИО сотрудника, Дата размещения заказа, Срок выполнения заказа, Дата завершения работ, Вознаграждение сотрудника*. Включить в создаваемую таблицу информацию о заказах на работы, заключенные одной из фирм.

2. **Запрос6** на добавление в таблицу4 информации о заказах другой фирмы.

3. **Запрос7** на обновление таблицы4. Если работы были завершены раньше срока, увеличить «вознаграждение сотрудника» на 10%.

4. **Запрос8** на удаление из таблицы4 невыполненных заказов.

#### **Вариант 18. БД Реализация готовой продукции**

1. **Запрос5** на создание таблицы4 с полями *Код сделки, Дата сделки, Наименование Товара, Покупатель, Оптовая Цена, Количество, Сумма сделки, Телефон Покупателя*. Включить в создаваемую таблицу записи о сделках, проведенных в текущем месяце.

2. **Запрос6** на добавление в таблицу 4 записей о сделках, проведенных в предыдущем месяце.

3. **Запрос7** на обновление таблицы 4. Если *Количество* превышает 200 шт., то *Оптовую Цену* уменьшить на 5%.

4. **Запрос8** на удаление из таблицы 4 записей о сделках, сумма которых менее 1 000 000 рублей.

#### **Вариант 19. БД Транспорт**

1. **Запрос5** на создание таблицы4 с полями *Номер рейса, Фамилия водителя, Марка автомобиля, Вес груза в тоннах*. Включить в таблицу записи, касающиеся автомобилей марки «МАЗ».

2. **Запрос6** на добавление в таблицу4 записей об автомобилях марки «VOLVO» .

3. **Запрос7** на удаление из таблицы4 записей о перевозках до 31.12.2005г.

4. **Запрос8** на обновление таблицы4 – Для рейсов, вес груза которых более 100 000 т заменить «марку автомобиля» на *БелАЗ*.

#### **Вариант 20. БД Расписание**

1. **Запрос5** на создание таблицы3 с полями *№ поезда; Ф.И.О. начальника поезда; пункт отправления; пункт назначения, наличие вагона-ресторана, количество вагонов*. В таблицу включить информацию о поездах, следующих в Москву.

2. **Запрос6** на добавление в таблицу3 записей о поездах, следующих в Брест.
3. **Запрос7** на удаление из таблицы3 сведений о пассажирских поездах.
4. **Запрос8** на обновление таблицы3. Если в поезде есть вагон-ресторан, увеличить количество вагонов на 1.

#### **Вариант 21. БД Соревнования**

1. **Запрос5** на создание таблицы3 с полями *Место проведения, ФИО, Вид спорта, Спортивный разряд, Рост, Вес*. В таблицу занести протоколы соревнований, которые были проведены на стадионе «Динамо».
2. **Запрос6** на добавление в таблицу3 записей о соревнованиях, проведенных в спорткомплексе «Олимпийский».
3. **Запрос7** на удаление из таблицы3 записей. Удалите информацию о тех участниках, вес которых не соответствует норме. (*Норма = рост – 100*).
4. **Запрос8** на обновление таблицы3. В связи с переносом соревнований по легкой атлетике на стадион «Трактор», изменить соответствующие записи в таблице3.

#### **Вариант 22. БД Праздничное бюро**

1. **Запрос5** на создание таблицы4 с полями *Наименование мероприятия, ФИО сотрудника, место проведения, Дата проведения, Количество мест, Цена за 1 место*. Включить в создаваемую таблицу записи о корпоративных мероприятиях.
2. **Запрос6** на добавление в таблицу4 записей о фуршетах.
3. **Запрос7** на обновление таблицы4. Если количество мест больше 100, то цену за 1 место уменьшить на 15 %.
4. **Запрос8** на удаление из таблицы4 записей о мероприятиях, проведенных до 31.01.08г.

#### **Вариант 23. БД Составление учебного плана**

1. **Запрос5** на создание таблицы4 с полями *Название Предмета, Количество часов, ФИО преподавателя, Ученая степень*. Включить в таблицу записи о преподавателях, имеющих ученую степень – кандидат наук.
2. **Запрос6** на добавление в таблицу4 записей о преподавателях – докторов наук.
3. **Запрос7** на обновление таблицы4 – для дисциплин с формой контроля – экзамен, увеличить количество часов на 10%.

4. **Запрос8** на удаление из таблицы4 записей о дисциплинах, количество часов которых менее 34.

#### **Вариант 24. БД Туристическая фирма**

1. **Запрос5** на создание таблицы4 с полями *Страна, Название Отеля, ФИО клиента, Адрес, Цена*. В таблицу включить информацию о клиентах, проживающих в Минске.
2. **Запрос6** на добавление в таблицу4 записей о клиентах, проживающих в Витебске.
3. **Запрос7** на удаление из таблицы4 сведений о путевках в Турцию.
4. **Запрос8** на обновление таблицы4. В связи с увеличением курса евро, увеличить стоимость путевок в страны Евросоюза на 5%.

#### **Вариант 25. БД Грузовые перевозки**

1. **Запрос5** на создание таблицы4 с полями *Пункт назначения, Дата Отправки, ФИО водителя, Дата рождения, Премия в %*. В таблицу включить информацию о рейсах в Москву.
2. **Запрос6** на добавление в таблицу4 информации о рейсах в Варшаву.
3. **Запрос7** на удаление из таблицы4 сведений о тех рейсах, которые были организованы позднее 1.05.2008 г.
4. **Запрос8** на обновление таблицы4. Увеличить Премию на 10 % тем водителям, которым в текущем году исполняется 50 лет.

#### **Вариант 26. БД Прокат автомобилей**

1. **Запрос5** на создание таблицы4 с полями *Марка автомобиля, Стоимость проката, ФИО клиента, Длительность проката*. В таблицу включить информацию об автомобилях, стоимость проката которых не превышает 20 у.е.
2. **Запрос6** на добавление в таблицу4 информации об автомобилях, стоимость проката которых выше 50у.е.
3. **Запрос7** на удаление из таблицы4 сведений о прокате автомобилей марки *Opel*.
4. **Запрос8** на обновление таблицы4. Если Длительность проката превышает 7 дней, Стоимость проката уменьшить на 20 %.

## Лабораторная работа №5 СОЗДАНИЕ ПРОСТЫХ ОТЧЕТОВ

**Цель лабораторной работы:** приобрести навыки создания простых отчетов.

### Порядок выполнения лабораторной работы

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания простого отчета. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.

### 1. Краткие теоретические сведения

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.).

Отчет изначально разбит на разделы: заголовок, верхний колонтитул, область данных, нижний колонтитул, примечание. Заголовок выводится только в начале, а примечание только в конце. Колонтитулы появляются на каждой странице и содержат, как правило, дату и номер страницы. В области данных размещаются поля таблиц и запросов, на которых строится отчет, тексты подписей, линии, прямоугольники.

Перечислим средства, которые используются в *MS Access* для создания отчетов.

1. **Конструктор** – позволяет самостоятельно разрабатывать собственные отчеты с заданными свойствами.
2. **Мастер отчетов** – позволяет достаточно быстро создать отчет на основе выбранных полей.
3. **Автоотчет: в столбец** – автоматическое создание отчета с полями, расположенными в один столбец.
4. **Автоотчет: ленточный** – автоматическое создание ленточного отчета.
5. **Мастер диаграмм** – создание отчета, в котором данные отображаются в виде диаграмм.
6. **Почтовые наклейки**, позволяющие создать отчет для почтовых наклеек.

## 2. Пример создания отчета

### 2.1. Создание отчета с использованием мастера отчетов

Создать **Отчет1** для вывода информации о стипендии с указанием **Ф.И.О.** студента, **среднего балла**, **вида и размера стипендии**. Информация должна быть сгруппирована по месту рождения студентов.

Информация, которая должна войти в отчет, уже имеется в таблице **Тбл2-НазначениеСтипендии**. Поэтому для создания простого отчета на основе готовой таблицы проще всего использовать **Мастер отчетов**.

Для создания отчета выполните следующие действия.

1. В окне БД **Деканат** на вкладке **Отчеты** нажмите кнопку **Создать**.
2. В открывшемся диалоговом окне **Новый отчет** укажите **Мастер отчетов**, а в качестве источника данных выберите таблицу **Тбл2-НазначениеСтипендии** (рисунок 5.1).

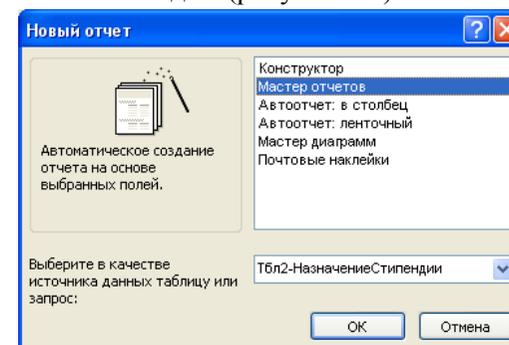


Рисунок 5.1 – Диалоговое окно Новый отчет

3. В окне **Создание отчетов** выберите все доступные поля и нажмите кнопку **Далее** (рисунок 5.2).
4. В соответствии с заданием на вопрос **Добавить уровни группировки?** выделите поле **МестоРождения** и перенесите его в правую часть окна с помощью кнопки . При этом поле, по которому осуществляется группировка, помещается в верхней части правого списка в отдельной рамке и выделяется синим цветом, как показано на рисунке 5.3. Нажмите кнопку **Далее**.
5. В следующем окне задайте порядок сортировки полей, как указано на рисунке 5.4. После чего нажмите кнопку **Далее**.
6. В следующих окнах выберите вид макета для отчета (*ступенчатый*), ориентацию (*книжная*) и стиль отчета (*деловой*).

7. В последнем окне задайте имя отчета **Отчет1** и нажмите кнопку **Готово**. Просмотрите созданный отчет.

6. Отредактируйте созданный отчет. Для этого перейдите в режим *Конструктор* (**Вид** → **Конструктор**). Измените название отчета на «Начисление стипендии». А также откорректируйте размеры полей таким образом, чтобы их содержимое полностью выводилось на экран. Окончательный вид отчета показан на рисунке 5.5.

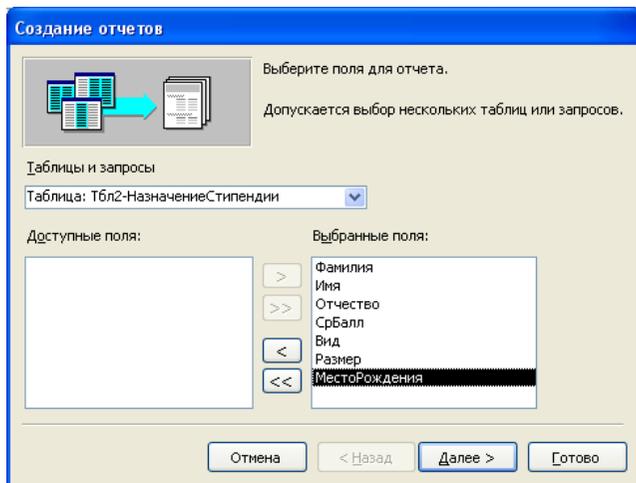


Рисунок 5.2 – Выбор полей, входящих в отчет

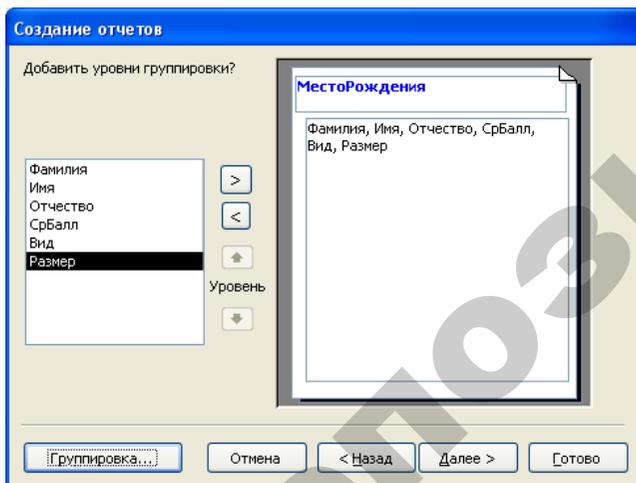


Рисунок 5.3 – Выбор уровня группировки

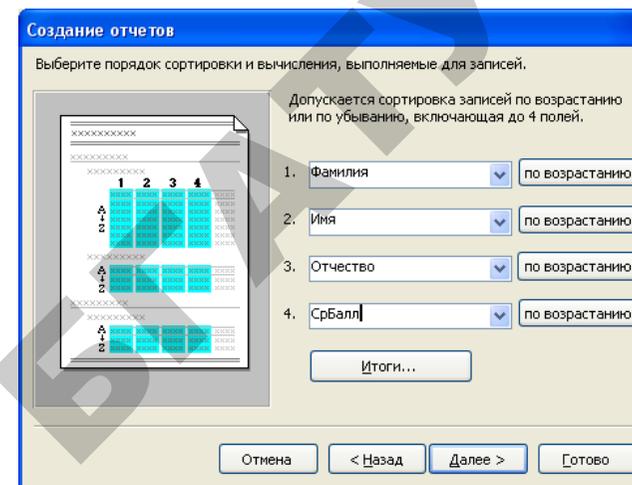


Рисунок 5.4 – Определение порядка сортировки полей

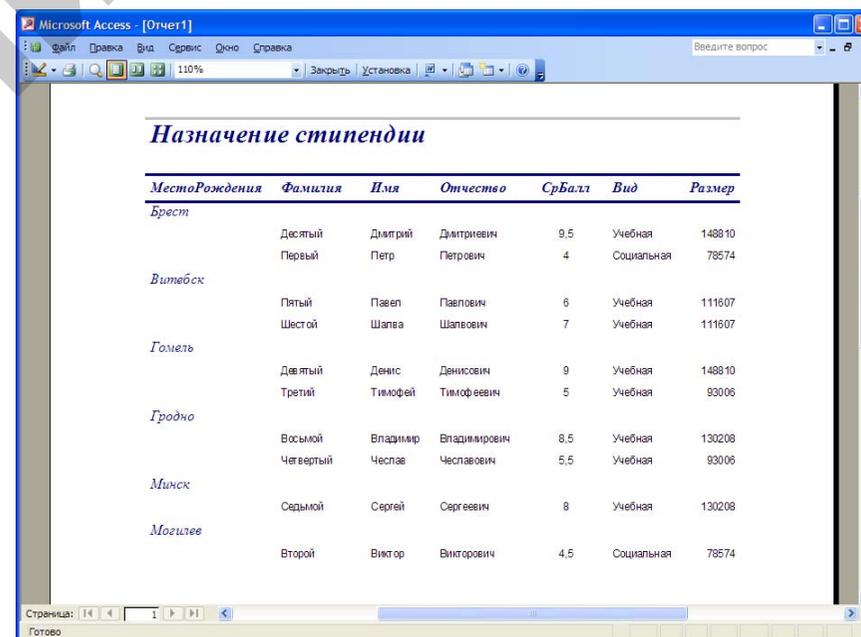


Рисунок 5.5 – Окончательный вид отчета **Отчет1**

## 2.2. Создание отчета с получением итогов

Создать **Отчет2** для вывода информации об успеваемости с указанием фамилии студента, названия дисциплины и полученной оценки. Информация должна быть сгруппирована по фамилии студента. Рассчитать средний балл каждого студента.

1. Для создания отчета с помощью Мастера отчетов выполните следующее. В окне **Новый отчет** выберите опцию **Мастер отчетов**, окно для выбора источника данных оставьте незаполненным.

2. Источниками данных для отчета являются таблицы **ЛичныеСведения** (поле *Фамилия*), **Дисциплины** (поле *НазваниеДисциплины*), **Успеваемость** (поле *Оценка*). Из раскрывающегося списка *Таблицы и запросы* выберите **Таблица: ЛичныеСведения**. Из панели *Доступные поля*: выделите и перенесите в панель *Выбранные поля*: поле **Фамилия**. Повторите эти действия для полей **НазваниеДисциплины** и **Оценка** из таблиц **Дисциплины** и **Успеваемость** соответственно. Окончательный вид диалогового окна показан на рисунке 5.6.

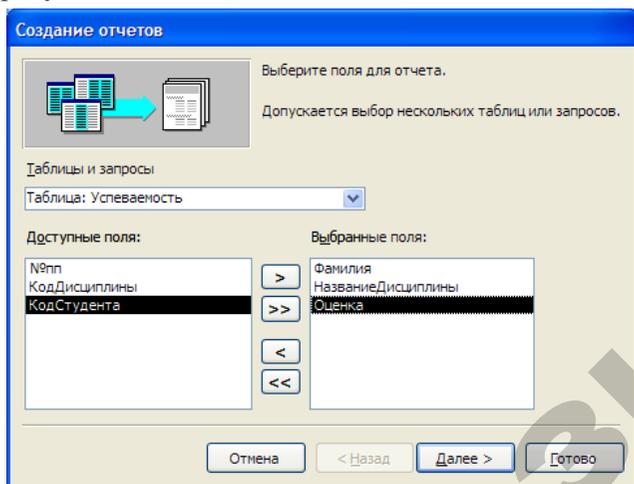


Рисунок 5.6 – Выбор полей для Отчета2

3. В следующем окне согласитесь с типом представления данных (рисунок 5.7). Нажмите кнопку **Далее**.

4. В следующем окне убедитесь, что группировка установлена по полю **Фамилия**.

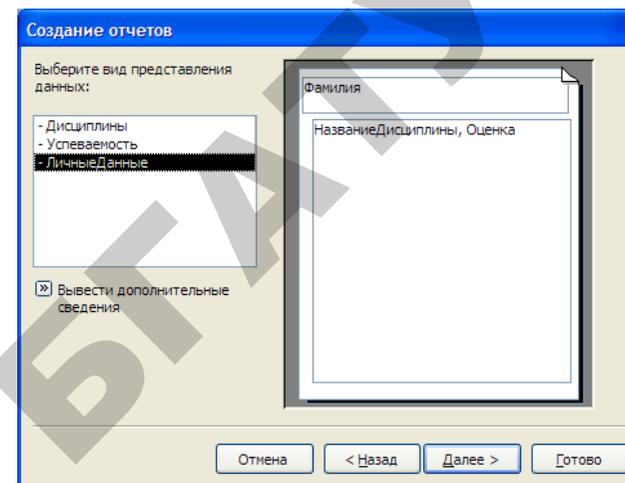


Рисунок 5.7 – Выбор типа представления данных

5. Задайте порядок сортировки для полей **НазваниеДисциплины** и **Оценка**. Для вычисления среднего балла каждого студента нажмите на кнопку **Итоги** (рисунок 5.8).

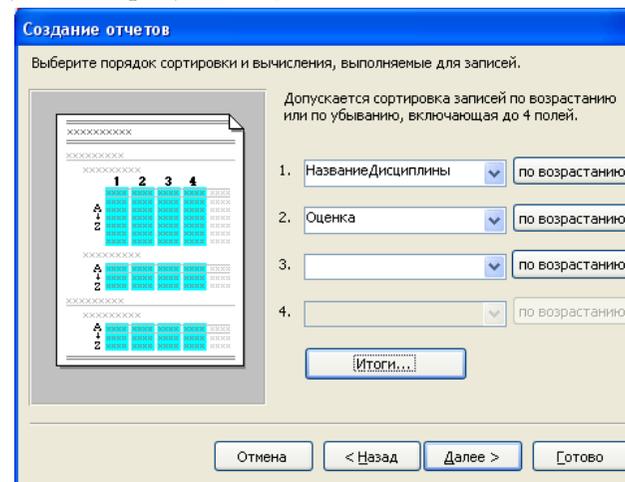


Рисунок 5.8 – Определение порядка сортировки полей в **Отчете2**

6. Откроется диалоговое окно **Итоги**, в котором выберите функцию **Avg** для вычисления среднего балла для поля **Оценка** (рисунок 5.9).

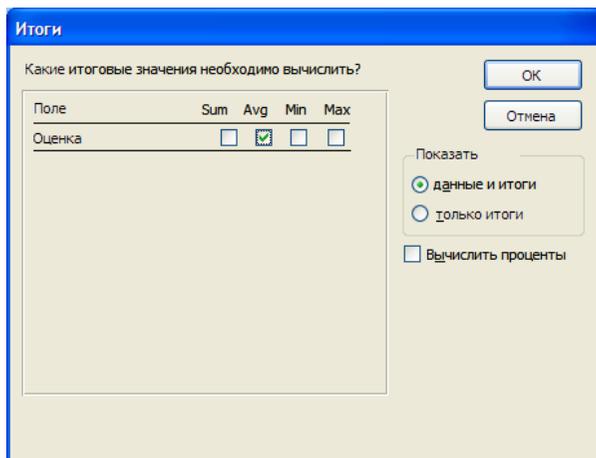


Рисунок 5.8 – Вычисление итогов для отчета

7. В следующих окнах выберите вид макета для отчета, ориентацию и стиль отчета по своему усмотрению.

8. В последнем окне задайте имя **Отчет2** и нажмите кнопку **Готово**.

9. В режиме *Конструктор* отредактируйте **Отчет2** таким образом, чтобы он принял вид, как показано на рисунке 5.10.

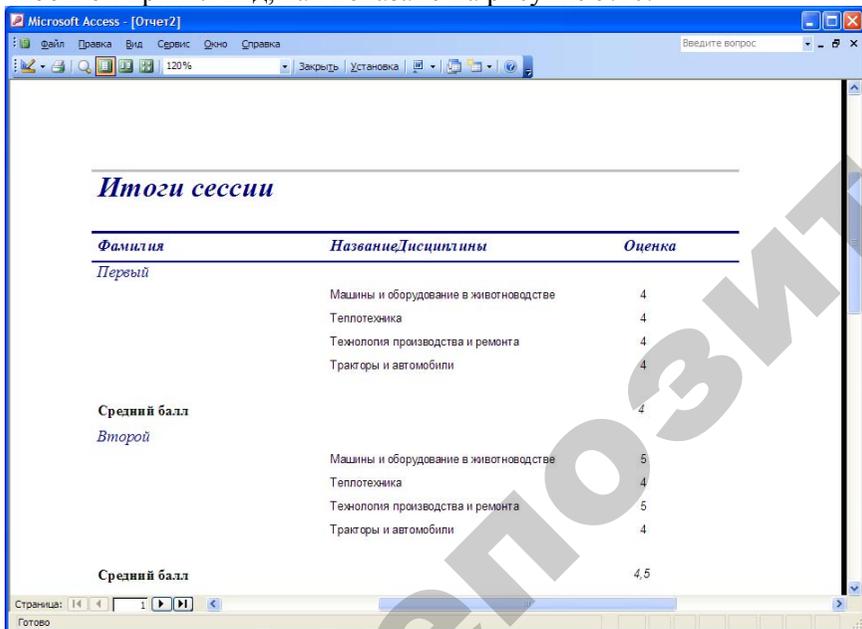


Рисунок 5.10 – Окончательный вид **Отчета2**

### 3. Вопросы для самоконтроля

1. Для чего служат отчеты?
2. Чем отчеты отличаются от форм?
3. Какие разделы содержит простой отчет?
4. Что обычно содержат колоннитулы?
5. Что размещается в области данных?
7. Какие средства используются в *MS Access* для создания отчетов?

### 4. Варианты индивидуальных заданий

#### Вариант 1. БД Факультет

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по кафедрам.

#### Вариант 2. БД Программы

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по фирмам изготовителям.

#### Вариант 3. БД Склад

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по фирмам-поставщикам.

#### Вариант 4. БД Платежи

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по названию организации.

#### Вариант 5. БД Комплектующие к ПК

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по наименованиям изделий.

#### Вариант 6. БД Музыка

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по названиям компакт-дисков.

#### Вариант 7. БД Абитуриент

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по наименованию предмета.

#### Вариант 8. БД Зарплата

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по названию цеха.

#### Вариант 9. БД Телефон

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по городам.

**Вариант 10. БД Отдел**

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по областям.

**Вариант 11. БД Квартира**

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по количеству комнат.

**Вариант 12. БД Справочник врача**

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по диагнозу.

**Вариант 13. БД Почта**

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по городам.

**Вариант 14. БД Автомобили**

Создать **Отчет1**, включающий поля запроса1. Выполнить группировку по году выпуска.

**Вариант 15. БД Учет компьютерных средств**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по году ввода в эксплуатацию.

**Вариант 16. БД Успеваемость**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по предметам.

**Вариант 17. БД Заказы**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по фирмам-клиентам.

**Вариант 18. БД Реализация готовой продукции**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по дате сделки.

**Вариант 19. БД Транспорт**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по маркам автомобилей.

**Вариант 20. БД Расписание**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по пункту назначения.

**Вариант 21. БД Соревнования**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по виду спорта.

**Вариант 22. БД Праздничное бюро**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по месту проведения.

**Вариант 23. БД Составление учебного плана**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по названию предмета.

**Вариант 24. БД Туристическая фирма**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по странам.

**Вариант 25. БД Грузовые перевозки**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по пункту отправления.

**Вариант 26. БД Прокат автомобилей**

Сформировать **Отчет1**, включающий поля запроса1. Выполнить группировку по марке автомобиля.

## Лабораторная работа №6 РАЗРАБОТКА СЛОЖНОЙ ФОРМЫ

**Цель лабораторной работы:** приобрести навыки разработки сложной формы.

### Порядок выполнения лабораторной работы

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания сложной формы. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.

### 1. Краткие теоретические сведения

Форма используется не только для ввода, но и для отображения данных. Источником записей формы являются поля в базовых таблицах и запросах. Связь между формой и ее источником записей создается при помощи графических объектов, которые называют элементами управления.

*Элемент управления* – объект графического интерфейса, позволяющий пользователям управлять приложением. Элементы управления используются для отображения данных или параметров, для выполнения действий, либо для упрощения работы с интерфейсом пользователя.

Элементы управления выводятся на экран с использованием команды **Панель элементов** меню **Вид**, либо с помощью кнопки  на панели инструментов **Конструктор форм**. Панель элементов управления показана на рисунке 6.1.

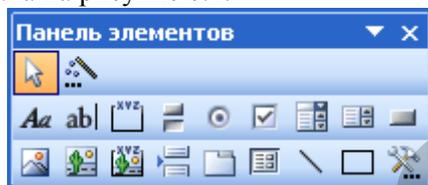


Рисунок 6.1 – Панель элементов управления

Перечислим наиболее часто используемые элементы управления.

1.  (*Надпись*). Служит для ввода и отображения текста (название формы, надпись над строкой ввода и т. д.).
2.  (*Поле*). Позволяет размещать в форме текстовую надпись и поле для ввода или вывода данных.

3.  (*Кнопка*). Используется для вызова макросов, процедур или выполнения какого-либо действия. На кнопке можно поместить текст или рисунок.
4.  (*Флажок*). Обычно используется для представления данных логического типа.
5.  (*Переключатель*). Переключатель, значение которого может зависеть от значений других переключателей. Этот компонент еще называют "зависимым переключателем". Часто используется для выбора одного значения поля из фиксированного списка.
6.  (*Группа переключателей*). Служит для создания блока переключателей, зависящих друг от друга. То есть, например списка значений, из которых можно выбрать только одно.
7.  (*Список*). Может содержать фиксированный набор значений или значения из заданного поля одной из таблиц. Позволяет не вводить данные, а выбирать их из списка.
8.  (*Поле со списком*). Представляет собой комбинацию элементов *Поле* и *Список*. Применяется так же, как и список, но занимает меньше места в форме, поскольку список открывается только после щелчка на раскрывающей кнопке.
9.  (*Рисунок*). Визуальный элемент, предназначенный для отображения графической информации. Он не может быть связан с каким-либо полем таблицы, и чаще всего используется для отображения на форме логотипа фирмы или любой другой статичной графической информации. При добавлении этого визуального элемента на форму или в отчет автоматически выводится диалог открытия файла, в котором необходимо указать путь к файлу с графической информацией.
10.  (*Подчиненная форма/отчет*). Позволяет отобразить еще одну форму или отчет в основном объекте.

Разработчик формы может перемещать элементы управления, изменять их размеры и задавать свойства шрифта для элементов. В форму можно также добавлять элементы управления для отображения вычисляемых значений, итогов, текущей даты, времени и других полезных сведений. При создании формы вручную элементы

управления размещают на ней так, как удобно пользователю. Созданные элементы управления формы выравнивают с помощью команды «Формат – Выровнять».

### 1.1. Создание форм

Как и другие объекты *Microsoft Access*, формы можно создавать вручную или автоматически, причем несколькими способами. В отличие от таблиц и запросов формы состоят из многочисленных элементов управления, и от того, насколько аккуратно эти элементы расположены на экране, зависит внешний вид формы. Автоматические средства позволяют создавать аккуратные формы и не задают пользователю лишних вопросов. Начинать работу лучше с них.

Быстро создать форму помогают команда «Автоформа» или мастер. С помощью автоформ можно создавать формы, в которых выводятся все поля и записи базовой таблицы или запроса. Мастер выводит на экран вопросы и создает форму на основании ответов пользователя. После этого пользователь имеет возможность доработать и изменить форму в режиме конструктора.

Структуру формы составляют ее разделы, а разделы содержат элементы управления.

### 1.2. Разделы формы

Самый простой способ познакомиться с разделами формы состоит в том, чтобы взять готовую форму, например созданную с помощью Мастера, и посмотреть ее структуру в режиме Конструктора. Обратите внимание на то, что рядом с ней открывается панель элементов, содержащая заготовки и инструменты для создания элементов управления формы. Размеры разделов и размеры рабочего поля формы можно изменять с помощью мыши. В структуре формы четко видны три раздела: раздел заголовка формы, область данных и раздел примечания формы. Все, что содержится в области данных, является элементами управления.

### 1.3. Создание надписей

Редактирование форм состоит в создании новых или изменении имеющихся элементов управления, а также в изменении их взаимного расположения.

При рассмотрении приемов создания новых элементов управления воспользуемся тем фактом, что Мастер, создавший форму, не заполнил ее раздел заголовка. Перетащив вниз разделительную границу между заголовком и областью данных, мы можем освободить сверху достаточно места для создания крупной надписи. На панели

элементов существует специальный элемент управления для создания заголовков, который называется «Надпись». Щелкнув по нему, а потом по форме, мы получаем текстовую рамку, в которую можно вводить произвольный текст. При вводе текста не надо заботиться о его форматировании. Неважно, как он выглядит и где расположен. Закончив ввод, надо нажать клавишу «ENTER», после чего можно приступить к оформлению текста. Для форматирования элемента управления его надо сначала выделить. Для этого служит инструмент «Выбор объектов». При выделении элемента управления вокруг него образуется рамка с восемью маркерами (по углам и по центрам сторон рамки). Рамку можно растягивать или сжимать методом перетаскивания границ. При наведении на маркер указатель мыши меняет форму, принимая вид открытой ладони. В этот момент рамку можно перемещать. Особую роль играет левый верхний маркер рамки. При наведении на него указатель мыши принимает форму указательного пальца. В этом случае надпись элемента и его поле можно перемещать независимо друг от друга. Когда объект выделен, можно изменять параметры шрифта, метод выравнивания текста и другие элементы форматирования. Это выполняют обычными средствами форматирования, доступными через соответствующую панель инструментов *Microsoft Access*. Если щелкнуть на выделенном элементе правой кнопкой мыши, откроется его контекстное меню, в котором имеются дополнительные возможности изменения оформления.

### 1.4. Создание и редактирование связанных полей.

Элемент управления «Надпись» называют свободным полем. Текст, введенный в него, остается неизменным независимо от того, какую запись в этот момент просматривают в форме. Совсем иначе обстоит дело с элементами управления, в которых отображается содержимое полей таблицы. Такие элементы управления называют связанными полями. Для их создания служит элемент «Поле» на панели инструментов. При создании связанного поля вместе с ним одновременно образуется еще один элемент управления – «Присоединенная надпись». Она перемещается вместе со связанным полем и образует с ним единое целое. Оторвать поле от присоединенной надписи позволяет уже упомянутый маркер, расположенный в левом верхнем углу. При наведении на него указатель мыши принимает форму указательного пальца. В этот момент связанное поле можно оторвать от присоединенной надписи и перемещать отдельно. Перемещать элементы управления и изменять их размеры с помо-

щью мыши не слишком удобно. Гораздо удобнее использовать для этой цели курсорные клавиши в комбинации с клавишами «SHIFT» или «CTRL». В первом случае происходит изменение размеров элемента управления, а во втором — изменение его расположения. Чтобы элементы управления располагались в форме ровными рядами, существуют специальные команды выравнивания. Сначала надо выделить группу элементов управления с помощью инструмента «Выбор объектов» (группа выбирается при нажатой клавише SHIFT), а потом дать команду «Формат — Выровнять» и выбрать метод выравнивания.

## 2. Пример создания сложной формы

### 2.1. Создание многотабличной формы

Создать форму **ИтогиСессии** на базе таблицы **Дисциплины**, содержащую подчиненную форму, которая отображает оценки студентов по выбранному предмету.

Формы, которые были созданы в Лабораторной работе № 2, включали поля из одной таблицы. На практике часто возникает необходимость просмотра и редактирования данных из нескольких таблиц одновременно. В этих случаях для отображения записей основной и подчиненной таблиц можно использовать многотабличную иерархическую форму. Иерархической является форма, содержащая главную форму и одну или несколько подчиненных форм.

Создадим многотабличную форму с помощью Мастера форм.

1. На вкладке **Формы** нажмите кнопку **Создать**. В диалоговом окне **Новая форма** выберите опцию **Мастер форм**, источник данных не выбирать. Нажмите **ОК**.
2. В открывшемся окне **Создание форм** выберите из списка **Таблицы и запросы** главную таблицу **Дисциплины** и поместите в список **Выбранные поля** поле **НазваниеДисциплины**.
3. Выберите в окне **Создание форм** из списка **Таблицы и запросы** таблицу **ЛичныеДанные** и поместите в список **Выбранные поля**; поля **Фамилия, Имя, Отчество**.
4. Далее из таблицы **Успеваемость** добавьте в список **Выбранные поля** поле **Оценка**. Окно **Создание форм** должно иметь вид, как на рисунке 6.2.

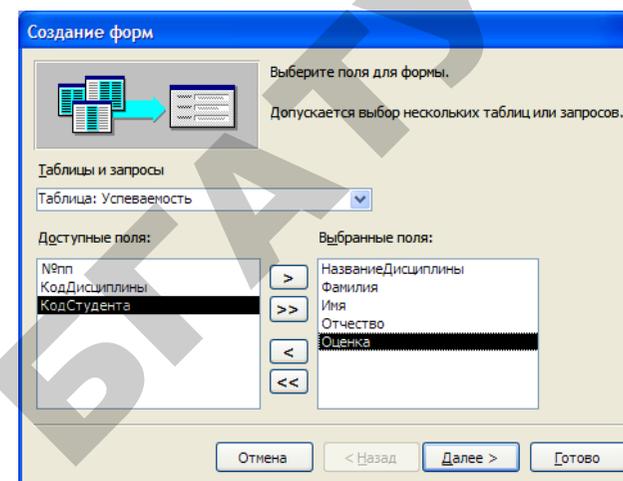


Рисунок 6.2 – Выбор полей для сложной формы

5. Затем нажмите кнопку **Далее**, в результате откроется следующее окно диалога **Создание форм** (рисунок 6.3). В этом окне определяем главную и подчиненную таблицу. По условию, главной таблицей является таблица **Дисциплины**, подчиненные таблицы – **Успеваемость** и **ЛичныеДанные**. Нажмите кнопку **Далее**.

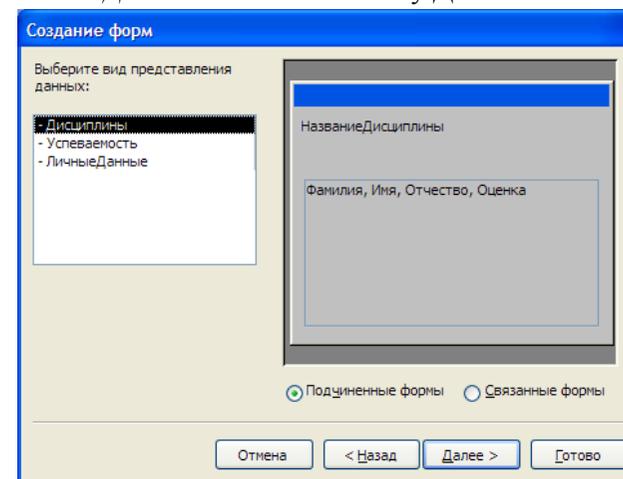


Рисунок 6.3 – Выбор главной формы

6. В следующем окне выберите внешний вид подчиненной формы, например, **Табличный**. Нажмите **Далее**.

7. В последующих диалоговых окнах выберите стиль, задайте имена форм: форма – **ИтогиСессии**; подчиненная форма – **Полученные оценки подчиненная форма**.

8. Нажмите кнопку **Готово**. Созданная мастером форма выводится на экран.

9. Перейдите в режим *Конструктор*, измените размеры для главной и подчиненной форм, добавьте поясняющие надписи. Окончательный вид формы **ИтогиСессии** представлен на рисунке 6.4.

	Фамилия	Имя	Отчество	Оценка
▶	Первый	Петр	Петрович	4
	Второй	Виктор	Викторович	4
	Третий	Тимофей	Тимофеевич	5
	Четвертый	Чеслав	Чеславович	5
	Пятый	Павел	Павлович	6
	Шестой	Шалва	Шалвович	7
	Седьмой	Сергей	Сергеевич	8
	Восьмой	Владимир	Владимирович	8
	Девятый	Денис	Денисович	9
	Десятый	Дмитрий	Дмитриевич	9
*				

Рисунок 6.4 – Форма **ИтогиСессии**

## 2.2. Создание сложной формы

Создать форму **Информация о студентах**, в которой выводится информация о личных данных: код студента, Ф.И.О., дата рождения, место рождения, рост, вес, наличие общежития. Форма должна содержать две подчиненных формы: **Успеваемость**, в которой выводится название дисциплин и оценки и **Стипендия**, в которой выводится средний балл по дисциплинам, вид и размер стипендии.

Перед разработкой сложной формы следует определить, какая информация потребуется и в каком порядке создавать части формы. Планирование сократит общую продолжительность разработки.

В форме **Информация о студентах**, которую мы будем создавать, главная форма будет подготовлена на базе таблицы **Личные данные**, подчиненная форма **Стипендия** – базе таблицы **НазначениеСтипендии**, а подчиненная форма **Успеваемость** – на базе специально созданного запроса **Успеваемость**.

Создадим запрос **Успеваемость**, на котором будет основана соответствующая подчиненная форма.

1. В диалоговое окно конструктора запроса необходимо добавить таблицы **Дисциплины**, **Успеваемость** и **ЛичныеДанные**.

2. Поместить в бланк запроса следующие поля: **НазваниеДисциплины**, **Оценка** и **КодСтудента**.

3. Выполнить запрос нажатием клавиши меню – **Запуск**. Результат выполнения запроса приведен на рисунке 6.5.

4. Сохраните запрос под именем **ЗпрУспеваемость** и убедитесь, что он появился в объектах **Запросы**.

НазваниеДисциплины	Оценка	КодСтудента
Теплотехника	4	74060301
Тракторы и автомобили	4	74060301
Машины и оборудование в животноводстве	4	74060301
▶ Технология производства и ремонта	4	74060301

Рисунок 6.5 – Результат выполнения запроса **ЗпрУспеваемость**

Создадим главную форму **Информация о студентах**.

1. На вкладке **Таблицы** выберите таблицу **Личные данные**.

2. На панели инструментов нажмите кнопку – **Новый объект:автоформа**. Мастер форм создаст простейшую форму из одного столбца, содержащего все поля из таблицы **Личные данные**, как показано на рисунке 6.6.

3. Сохраните форму под именем **Информация о студентах**.

Личные Данные

КодСтудента: 74060301

Фамилия: Первый

Имя: Петр

Отчество: Петрович

ДатаРождения: 01.01.1988

МестоРождения: Брест

Рост: 171

Вес: 71

Телефон: 2010101

Общезнание:

Запись: 1 из 10

Рисунок 6.6. – Форма, созданная из таблицы **Личные данные**

Создадим подчиненную форму **Стипендия**.

1. На вкладке **Формы** нажмите кнопку **Создать**. Дважды щелкните на строке Мастер форм.
2. В окне **Создание форм** из списка **Таблицы и запросы** выберите таблицу **Тбл2-НазначениеСтипендии**.
4. Выберите поля: **СрБалл**, **Вид** и **Размер**. Нажмите кнопку **Далее**.
5. Предлагаемый макет в один столбец подходит для формы. Нажмите кнопку **Далее**.
6. Выберите стиль **Стандартный** и нажмите кнопку **Далее**. Назовите форму **Стипендия подформа** и нажмите кнопку **Готово**. Она должна выглядеть так, как показано на рисунке 6.7.
7. Закройте форму **Стипендия подформа**.

Стипендия подформа

СрБалл: 4

Вид: Социальная

Размер: 78574

Запись: 1 из 10

Рисунок 6.7 – Подчиненная форма **Стипендия**

Создадим подчиненную форму **Успеваемость**.

1. На вкладке **Формы** нажмите кнопку **Создать**. Дважды щелкните на строке Мастер форм.
2. В окне **Создание форм** из списка **Таблицы и запросы** выберите запрос **ЗпрУспеваемость**.
3. Выберите поля: **НазваниеДисциплины**, **Оценка** и **КодСтудента**. Нажмите кнопку **Далее**.
4. Выберите вид представления данных: **Успеваемость** и нажмите кнопку **Далее**.
7. Выберите внешний вид формы **Табличный** и нажмите кнопку **Далее**.
8. Выберите стиль **Стандартный** и нажмите кнопку **Далее**. Назовите форму **Успеваемость подформа** и нажмите кнопку **Готово**. Поле **КодСтудента** служит только для связи с главной формой, поэтому его можно не показывать. Форма должна выглядеть так, как показано на рисунке 6.8.

Успеваемость подформа

НазваниеДисциплины	Оценка
Теплотехника	4
Теплотехника	5
Теплотехника	5
Теплотехника	6

Запись: 5 из 40

Рисунок 6.8 – Подчиненная форма **Успеваемость**

#### Включение подчиненных форм в главную форму

Теперь, когда созданы отдельно главная и две подчиненные формы, можно их объединить в одну сложную форму.

1. Откройте форму **Информация о студентах** в режиме конструктора.
2. Разверните окно формы.
3. Перетащите правый нижний угол формы вправо и вниз, чтобы увеличить размер формы (для размещения подчиненных форм).
4. Как видно из рисунка 6.6, поля, в которых появляются данные, слишком велики и имеют разную длину. Изменим размеры всех полей одновременно. Для этого, удерживая клавишу **Shift**, выделите все поля, кроме поля **Флажок**.
5. В меню **Формат** выберите **Размер** → **по самому узкому**. Форма примет вид, показанный на рисунке 6.9. При необходимости можно

аналогичным приемом еще раз выделить все поля и с помощью маркеров установить желаемые размеры полей.

6. Щелкните в окне БД, чтобы оно оказалось сверху окна формы (возможно, придется изменить размеры окна БД).

7. В окне БД на вкладке **Формы** наведите указатель на имя формы **Успеваемость подформа**.

8. Перетащите эту форму в окно главной формы. Белый прямоугольник, изображающий подчиненную форму **Успеваемость подформа**, называется *элементом управления подчиненной формы/отчета*.

9. Переместите белый прямоугольник, изображающий подчиненную форму, справа от полей главной формы.

10. Щелкните в окне БД.

11. Перетащите форму **Стипендия подформа** в окно главной формы.

12. Поместите ее под подчиненной формой **Успеваемость подформа**.

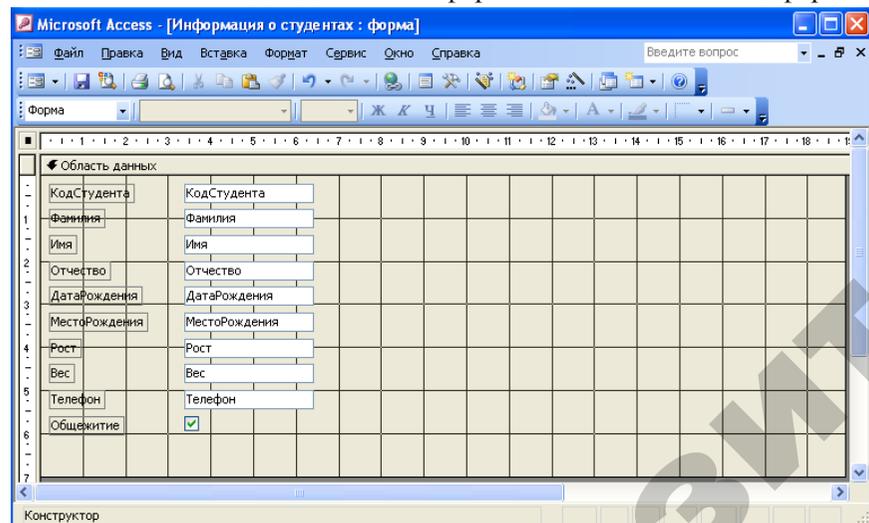


Рисунок 6.9 – Главная форма **Информация о студентах**

13. Выделите правой кнопкой подформу **Стипендия подформа** так, чтобы указатель принял форму ладони и в подменю выберите **Свойства**. Появится окно диалога **Подчиненная форма/отчет: Стипендия подформа**, показанная на рисунке 6.10.

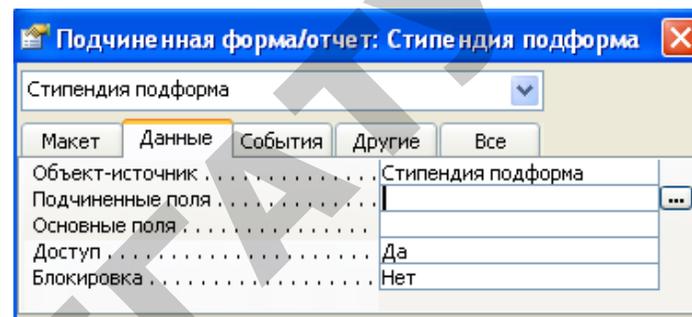


Рисунок 6.10 – Окно диалога

Это окно используется для установления связи между подчиненной формой и главной формой при помощи связывания подчиненного поля (в подчиненной форме) и основного поля (в главной форме).

14. Щелкните в поле **Подчиненные поля**.

15. Затем нажмите кнопку построителя (маленькую кнопку с многоточием, которая появилась справа от поля **Подчиненные поля**, а не кнопку на панели инструментов). Появится окно диалога **Связь с полями подчиненной формы**, в котором Access самостоятельно выбрал подходящие для связи поля – в данном случае поле **Фамилия** в обеих формах. Нажмите **ОК**.

16. Закройте окно диалога **Подчиненная форма/отчет: Стипендия подформа**.

17. В подчиненной форме **Успеваемость подформа** щелкните по присоединенной надписи **Успеваемость подформа** и измените текст на **Успеваемость** жирным шрифтом.

18. В подчиненной форме **Стипендия подформа** щелкните по присоединенной надписи **Стипендия подформа** и измените текст на **Стипендия** жирным шрифтом.

19. Нанесем на форму надпись. Для этого перейдите в меню **Вид Microsoft Access** и отметьте пункт **Заголовок/примечание формы**. В режиме конструктора на форме появятся два раза дела: **Заголовок формы** и **Примечание формы**.

20. Пользуясь указателем мыши, раздел **Примечание формы** можно убрать.

21. Откройте **Панель элементов** и элемент **Надпись** поместите в середине раздела **Заголовок формы**.

22. Наберите текст **Информация о студентах**. Отредактируйте ее местоположение, размер шрифта и цвет.

23. Переключаясь между режимами **Конструктор** и **Режим формы**, добейтесь, чтобы главная форма выглядела так, как она показана на рисунке 6.11.

**Информация о студентах**

КодСтудента: 74060301

Фамилия: Первый  
Имя: Петр  
Отчество: Петрович  
ДатаРождения: 01.01.1988  
МестоРождения: Брест  
Рост: 171  
Вес: 71  
Телефон: 2010101  
Общежитие:

**Успешность**

НазваниеДисциплины	Оценка
Теплотехника	4
Тракторы и автомобили	4
Машины и оборудование в животноводстве	4
Технология производства и ремонта	4

Запись: 1 из 4

**Стипендия**

СрБалл: 4  
Вид: Социальная  
Размер: 78574

Запись: 1 из 1

Рисунок 6.11 – Форма **Информация о студентах**

24. Убедитесь в правильном функционировании формы и сохраните ее под именем **Информация о студентах**.

### 3. Вопросы для самоконтроля

1. Что такое элемент управления?
2. Для чего используются элементы управления?
3. Как вывести на экран **Панель элементов**?
4. Перечислите наиболее часто используемые элементы управления и укажите их назначение.
5. Что такое свободное и связанное поле?
6. Как создаются и редактируются связанные поля?

## 4. Варианты индивидуальных заданий

### Вариант 1. БД Факультет

Создать форму **Информация о сотрудниках**, в которой выводится информация о личных данных: порядковый номер, фамилия, дата рождения, ученая степень, стаж работы, номер телефона. Форма должна содержать подчиненную форму **Кафедра**, в которой выводится название кафедры.

### Вариант 2. БД Программы

Создать форму **Информация о программных продуктах**, в которой выводится следующая информация: порядковый номер, шифр программного продукта, наименование продукта, тип, цена одного экземпляра, количество экземпляров. Форма должна содержать подчиненную форму **Фирма**, в которой выводится название фирмы.

### Вариант 3. БД Склад

Создать форму **Информация о товарах**, в которой выводится следующая информация: порядковый номер, цена товара, количество экземпляров. Форма должна содержать подчиненную форму **Товар**, в которой выводится название товара и производитель и подчиненную форму **Фирма**, в которой выводится название фирмы.

### Вариант 4. БД Платежи

Создать форму **Информация о платежах**, в которой выводится следующая информация: порядковый номер, дата платежа, сумма платежа, пеня. Форма должна содержать подчиненную форму **Организация**, в которой выводится название организации и подчиненную форму **Банк**, в которой выводится название банка.

### Вариант 5. БД Комплектующие к ПК

Создать форму **Информация о комплектующих**, в которой выводится следующая информация: порядковый номер, цена, количество. Форма должна содержать подчиненную форму **Изделия**, в которой выводится наименование изделия и подчиненную форму **Фирма**, в которой выводится наименование фирмы.

### Вариант 6. БД Музыка

Создать форму **Информация о компакт-дисках**, в которой выводится следующая информация: порядковый номер, название, цена. Форма должна содержать подчиненную форму **Содержание**, в которой выводится название произведения, длительность воспроизведения и подчиненную форму **Исполнитель**, в которой выводится фамилия исполнителя.

#### **Вариант 7. БД Абитуриент**

Создать форму **Информация об абитуриентах**, в которой выводится следующая информация: порядковый номер, оценка. Форма должна содержать подчиненную форму **Абитуриент**, в которой выводится Ф.И.О., год рождения, средний балл аттестата и подчиненную форму **Предмет**, в которой выводится наименование предмета.

#### **Вариант 8. БД Зарплата**

Создать форму **Информация о зарплате**, в которой выводится следующая информация: Ф.И.О., домашний адрес, телефон, паспортные данные. Форма должна содержать подчиненную форму **Ведомость**, в которой выводится название цеха, больничные дни, начисляемый заработок, величина удержаний, дата выдачи.

#### **Вариант 9. БД Телефон**

Создать форму **Информация о разговорах**, в которой выводится следующая информация: порядковый номер, номер телефона, дата разговора, время начала разговора, время окончания разговора. Форма должна содержать подчиненную форму **Телефоны**, в которой выводится фамилия владельца, адрес владельца и подчиненную форму **Город**, в которой выводится название города.

#### **Вариант 10. БД Отдел**

Создать форму **Информация о сотрудниках**, в которой выводится следующая информация: порядковый номер, фамилия сотрудника, дата рождения, оклад, премия в процентах. Форма должна содержать подчиненную форму **Область**, в которой выводится наименование области и населенный пункт.

#### **Вариант 11. БД Квартира**

Создать форму **Информация о квартирах**, в которой выводится следующая информация: порядковый номер, дата выставления на продажу, количество комнат, общая площадь, наличие телефона, оценочная стоимость квартиры. Форма должна содержать подчиненную форму **Район**, в которой выводится название района.

#### **Вариант 12. БД Справочник врача**

Создать форму **Информация об пациентах**, в которой выводится следующая информация: порядковый номер, дата обращения к врачу. Форма должна содержать подчиненную форму **Пациент**, в которой выводится Ф.И.О., год рождения, домашний адрес и подчиненную форму **Диагноз**, в которой выводится диагноз.

#### **Вариант 13. БД Почта**

Создать форму **Информация о почтовых отправлениях**, в которой выводится следующая информация: порядковый номер, дата отправки, адрес получателя, фамилия получателя, адрес отправителя, фамилия отправителя, вес. Форма должна содержать подчиненную форму **Город**, в которой выводится название города.

#### **Вариант 14. БД Автомобили.**

Создать форму **Информация об автомобилях**, в которой выводится следующая информация: порядковый номер, год выпуска, пробег, стартовая цена. Форма должна содержать подчиненную форму **Марка**, в которой выводится марка автомобиля и подчиненную форму **Страна**, в которой выводится название страны изготовителя.

#### **Вариант 15. БД Учет компьютерных средств.**

Создать форму **Информация о компьютерах**, в которой выводится следующая информация: инвентарный № компьютера, наименование компьютера, место расположения (№ ауд.), год ввода в эксплуатацию, стоимость. Форма должна содержать подчиненную форму **Подразделение**, в которой выводится наименование подразделения.

#### **Вариант 16. БД Успеваемость.**

Создать форму **Информация об успеваемости**, в которой выводится следующая информация: код студента, Ф.И.О., дата рождения, год поступления, группа, стипендия. Форма должна содержать подчиненную форму **Оценка**, в которой выводится оценка, дата сдачи экзамена и подчиненную форму **Предмет**, в которой выводится наименование предмета.

#### **Вариант 17. БД Заказы.**

Создать форму **Информация о заказах**, в которой выводится следующая информация: № заказа, дата размещения заказа, срок выполнения заказа, дата завершения работ, стоимость, описание работы. Форма должна содержать подчиненную форму **Клиент**, в которой выводится название фирмы, форма оплаты, номер телефона и подчиненную форму **Сотрудник**, в которой выводится Ф.И.О. сотрудника, вознаграждение, контактный телефон.

#### **Вариант 18. БД Реализация готовой продукции.**

Создать форму **Информация о сделках**, в которой выводится следующая информация: дата сделки, количество товара. Форма должна содержать подчиненную форму **Товар**, в которой выводится

наименование товара, оптовая цена, описание товара и подчиненную форму **Покупатель**, в которой выводится наименование покупателя, телефон, район, адрес.

#### **Вариант 19. БД Транспорт.**

Создать форму **Информация о перевозках**, в которой выводится следующая информация: дата рейса, вес груза, пробег. Форма должна содержать подчиненную форму **Водитель**, в которой выводится фамилия водителя и подчиненную форму **Автомобиль**, в которой выводится номер автомобиля, марка автомобиля.

#### **Вариант 20. БД Расписание.**

Создать форму **Информация о поезде**, в которой выводится следующая информация: № поезда, категория, Ф.И.О. начальника поезда, количество вагонов, наличие вагона-ресторана. Форма должна содержать подчиненную форму **Маршрут**, в которой выводится пункт отправления, пункт назначения, время отправления, время прибытия.

#### **Вариант 21. БД Соревнования.**

Создать форму **Информация о соревнованиях**, в которой выводится следующая информация: место проведения, дата, Ф.И.О. участника, вид спорта, результат, занятое место. Форма должна содержать подчиненную форму **Участник**, в которой выводится Ф.И.О. участника, дата рождения, спортивный разряд, рост, вес.

#### **Вариант 22. БД Праздничное бюро.**

Создать форму **Информация о мероприятиях**, в которой выводится следующая информация: код мероприятия, место проведения, дата, время начала, время окончания, количество мест. Форма должна содержать подчиненную форму **Сотрудник**, в которой выводится Ф.И.О., комиссионные и подчиненную форму **Мероприятие**, в которой выводится название мероприятия, цена за одно место.

#### **Вариант 23. БД Составление учебного плана.**

Создать форму **Информация о преподавателях**, в которой выводится следующая информация: код преподавателя, Ф.И.О., ученая степень. Форма должна содержать подчиненную форму **Предмет**, в которой выводится название, количество часов, итоговый контроль и подчиненную форму **Группа**, в которой выводится номер группы.

#### **Вариант 24. БД Туристическая фирма.**

Создать форму **Информация о предложениях**, в которой выводится следующая информация: код путевки, страна, климат, продолжительность, цена, категория отеля, название отеля. Форма должна содержать подчиненную форму **Путевка**, в которой выводится дата отправления, скидка и подчиненную форму **Клиент**, в которой выводится Ф.И.О., адрес, телефон.

#### **Вариант 25. БД Грузовые перевозки.**

Создать форму **Информация о маршрутах**, в которой выводится следующая информация: код маршрута, пункт назначения, дальность, оплата. Форма должна содержать подчиненную форму **Водитель**, в которой выводится Ф.И.О., дата рождения, стаж и подчиненную форму **Маршрут**, в которой выводится дата отправки, дата возвращения, премия.

#### **Вариант 26. БД Прокат автомобилей.**

Создать форму **Информация об автомобилях**, в которой выводится следующая информация: код автомобиля, марка, стоимость автомобиля, стоимость проката, год выпуска. Форма должна содержать подчиненную форму **Клиент**, в которой выводится фамилия, имя, отчество, адрес, телефон и подчиненную форму **Даты**, в которой выводится дата выдачи, дата возврата.

## Лабораторная работа №7

### СОЗДАНИЕ МАКРОСОВ И ГЛАВНОЙ КНОПОЧНОЙ ФОРМЫ

**Цель лабораторной работы:** приобрести навыки создания макросов и главной кнопочной формы.

#### Порядок выполнения лабораторной работы

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания макросов и главной кнопочной формы. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.

#### 1. Краткие теоретические сведения

Макрос представляет собой автоматизированную последовательность команд, направленную на выполнение каких-либо действий. С помощью макросов можно запускать одновременно запросы, отчеты, формы, таблицы, контролировать правильность ввода данных, выполнять различные действия после щелчка на кнопке, к которой подключен макрос и др. Для создания и редактирования макросов в Access предназначено специальное окно для конструирования макросов.

Главная кнопочная форма представляет собой форму, которая позволяет связывать отдельные объекты базы данных в единое целое. Кнопочную форму также можно использовать как пользовательское диалоговое окно для ввода данных и выполнения действий, определяемых введенными данными.

#### 2. Пример создания макросов и главной кнопочной формы

##### 2.1. Создание макросов

*Создать макросы для быстрого запуска запросов.*

1. В окне БД **Деканат** на вкладке **Макросы** нажмите кнопку **Создать**. В результате откроется окно конструктора макросов.
2. В ячейке столбца **Макрокоманда** из раскрывающегося списка выберите команду **ОткрытьЗапрос**. Имя запроса **Запрос1** задайте в области конструктора макросов *Аргументы макрокоманды*. Окончательный вид окна конструктора макросов представлен на рисунке 7.1.
3. Сохраните макрос под именем **Макрос\_ОткрытьЗапрос1**, закройте созданный макрос.

4. Запустите макрос **Макрос\_ОткрытьЗапрос1** на выполнение и убедитесь, что в результате открывается Запрос1.
5. Самостоятельно создайте макрос **ОткрытьПерекрестныйЗапрос**.

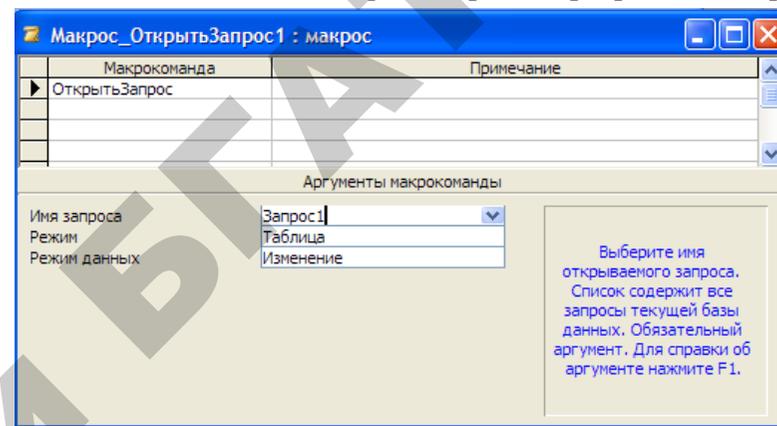


Рисунок 7.1 – Окно конструктора макросов

##### 2.2. Пример создания кнопочной формы

*Создать кнопочную форму для быстрого доступа к созданным формам, запросам и отчетам.*

1. Выполните команду **Сервис** → **Служебные программы** → **Диспетчер кнопочных форм**.
2. Подтвердите создание кнопочной формы, щелкнув по кнопке **Да**.
3. Для создания новой страницы кнопочной формы нажмите кнопку **Создать** в окне **Диспетчера кнопочных форма** (рисунок 7.2).

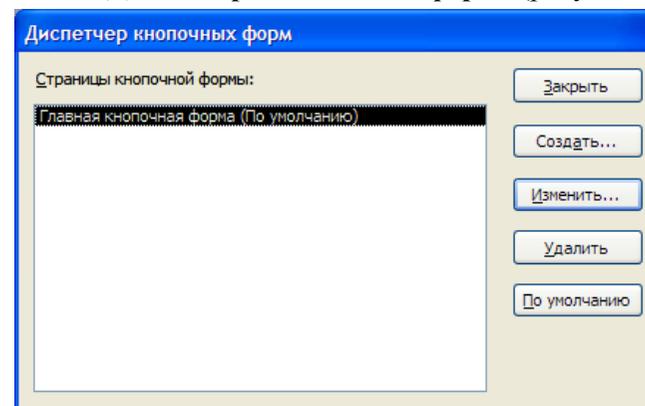


Рисунок 7.2 – Диалоговое окно Диспетчер кнопочных форм

4. В появившемся диалоговом окне **Создание** введите имя страницы кнопочной формы **Кнопочная форма**, как показано на рисунке 7.3. Щелкните по кнопке **ОК**.

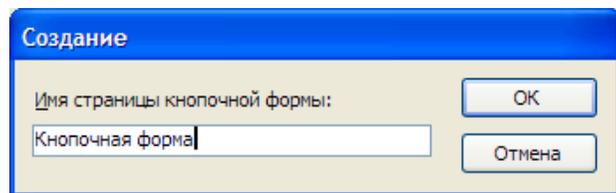


Рисунок 7.3 – Задание имени страницы кнопочной формы

5. В окне **Диспетчера кнопочных форм** выберите созданную страницу **Кнопочная форма** и щелкните по кнопке **Изменить**.

6. Далее в окне **Изменение страницы кнопочной формы** нажмите кнопку **Создать** (рисунок 7.4).

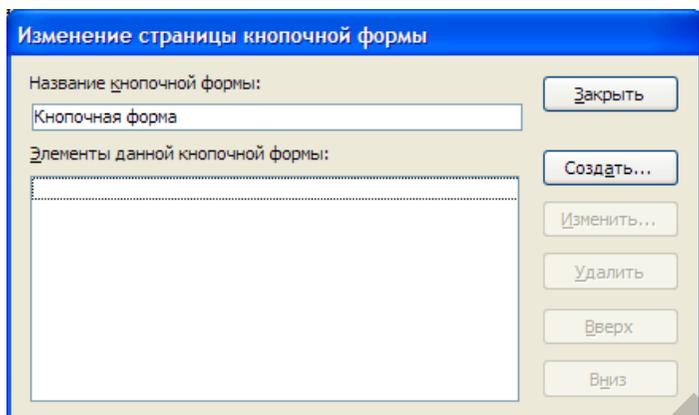


Рисунок 7.4 – Диалоговое окно Изменение страницы кнопочной формы

7. В диалоговом окне **Изменение элемента кнопочной формы** (рисунок 7.5) выполните следующие действия:

а) в поле **Текст** введите имя создаваемой кнопки – **Ввод и редактирование таблицы Дисциплины**;

б) список доступных команд высвечивается при нажатии на стрелку в поле **Команды**. Выберите команду **Открыть форму для изменения**;

в) в поле **Форма** из раскрывающегося списка выберите форму **Дисциплины**;

г) нажмите кнопку **ОК** для возвращения в меню **Изменение страницы кнопочной формы**.

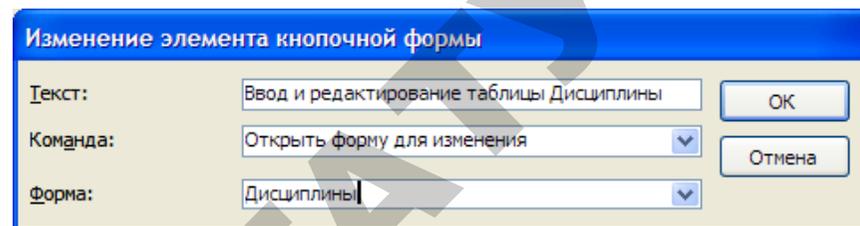


Рисунок 7.5 – Диалоговое окно Изменение элемента кнопочной формы

8. Самостоятельно создайте следующие кнопки:

- для ввода и редактирования таблицы **ЛичныеДанные**;
- для открытия формы **Информация о студентах**;
- для открытия **Отчета1**;
- для выполнения макросов, созданных в разделе 2.1.

9. После определения всех элементов кнопочной формы (рисунок 7.6) закройте окно **Изменение элемента кнопочной формы** кнопкой **Закреть**.

10. В окне **Диспетчер кнопочных форм** выберите страницу **Кнопочная форма** и щелкните по кнопке **По умолчанию**, затем закройте диалоговое окно кнопкой **Закреть** (рисунок 7.7).

11. Откройте кнопочную форму на вкладке **Формы** и просмотрите ее возможности.

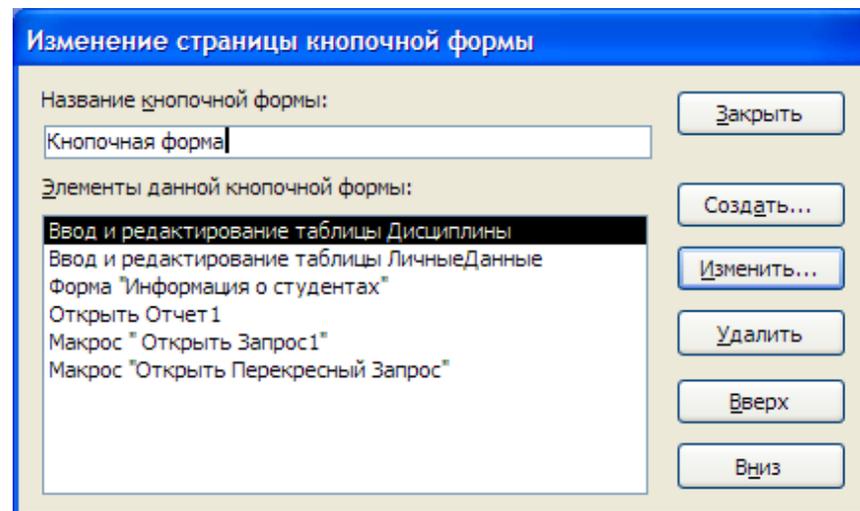


Рисунок 7.6 – Определение элементов Кнопочной формы

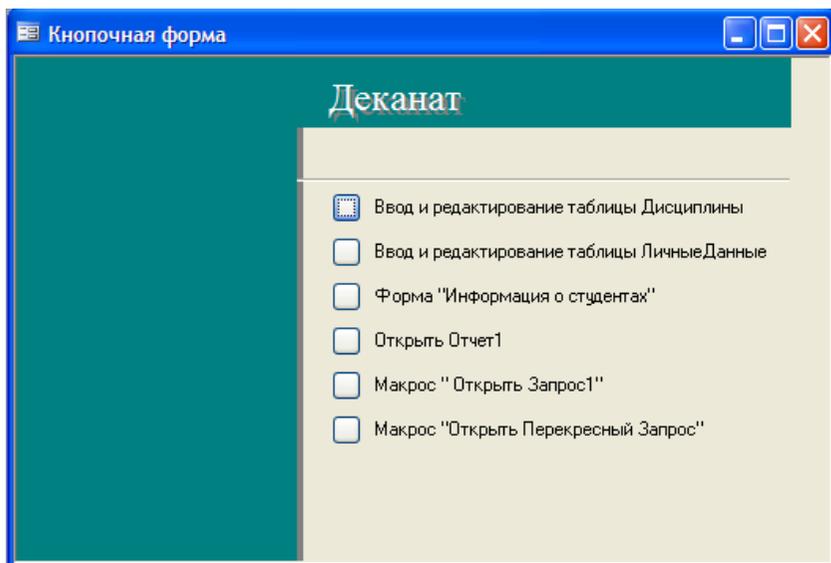


Рисунок 7.7 – Кнопочная форма БД Деканат

### 2.3. Автоматическое открытие кнопочной формы при загрузке базы данных

Чтобы кнопочная форма открывалась автоматически при загрузке базы данных, выберите в меню **Сервис** команду **Параметры запуска**. Заполните окно Параметры запуска в соответствии с рисунком 7.8.

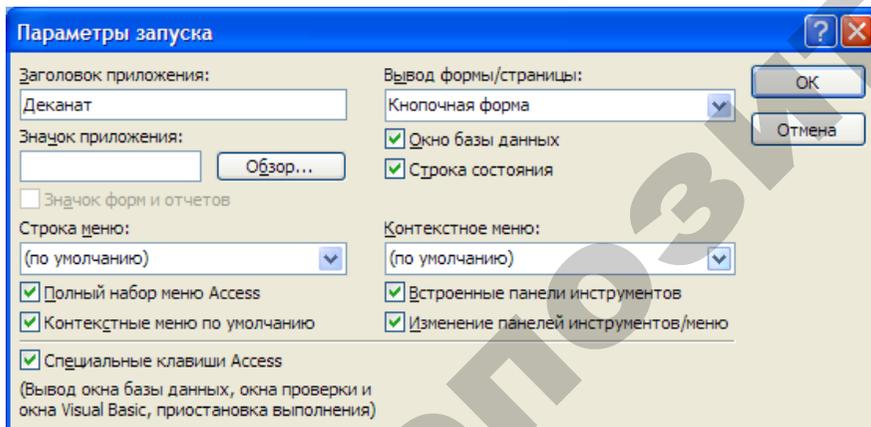


Рисунок 7.8 – Диалоговое окно Параметры запуска

### 3. Вопросы для самоконтроля

1. Для чего используются макросы?
2. Как создать макрос в приложении *MS Access*?
3. Что такое кнопочные формы?
4. С какой целью создаются кнопочные формы?
5. Как в *MS Access* создать кнопочную форму?
6. Как можно включить в форму кнопки?
7. Как открыть таблицу или запрос с использованием кнопочной формы?
8. Как изменить настройки параметров запуска базы данных?

### 4. Варианты индивидуальных заданий

Создать главную кнопочную форму, которая позволит автоматизировать выполнение следующих функций:

1. просмотр содержимого базы данных с возможностью их редактирования в режиме таблицы;
2. просмотр содержимого базы данных путем открытия созданных форм и запросов;
3. просмотр созданных отчетов;
4. закрытие базы данных.

Настроить базу данных таким образом, чтобы при ее открытии на экран автоматически выводилась главная кнопочная форма.

## ЛИТЕРАТУРА

1. Информатика. Базовый курс / Симонович С. В. [и др.]; под ред. С. В. Симоновича. – Санкт–Петербург: Питер, 2010. – 419с.
2. Тутьгин, В.С. Информатика. Microsoft Office Professional / В.С.Тутьгин. – СПб: БХВ-Петербург, 2005. – 368 с.
3. Кузнецов, С. Д. Основы баз данных / С. Д. Кузнецов – СПб: Бином, 2007. – 488 с.
4. Золотова, С. И. Практикум по Access / С. И. Золотова. – Москва: Финансы и статистика, 2007. – 144 с.
5. Берлинер, Э. М. Microsoft Office 2003 / Берлинер Э. М., Глазырина И. Б., Глазырин Б. Э. - СПб: Бином-Пресс, 2004. – 576 с
6. Кошелев, В.Е. Access 2003. Практическое руководство / В.Е. Кошелев. – СПб: Бином, 2008. – 464 с.

### Материальное обеспечение занятий

1. Операционная система *Microsoft Windows XP Professional*
2. Обучающая система компании МедиаХауз «Как работать с офисными программами».
3. Обучающая система «Практический курс *Access XP*».
4. Пакет прикладных программ *Micrsoft Office*.
5. Электронные материалы для дисциплины «Информационные технологии».

### Основные операторы, используемые при создании выражений

Оператор	Описание	Примеры
<b>And</b>	Записи, удовлетворяющие одному и другому условию одновременно	<b>Like "трактор*" And Like "авто*"</b> (выбор записей, которые в поле <i>НазваниеДисциплины</i> содержат слова «тракторы» и «автомобили»)
<b>Or</b>	Записи, удовлетворяющие хотя бы одному из двух или более условий	<b>"Первый" Or "Второй"</b> (выбор записей, которые в поле <i>Фамилия</i> содержат слова «Первый» или «Второй»)
<b>Not</b>	Записи, не удовлетворяющие данному условию	<b>not «Первый»</b> (выбор записей, которые в поле <i>Фамилия</i> содержат любое значение, отличное от «Первый»)
<b>&amp;</b>	Слияние двух строковых выражений	<b>[Фамилия]&amp;[Имя]&amp;[Отчество]</b> (объединяет поля <i>Фамилия</i> , <i>Имя</i> , <i>Отчество</i> в одно поле)
<b>Like</b>	Записи, содержащие данное выражение	<b>Like "П*"</b> (значения, начинающиеся в буквы «П») <b>Like "*ан"</b> (значения, заканчивающиеся на «ан») <b>Like "[б-г]*"</b> (значения, начинающиеся с одной из букв «б», «в», «г») <b>Like «*ло*»</b> (значения, содержащие «ло»)
<b>Between... And</b>	Записи, имеющие значение между двумя указанными	<b>Between 02.02.1988 And 04.04.1988</b> (даты между 2 февраля 1988 года и 4 апреля 1988 года)
<b>Is Null</b>	Записи, не имеющие значения в данном поле	<b>Is Null</b> (в поле <i>Телефон</i> находит записи, в которых телефонный номер не был введен)

Оператор	Описание	Примеры
<b>Is Not Null</b>	Записи, имеющие значение в данном поле	<b>Is Not Null</b> (в поле <i>Телефон</i> находит записи, в которых телефонный номер был введен)
=	Равно	= <b>180</b> (значение равно 180)
>, >=	Больше, больше или равно	> <b>01.01.2000</b> (дата после 1 января 2000 года)
<, <=	Меньше, меньше или равно	< <b>01.02.2010</b> (дата до 1 февраля 2010 года)
<>	Не равно	<> «Минск» (значение, отличное от «Минск»)
*, /, +, -	Умножение, деление, сложение, вычитание	<b>[Цена]*[Количество]</b> (произведение поля <i>Цена</i> на значение поля <i>Количество</i> )

### Функции, используемые при групповых операциях в запросах

Функция	Описание
<b>Avg</b>	Возвращает среднее арифметическое всех значений данного поля в каждой группе
<b>Count</b>	Количество непустых записей запроса
<b>First</b>	Возвращает первое значение данного поля в группе
<b>Last</b>	Возвращает последнее значение данного поля в наборе
<b>Max</b>	Возвращает наибольшее значение, найденное в данном поле внутри каждой группы
<b>Min</b>	Возвращает наименьшее значение, найденное в данном поле внутри каждой группы
<b>Sum</b>	Возвращает сумму всех значений данного поля в каждой группе
<b>StDev</b>	Возвращает стандартное отклонение всех значений данного поля в каждой группе. Эта функция применяется только к числовым или денежным полям. Если в группе меньше двух строк, <i>Microsoft Access</i> возвращает значение Null
<b>Var</b>	возвращает дисперсию значений данного поля в каждой группе. Эта функция применима только к числовым или денежным полям. Если в группе менее двух строк, <i>Access</i> возвращает значение Null

### Функции для работы с данными типа дата/время

#### Date()

Функция возвращает текущую системную дату в виде *дд.мм.гггг*, где *дд* – день (01-31), *мм* – месяц (01-12), *гггг* – год.

#### Now()

Функция возвращает текущую дату и время в соответствии с системной датой компьютера.

#### Time()

Функция возвращает текущее время в соответствии с системным временем компьютера.

#### DateAdd (*интервал*; *количество*; *дата*)

Функция возвращает дату, к которой прибавлен указанный интервал времени. Аргументами функции являются:

*интервал* – строковое выражение, определяющее интервал времени (день, неделя, месяц);

*количество* – числовое значение, определяющее количество интервалов времени, которые следует добавить (*количество* – положительное число) или вычесть (*количество* – отрицательное число) из *даты*.

Аргумент *интервал* принимает следующие значения: уууу – год; q – квартал; m – месяц; у – день года; d – день; w – день недели; ww – неделя; h – час; n – минута; s – секунда.

*Пример*: DateAdd("уууу";2;Date()) – к текущей дате прибавить 2 года.

#### DateDiff(*интервал*, *дата1*, *дата2*[, *первый\_день\_недели* [,*первая\_неделя\_года*]])

Функция используется для вычисления количества временных интервалов между двумя датами.

Аргумент *интервал* – строковое выражение, определяющий интервал времени, в которых будет выводиться результат вычитания из *даты1* *даты2*.

Значения, которые может принимать *интервал* см. в функции DateAdd.

Аргумент *первый\_день\_недели* – необязательный.

*Пример*: DateDiff("m";#19.02.2009#;#19.10.2009#) – будет получен результат 8 месяцев.

#### Day(*дата*)

Функция возвращает целое число в диапазоне от 1 до 31, обозначающее день месяца *даты*.

*Пример*: Day(#01.12.2009#) – будет получен результат «01».

#### Month(*дата*)

Функция возвращает целое число в диапазоне от 1 до 12, обозначающее месяц *даты*.

*Пример*: Day(#01.12.2009#) – будет получен результат «12».

#### Year(*дата*)

Функция возвращает целое число, обозначающее год *даты*.

*Пример*: Day(#01.12.2009#) – будет получен результат «2009».

#### Hour(*время*)

Функция возвращает целое число в диапазоне от 0 до 23, обозначающее час суток.

#### Minute(*время*)

Функция возвращает целое число в диапазоне от 0 до 59, обозначающее минуту часа.

#### Second(*дата*)

Функция возвращает целое число в диапазоне от 0 до 59, обозначающее секунду минуты.

**Функции для работы со строковыми данными**

**Lcase (строка)** – функция преобразует символы *строки* в нижний регистр.

**Ucase (строка)** – функция преобразует символы *строки* в верхний регистр.

**Len (строка)** – функция используется для подсчета количества символов в *строке*.

**Left (строка; длина)** – функция возвращает из *строки* указанное число символов от левого края строки.

**Right (строка; длина)** – функция возвращает из *строки* указанное число символов от правого края строки.

**Mid (строка; начало\_поиска[, длина])** – функция возвращает из *строки* указанное число символов. Аргумент *начало\_поиска* определяет место в строке, начиная с которого берутся символы.

**Replace(строка; искомая\_строка; строка\_замены [; начало [; сравнение])** – функция используется для замены в *строке* символа или подстроки на другую подстроку.

Аргументами функции являются:

*строка* – исходная строка;

*искомая\_строка* – подстрока, которую требуется найти;

*строка\_замены* – подстрока, на которую производится замена;

*начало, число, сравнение* – необязательные аргументы.

Шакирин Анатолий Иванович, Львова Ольга Михайловна

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

Разработка СУБД в Microsoft Office Access

*Лабораторный практикум*

Ответственный за выпуск *Ю.Н. Силкович*

Подписано в печать 21.01.2010 г. Формат 60×84<sup>1</sup>/<sub>16</sub>. Бумага офсетная. Ризография. Усл. печ. л. 7,21. Уч.-изд. л. 5,63. Тираж 85 экз. Заказ 75.

Издатель и полиграфическое исполнение:  
учреждение образования

«Белорусский государственный аграрный технический университет».

ЛИ № 02330/0131734 от 10.02.2006.

ЛП № 02330/0131656 от 02.02.2006.

Пр. Независимости, 99–2, 220023, Минск.