

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЙ И ДИЗАЙНА»

**Кафедра информационных систем и компьютерного дизайна**

**Методические указания и задания к контрольной работе  
по дисциплине «Информационные системы и технологии».**

**Часть №1**

**направление подготовки 09.03.03. «Прикладная информатика»**

**профиль подготовки: *Прикладная информатика в экономике***

***Прикладная информатика в дизайне***

**заочной формы обучения**

**Составитель:**

**С.В. Лебедева**

**Санкт-Петербург**

**2016**



# **1. ВВЕДЕНИЕ**

## **1.1. Цель дисциплины**

Получение теоретических знаний и практических навыков по основам архитектуры и функционирования информационных систем (ИС). Сформировать компетенции обучаемого по применению современных информационных технологий для разработки информационных технологий и систем.

## **1.2. Задачи дисциплины**

В результате изучения первой части дисциплины студенты должны:

- приобрести прочные знания и практические навыки по применению современных информационных систем и технологий;
- обладать практическими навыками использования функциональных и обеспечивающих подсистем;
- свободно ориентироваться в различных видах информационных систем, знать их архитектуру;
- иметь представление об основных тенденциях развития современных информационных технологий и систем;
- обладать практическими навыками использования информационных технологий в различных информационных системах.

Дисциплина «ИСиИТ» изучается студентами на втором курсе в 4 – м семестре. Учебный план предусматривает контрольную работу и экзамен. Контрольная работа выполняется в установленные учебным графиком сроки.

## **1.3. Структура и содержание дисциплины:**

### **Учебный модуль 1. Информационные системы**

Тема 1. Основные процессы преобразования информации.

Тема 2. Определение, общие принципы построения и цели разработки информационных систем.

Тема 3. Архитектура и жизненный цикл ИС.

### **Учебный модуль 2. Информационные технологии**

Тема 4. Основные понятия, терминология и классификация информационных технологий.

Тема 5. Информационно - коммуникационные технологии общего назначения.

Тема 6. Информационные системы и технологии интеллектуальной поддержки управленческих решений

### **Учебный модуль 3. Типология ИС**

Тема 7. Документальные информационно-поисковые системы (ДИСП). Принципы построения, структура и основные этапы функционирования ДИСП. Понятие pertinентности и релевантности. Поисковый образ документа. Поисковое предписание. Формальное представление семантики документа. Информационно-поисковые языки. Автоматизированные информационно-поисковые системы, классификаторы.

Тема 8. Фактографические информационные системы. Предметная область (ПО). Информационный объект ПО (сущность), понятие способы задания. Концептуальное моделирование фактографических ИС (модель «сущность-связь»). Программные средства реализации фактографических ИС.

Тема 9. Модели данных, назначение, классификация. Классификация моделей данных. Общие понятия СУБД. Классификация СУБД. Функции СУБД. Архитектура СУБД, независимость данных, объекты моделирования, схемы СУБД. Типы моделей данных.

Сетевая и иерархическая модели данных. Представление данных, операции над данными, ограничение целостности. Реляционная модель данных. Понятие отношения. Схемы отношений. Общие свойства отношений. Объектно-связанная модель.

## **КОНТРОЛЬНЫЕ ЗАДАНИЯ**

При изучении дисциплины ИС и Т студенты, обучающиеся по заочной форме, выполняют контрольные работы, предусмотренные учебным графиком. Контрольные работы представляют собой решение задач на заданные темы. Задачи выбираются студентами из приведенного в данных методических указаниях списка контрольных заданий. Номер варианта контрольного задания, которое должен выполнять студент, соответствует последней цифре номера его зачетной книжки.

### **ЗАДАЧА 1**

**I.** Создать базу данных турбюро, содержащую следующие сведения: Номер путевки; Код маршрута; Наименование маршрута; Дата отправления; Продолжительность дней; Цена за день, руб; Признак оплаты (наличный или безналичный расчет); количество продаж маршрута.

Заполнить базу, исходя из следующей семантики: а) несколько путевок по одному маршруту, б) несколько путевок за один день.

1. Создать составную форму для ввода данных;
2. Ввести в БД 10 записей;
3. Определить процент спроса для каждого маршрута;
4. Подсчитать общую стоимость путевок, оплаченных по безналичному расчету;
5. Вывести на экран список всех отъезжающих по одному маршруту в один и тот же день;
6. Увеличить на 35% стоимость путевок, оплаченных по наличному расчету ;
7. Увеличить первоначальную стоимость путевок в зависимости от спроса (>70% на 20%, от 50% до 70% на 15%, меньше 50% на 5%);
8. Создать отчет, содержащий поля: номер путевки, продолжительность, стоимость. Вывести в отчете подсчет общей стоимости и стоимости по направлениям

### **ЗАДАЧА 2**

**I.** Создать базу данных оперативных данных АТП, содержащую следующие сведения: Номер путевого листа; Расстояние, км; Код маршрута; Наименование маршрута; Дата выдачи путевого листа; Стоимость бензина, руб/л; Гаражный номер автомобиля; Марка автомобиля; Расход бензина, л/км.

Заполнить базу, исходя из следующей семантики: а) несколько путевок по одному маршруту, б) несколько автомобилей одной марки.

1. Создать составную форму для ввода данных;
2. Ввести в БД 10 записей;
3. Вычислить стоимость пробега для каждой записи;
4. Вычислить процент расхода бензина для каждой марки автомобиля;
5. Увеличить стоимость пробега на 20% определенной марки автомобиля на определенном маршруте ;
6. Подсчитать общую стоимость пробега всех автомобилей;
7. Увеличить расход бензина на 20%, если расстояние маршрута от 100 до 150 км, на 25% - если от 150 км до 300 км и на 50%, если больше 300 км;
8. Создать отчет, содержащий поля: номер путевого листа, маршрут, расстояние. Вывести в отчете подсчет стоимости пробега по маршрутам и общей стоимости.

### **ЗАДАЧА 3**

**I.** Создать базу данных промышленного предприятия, содержащую следующие сведения: Код цеха; Наименование цеха; Код изделия; Наименование изделия; План выпуска, шт.; Стоимость, руб/шт.; Брак, %.

Заполнить базу, исходя из следующей семантики: а) несколько изделий в одном цеху, б) не менее 2 цехов.

1. Создать составную форму для ввода данных;
2. Ввести в БД 10 записей;
3. Вывести долю брака (%) по каждому цеху;
4. Подсчитать суммарный выпуск по плану и фактический выпуск изделий одного цеха;
5. Уменьшить стоимость изделий на 10%, по которым не выполнен план выпуска;
6. Подсчитать общую стоимость изделий по цехам (план);
7. Уменьшить план выпуска изделий на 15%, если брак более 15% , на 10%- если брак от 10% до 15% и увеличить в два раза, если брак меньше 10%;
8. Составить отчет, содержащий поля: наименование изделия, выпуск по плану, фактический выпуск. Подсчитать в отчете суммарную стоимость изделия по цехам по плану и фактического выпуска. Подсчитать общую стоимость по плану.

#### **ЗАДАЧА 4**

**I.** Создать базу данных отдела кадров, содержащую следующие сведения: Табельный номер; ФИО; Код профессии; Наименование профессии; Дата поступления на работу; Адрес; Оклад; Номер отдела; Признак увольнения (логический); Дата увольнения.

Заполнить базу, исходя из следующей семантики: а) несколько работников в одном отделе, б) не менее двух отделов, в) несколько работников одной профессии, работающих в разных отделах.

1. Создать составную форму для ввода данных;
2. Ввести в БД 10 записей;
3. Подсчитать количество работников в каждом отделе и общую зарплату по отделам;
4. Вычислить долю зарплаты (в %) по каждому отделу;
5. Вычислить количество отработанных дней уволенных работников;
6. Вычислить количество заплаченных денег уволенных работников за фактически отработанное время;
7. Увеличить зарплату работающим сотрудникам определенной профессии в зависимости от номера отдела (1 отдел на 10%, 2 отдел на 20%, другим отделам на 5%)
8. Создать отчет, содержащий поля: Табельный номер; ФИО; Оклад. Вывести подсчет окладов работников по отделам и общую сумму окладов.

#### **ЗАДАЧА 5**

**I.** Создать базу данных учебного заведения, содержащую следующие сведения:

Код факультета, номер группы, номер зачетки, ФИО, код дисциплины, наименование дисциплины, дата сдачи экзамена, оценка, стипендия, признак стипендии (да/нет).

Заполнить базу, исходя из следующей семантики: а) несколько студентов в группе, б) несколько групп на факультете, в) не менее двух экзаменов у некоторых студентов.

1. Создать составную форму для ввода данных;
2. Ввести в БД 10 записей;
3. Подсчитать общее количество студентов и общий расход стипендии по факультетам.
4. Вычислить долю (в %) расхода стипендии по факультетам;
5. Вывести на экран общее количество студентов, сдавших экзамен в определенный день;
6. Увеличить размер стипендии при наличии признака стипендии на 20%; студентам, сдавшим экзамен на отлично – на 30%.
7. Подсчитать средний балл одной из групп одного факультета по всем дисциплинам;

8. Создать отчет, содержащий поля: номер зачетки; ФИО; наименование дисциплины; оценка. Вывести в отчете количество студентов по факультетам, группам и общее количество студентов. Вывести в отчете средний балл по факультету, группе и общий средний балл.

#### **ЗАДАЧА 6**

**I.** Создать базу данных мелкооптового магазина, содержащую следующие сведения:

Номер магазина; Название магазина, Код товара; Наименование товара; Количество поступившего товара; Цена за ед., руб., количество проданного товара.

Заполнить базу, исходя из следующей семантики: а) несколько товаров в одном магазине, б) несколько одинаковых товаров по разной цене в разных магазинах, в) не менее 2 магазинов.

1. Создать составную форму для ввода данных;
2. Ввести в БД 10 записей;
3. Определить процент спроса товаров в каждом из магазинов.
4. Увеличить первоначальную стоимость товара в зависимости от спроса (>30% на 20%, от 20% до 30% на 15%, менее 20% – на 5%);

5. Подсчитать количество полностью проданных товаров;

6. Подсчитать общее количество и стоимость товара в одном из магазинов;

7. Увеличить количество наиболее популярного товара в магазинах на 20%.

8. Создать отчет, содержащий поля: адрес магазина; наименование товара; количество.

Подсчитать в отчете стоимость товара по магазинам и общую стоимость. Вывести среднее количество товара по магазинам и общее среднее количество.

#### **ЗАДАЧА 7**

**I.** Создать базу данных библиотеки, содержащую следующие сведения: Шифр книги; Код читателя; Наименование книги; Стоимость книги; Автор; ФИО читателя; Признак выдачи книги; Дата выдачи. Заполнить базу, исходя из следующей семантики: а) несколько книг одного автора, б) несколько выданных книг за один день, в) несколько выданных книг одному читателю.

1. Создать составную форму для ввода данных;

2. Ввести в БД 10 записей;

3. Посчитать долю (в %) выданных книг;

4. Увеличить первоначальную стоимость книг на 20% и подсчитать их общую стоимость выданных в одни руки;

6. Подсчитать процент спроса книг одного автора;

7. Увеличить залоговую стоимость книг в зависимости от спроса (>70% на 20%, от 50% до 70% на 15%, менее 50 на 10%);

8. Создать отчет, содержащий поля: Наименование книги; ФИО читателя; Стоимость. Подсчитать в отчете количество и стоимость книг, выданных читателям и общее количество и стоимость книг, имеющихся в библиотеке.

#### **ЗАДАЧА 8**

Создать базу данных музея, содержащую следующие сведения: Код музея; Код экспоната; Наименование экспоната; Стоимость экспоната; Наименование музея; Дата приобретения; Автор; Признак ценности экспоната (высокая, выше среднего, средняя), участие в выставке (логический). Заполнить базу, исходя из следующей семантики: а) несколько экспонатов в одном музее, б) несколько экспонатов одного автора, в) не менее двух музеев.

1. Создать составную форму для ввода данных;

2. Ввести в БД 10 записей;

3. Вычислить долю экспонатов (в%) каждого автора ;

4. Подсчитать общую стоимость и количество экспонатов, участвовавших в выставке.

5. Подсчитать общую стоимость и количество экспонатов одного автора;
6. Увеличить залоговую стоимость экспоната в зависимости от ценности (высокая - на 120%, выше среднего на 70%, средняя – на 50%);
7. Сравнить залоговую стоимость экспонатов одного автора в каждом из музеев (в %).
8. Создать отчет, содержащий поля: Наименование экспоната; Наименование музея; Стоимость экспоната.

Подсчитать в отчете количество и стоимость экспонатов по каждому музею и по всем музеям.

#### **ЗАДАЧА 9**

Создать базу данных договоров, содержащую следующие сведения: Код договора; Код заказчика; Наименование договора; Дата начала, Дата окончания, Стоимость договора; ФИО заказчика; Признак выполнения (логический); Ранг выполнения (1 очередь, 2 очередь, 3 очередь).

Заполнить базу, исходя из следующей семантики: а) несколько договоров одного заказчика, б) несколько заказчиков на одну дату.

1. Создать составную форму для ввода данных;
2. Ввести в БД 10 записей;
3. Вычислить долю (в%) выполняющихся договоров.
4. Увеличить первоначальную стоимость выполненных договоров на 15% ; и подсчитать их общую стоимость;
5. Подсчитать общую стоимость и количество договоров, у которых один заказчик;
6. Подсчитать количество и стоимость тех договоров, у которых продолжительность выполнения меньше 100 дней; высчитать их процентную долю.
7. Увеличить стоимость договоров в зависимости от ранга (1 очередь - 25%, вторая - на 15%, третья на 5%);
8. Создать отчет, содержащий поля: Наименование договора; Наименование заказчика; Стоимость; Продолжительность. Подсчитать в отчете количество и стоимость договоров, заключенных каждым заказчиком и общее количество и стоимость заключенных договоров.

#### **ЗАДАЧА 10**

Создать базу склада запчастей, содержащую следующие сведения: Шифр запчасти; Код поставщика; Наименование запчасти; Цена запчасти; ФИО/наименование поставщика; Адрес; Количество, штук; Признак существования запчасти на складе (логический). Заполнить базу, исходя из следующей семантики: а) несколько запчастей одного поставщика, б) несколько поставщиков одной запчасти.

1. Создать составную форму для ввода данных;
2. Ввести в БД 10 записей;
3. Вычислить процентную долю поставляемых запчастей каждым поставщиком
4. Подсчитать общую стоимость запчастей, имеющих на складе;
5. Вычислить процентное содержание запчастей, имеющих на складе;
6. Снизить стоимость запчасти в зависимости от процента (>30% - на 15%, от 10% до 30% - на 10%, менее 10 – на 5%);
7. Подсчитать общую стоимость запчастей, поставляемых одним поставщиком.
8. Создать отчет, содержащий поля: Наименование запчасти; ФИО поставщика; Стоимость; Количество. Подсчитать в отчете количество и стоимость запчастей, поставляемых каждым поставщиком и общее количество, а также стоимость запчастей, имеющих на складе.

## **2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ КОНТРОЛЬНОЙ РАБОТЫ**

При изучении дисциплины "Базы данных" студенты, обучающиеся по заочной форме, выполняют контрольные работы, предусмотренные учебным графиком. Контрольные работы представляют собой решение задач на заданные темы.

## ТЕМА1. ПЛОСКИЕ БАЗЫ ДАННЫХ

### 1.1. Плоская база данных. Общие понятия

*Плоская база данных – это БД, состоящая из одной таблицы, например, телефонный справочник.*

Одним из способов организации плоских БД является создание списков в рабочей книге EXCEL. Список можно отсортировать по алфавиту, значению или в хронологическом порядке в соответствии с содержанием определенного поля.

В *Excel* можно не только сортировать записи списка, но и фильтровать их. *Фильтрация*— это быстрый и легкий способ поиска подмножества данных и работы с ними в списке. В отфильтрованном списке отображаются только строки, отвечающие условиям отбора, заданным для столбца. *Excel* предоставляет две команды для фильтрации списков: *автофильтр*, включая фильтр по выделенному, для простых условий отбора; расширенный фильтр для более сложных условий отбора.

#### 1. Автофильтр для простых условий отбора

При использовании команды **Фильтр** на вкладке *Данные* (рис. 1 и рис. 2) – включается *Автофильтр* и справа от названий столбцов в фильтруемом списке появляются кнопки со стрелками .

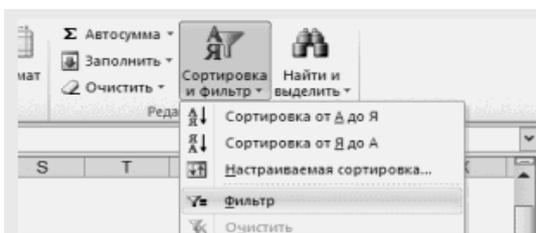


Рис. 1. Запуск команды **Фильтр** с ленты (вкладка *Главная*)

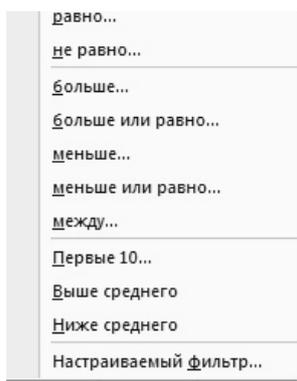


Рис. 2. Задание условий фильтра

#### 2. Расширенный фильтр для более сложных условий отбора

Для того, чтобы применить расширенный фильтр, необходимо предварительно подготовить данные для *Диапазона условий*, которые приложение *Excel* будет использовать для фильтрации. Данный диапазон должен состоять как минимум из двух строк, первая из

которых содержит заголовки полей списка, по которым необходимо хотите провести фильтрацию; другая состоит из условий, определенных пользователем.

Диапазон условий может находиться в любом месте рабочего листа или даже на отдельном рабочем листе

Рассмотрим в качестве примера следующую задачу: «*Отобразить всех студентов с одного факультета*».

*Решение.* Пусть имеем следующую плоскую базу данных **Студент** (табл. 1).

Таблица 1. База данных «Студент»

Фамилия	Имя	Отчество	Дата рождения	Место рождения	Факультет	Номер группы	Курс	Форма обучения
Первакова	Анастасия	Вадимовна	29.09.1996	Москва	ИБК	2-СД-8	2	Дневная
Маралев	Кирилл	Иванович	27.09.1994	Санкт-Петербург	ИБК	1-СД-8	1	Дневная
Гронская	Анна	Игоревна	12.06.1997	Вологда	ИБК	1-СВ-8	1	Вечерняя
Веселова	Юлия	Сергеевна	04.02.1997	Москва	ИБК	2-СД-8	2	Дневная
Погосов	Евгений	Анатолеви ч	04.07.1997	Санкт-Петербург	ИБК	1-СД-2	1	Дневная
Кудрявцев	Павел	Сергеевич	16.09.1994	Санкт-Петербург	ИБК	1-СД-6	1	Дневная
Оганина	Елизавета	Ильинична	03.07.1997	Москва	ИБК	2-СД-2	2	Дневная
Яковлева	Анна	Дмитриевна	04.01.1995	Санкт-Петербург	ИБК	1-СЗ-3	1	Заочная
Иванова	Эльвира	Мирзаевна	26.02.1996	Вологда	ИБК	1-СЗ-3	1	Заочная
Иванова	Анна	Леонидовна	01.10.1996	Санкт-Петербург	ИБК	1-СВ-9	1	Вечерняя
Корогодин	Владимир	Сергеевич	24.03.1997	Москва	ИБК	1-СД-9	1	Дневная
Зиновьева	Татьяна	Сергеевна	27.02.1993	Тосно	СЗППИ	1-СВ-9	1	Вечерняя

2. По условию задачи необходимо отобразить всех студентов с одного факультета, для этого:

2.1. Скопируем заголовки столбца - **Факультет**, отступив от последней строки БД две строчки

2.2. Введем условие – название одного из факультетов.

2.3. Выделим всю БД вместе с заголовками столбцов.

2.4. Вызовем команду **Расширенный фильтр** (вкладка **Данные** в группе **Сортировка и фильтр** кнопка **Дополнительно**), в появившемся диалоговом окне:

2.4.1. Поставим галочку во флажок «скопировать результат в другое место»;

2.4.2. В качестве *исходного диапазона* выделим весь список с заголовками столбцов;

2.4.3. В поле **Диапазон условий** укажем заранее подготовленный диапазон условий – заголовки «Факультет» и условие «ИБК»;

2.4.4. **Поместить результат в диапазон** – выполняя эту команду отступить две строчки от диапазона условий, выделить или одну ячейку, или две строки с количеством столбцов, равным количеству столбцов плоской БД;

2.4.5. Поставим флажок **Только уникальные записи** (рис. 3);

	A	B	C	D	E	F	G	H	I
1	Фамилия	Имя	Отчество	Дата рождения	Место рождения	Факультет	Номер группы	Курс	Форма обучения
2	Первакова	Анастасия	Вадимовна	29.09.1996	Москва	ИБК	2-СД-8	2	Дневная
3	Маралев	Кирилл	Иванович	27.09.1994	Санкт-Петербург	ИБК	1-СД-8	1	Дневная
4	Гронская	Анна	Игоревна	12.06.1997	Вологда	ИБК	1-СВ-8	1	Вечерняя
5	Веселова	Юлия	Сергеевна	04.02.1997	Москва	ИБК	2-СД-8	2	Дневная
6	Погосов	Евгений	Анатольевич	04.07.1997	Санкт-Петербург	ИБК	1-СД-2	1	Дневная
7	Кудрявцев	Павел					1-СД-6	1	Дневная
8	Оганина	Елизавета					2-СД-2	2	Дневная
9	Яковлева	Анна					1-СЗ-3	1	Заочная
10	Иванова	Эльвира					1-СЗ-3	1	Заочная
11	Иванова	Анна					1-СВ-9	1	Вечерняя
12	Корогодин	Владимир					1-СД-9	1	Дневная
13	Зиновьева	Татьяна					1-СВ-9	1	Вечерняя
14	Широкова	Марина					5-пЗ-2с	5	Заочная
15	Рожкова	Олеся					2-пЗ-3	2	Заочная
16									
17									
18	Факультет								
19	ИБК								
20									

**Расширенный фильтр**

Обработка

фильтровать список на месте

скопировать результат в другое место

Исходный диапазон: Лист2!\$A\$1:\$I\$15

Диапазон условий: Лист2!\$A\$18:\$A\$19

Поместить результат в диапазон: Лист2!\$A\$21

Только уникальные записи

OK Отмена

Рис. 3. Заполнение диалога расширенного фильтра

#### 2.4.6. Нажать «ОК».

### ЗАДАЧИ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

1. Создать плоскую БД «Студент» со следующими полями: Фамилия Имя, Отчество, Дата рождения, Место рождения, Факультет, Номер группы, Курс, Форма обучения;

2. Заполнить БД данными, руководствуясь следующими правилами:

2.1. В институте есть несколько факультетов (не менее 3);

2.2. На каждом факультете есть студенты разных курсов обучения (не менее 3 курсов на каждый факультет);

2.3. Каждый студент обучается на определенном курсе в определенной группе (не менее 2 студентов в группе; не менее 2 групп на курсе);

2.4. На каждом факультете есть студенты, обучающиеся на разных формах обучения (дневная (очная), вечерняя, заочная);

2.5. На каждом факультете есть студенты одной даты рождения.

**Используя автофильтр, решить следующие задачи:**

1. Отобразить всех студентов, чья фамилия начинается на букву А или на букву В;

2. Отобразить всех студентов из Москвы или Санкт-Петербурга

**Используя расширенный фильтр, решить следующие задачи:**

1. Отобразить всех студентов с одного факультета.

2. Отобразить всех студентов с одного факультета и одного года рождения.

3. Отобразить всех студентов с одного факультета, одного курса дневной формы обучения.

4. Отобразить всех студентов с одного факультета, из одной группы, из одного города и одного года рождения.

## 1.2. Применение функции «ЕСЛИ» для работы с БД

Функция ЕСЛИ применяется для вычисления определенного значения в зависимости от выполнения логического выражения.

Синтаксис:

**ЕСЛИ**(лог\_выражение; значение\_если\_истина; значение\_если\_ложь)

**Лог\_выражение** – это любое значение или выражение, использующее операторы отношения (>;<;>=;<=;;<>) и возвращающее в качестве результата логические константы true - ИСТИНА или false - ЛОЖЬ.

**Значение\_если\_истина** – это значение, которое возвращается или вычисляется, если лог\_выражение **выполняется**, т.е. возвращается логическая константа ИСТИНА.

**Значение\_если\_ложь** – это значение, которое возвращается или вычисляется, если лог\_выражение **не выполняется**, т.е. возвращается логическая константа ЛОЖЬ.

В качестве примера заполним свободный столбец БД по следующему правилу «вывести значения – «первокурсник», если студент обучается на 1 курсе и «старшекурсник», если студент обучается на курсах, отличных от 1».

**Решение.**

1. Во вторую ячейку свободного столбца БД (первые ячейки – заголовки столбцов) введем выражение =ЕСЛИ, в открывшемся диалоговом окне заполняем ее аргументы, согласно условию задачи (рис. 4

Номер группы	Курс	Форма обучения
2-СД-8	2	Дневная
1-СД-8	1	Дневная
1-СВ-8	1	Вечерняя
2-СД-8	2	Дневная
1-СД-2	1	Дневная
1-СД-6	1	Дневная
2-СД-2	2	Дневная
1-СЗ-3	1	Заочная
1-СЗ-3	1	Заочная
1-СВ-9	1	Вечерняя
1-СД-9	1	Дневная
1-СВ-9	1	Вечерняя
5-пЗ-2с	5	Заочная
2-пЗ-3	2	Заочная

Рис. 4. Заполнение аргументов функции ЕСЛИ

2. Протянем формулу по всему столбцу, используя правило автозаполнения. Результаты работы функции ЕСЛИ представлены на рис. 5.

Н	J
Курс	
2	старшекурсник
1	первокурсник
1	первокурсник
2	старшекурсник
1	первокурсник
1	первокурсник
2	старшекурсник
1	первокурсник
5	старшекурсник
2	старшекурсник

Рис. 5. Результаты работы функции ЕСЛИ

## ЗАДАЧИ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

**Используя функцию ЕСЛИ, решить следующие задачи:**

Создайте поле «Стипендия». Заполните его, используя функцию ЕСЛИ по следующему правилу:

«Если студент обучается по дневной форме обучения, стипендия равна 2000, если форма обучения вечерняя – 1000, если заочная – 500, если иная форма обучения -0».

2. Создайте поле «Наличие льгот», заполните его в произвольном порядке значениями «Да» или «Нет».

2.1. Создайте поле «Повышенная стипендия»;

2.2. Заполните его по следующему правилу:

«Если студент обучается на первом курсе и имеет льготы – повысить стипендию в 4 раза; если студент обучается на втором или третьем курсе и имеет льготы – повысить стипендию в 2,5 раза; если студент обучается на более старших курсах и имеет льготы – в 1.5 раза».

*Замечание:* если студенты не имеют льгот, стипендию им повышать не надо.

### 1.3. Функции для работы с БД

В Excel существуют функции для анализа данных в списках или БД. Каждая из этих функций (а всего их 13) использует три аргумента – *базу данных, поле и критерий*.

Наиболее часто используются следующие функции для обработки данных по БД – БСЧЕТ, БДСУММ, ДСРЗНАЧ (количество, сумма и среднее значение числовых полей БД).

В качестве примера рассмотрим решение следующей задачи:

*Подсчитать общее количество студентов одного из факультетов.*

**Решение.**

1. Выделим ячейку, в которую будет помещен результат выполнения функции;

2. Запустим мастера функций (значок  строки формул);

3. Выберем функцию БСЧЕТ из категории «Работа с базой данных» (рис. 6).

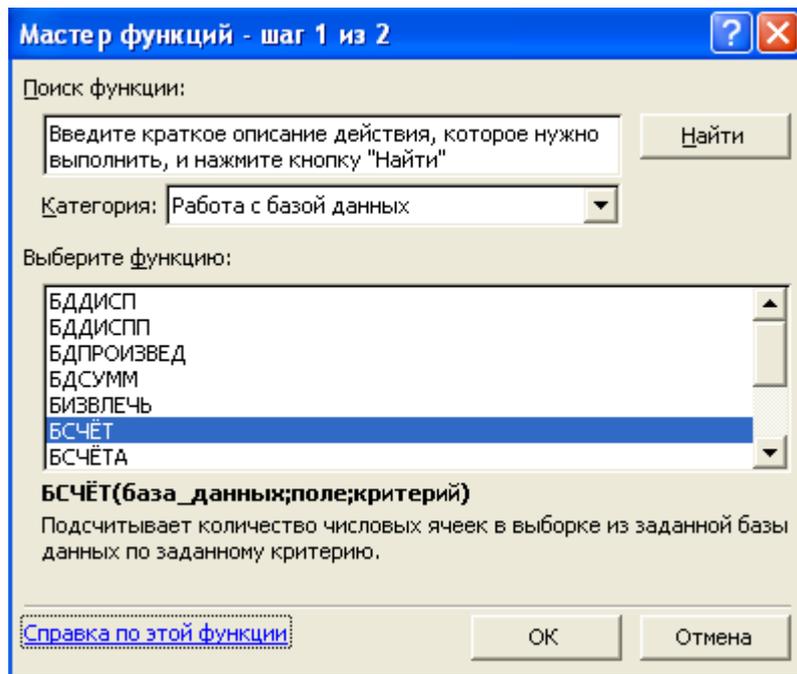


Рис. 6. Работа с мастером функций

4. В качестве аргументов функции выберем:

4.1. Аргумент «База\_данных» – выделим всю БД с заголовками столбцов;

4.2. Аргумент «Поле» для этой функции требует столбец с числовыми данными.

Можно выбрать любое поле, содержащее такие данные. В качестве примера выберем поле «Курс» (адрес заголовка столбца – Н1);

4.3. Аргумент «Критерий» должен быть подготовлен заранее (как в случае работы с расширенным фильтром). Воспользуемся условием отбора для расширенного фильтра, представленным на *рис. 3* – заголовок «Факультет» и условие «ИБК».

Заполнение аргументов функции представлено на *рис. 7*.

БСЧЁТ    X ✓    =БСЧЁТ(А1:И15;Н1;Критерии)

	A	B	C	D	E	F	G	H	I	J
	Фамилия	Имя	Отчество	Дата рождения	Место рождения	Факультет	Номер группы	Курс	Форма обучения	
1	Первакова	Анастас		29.09.1996	Москва	ИБК	2-СД-8	2	Дневная	
2	Маралев	Кирилл	Иванович	27.09.1994	Санкт-Петербург	ИБК	1-СД-8	1	Дневная	
3	Гронская	Анна	Игоревна	12.06.1997	Вологда	ИБК	1-СВ-8	1	Вечерняя	
4	Веселова	Юлия	Сергеевна	04.02.1997	Москва	ИБК	2-СД-8	2	Дневная	
5	Погосов	Евгений	Анатолевич	04.07.1997	Санкт-Петербург	ИБК	1-СД-2	1	Дневная	
6	Кудрявцев	Павел	Сергеевич	16.09.1994	Санкт-Петербург	ИБК	1-СД-6	1	Дневная	
7	Оганина	Елизавета	Ильинична	03.07.1995	Москва	ИБК	2-СД-2	2	Дневная	
8	Яковлева	Анна	Дмитриевна	04.01.1995	Санкт-Петербург	ИБК	1-СЗ-3	1	Заочная	
9	Иванова	Эльвира	Мирзаевна	26.02.1996	Вологда	ИБК	1-СЗ-3	1	Заочная	
10	Иванова	Анна	Леонидовна	01.10.1996	Санкт-Петербург	ИБК	1-СВ-9	1	Вечерняя	
11	Корогодин	Владимир	Сергеевич	24.03.1997	Москва	ИБК	1-СД-9	1	Дневная	
12	Зиновьева	Татьяна								
13	Широкова	Мария								
14	Рожкова	Олеся								
15										
16										
17										
18	Факультет									
19	ИБК									
20										
21	Фамилия	Имя	Отчество	Дата рождения	Место рождения	Факультет	Номер группы	Курс	Форма обучения	
22	Первакова	Анастасия	Вадимовна	29.09.1996	Москва	ИБК	2-СД-8	2	Дневная	
23	Маралев	Кирилл	Иванович	27.09.1994	Санкт-Петербург	ИБК	1-СД-8	1	Дневная	
24	Гронская	Анна	Игоревна	12.06.1997	Вологда	ИБК	1-СВ-8	1	Вечерняя	
25	Веселова	Юлия	Сергеевна	04.02.1997	Москва	ИБК	2-СД-8	2	Дневная	
26	Погосов	Евгений	Анатолевич	04.07.1997	Санкт-Петербург	ИБК	1-СД-2	1	Дневная	
27	Кудрявцев	Павел	Сергеевич	16.09.1994	Санкт-Петербург	ИБК	1-СД-6	1	Дневная	
28	Оганина	Елизавета	Ильинична	03.07.1995	Москва	ИБК	2-СД-2	2	Дневная	
29	Яковлева	Анна	Дмитриевна	04.01.1995	Санкт-Петербург	ИБК	1-СЗ-3	1	Заочная	
30	Иванова	Эльвира	Мирзаевна	26.02.1996	Вологда	ИБК	1-СЗ-3	1	Заочная	
31	Иванова	Анна	Леонидовна	01.10.1996	Санкт-Петербург	ИБК	1-СВ-9	1	Вечерняя	
32	Корогодин	Владимир	Сергеевич	24.03.1997	Москва	ИБК	1-СД-9	1	Дневная	
33										
34					Н1;Критерии)					
35										

**Аргументы функции**

БСЧЁТ

База\_данных: А1:И15 = {"Фамилия";"Имя";"О"

Поле: Н1 = "Курс"

Критерий: Критерии =

= 11

Подсчитывает количество числовых ячеек в выборке из заданной базы данных по заданному критерию.

**Критерий** диапазон, содержащий условие базы данных. Диапазон включает заголовки столбцов и одну ячейку с условием.

Справка по этой функции      Значение: 11

Рис. 7. Заполнение аргументов функции БСЧЕТ

**Замечание:** Все функции работают только с числовыми полями. При попытке ввода адреса нечислового поля в аргумент «Поле» возникнет ошибка выполнения функции.

5. В ячейку ввода функции будет возвращен результат работы функции.

Остальные функции работают аналогично.

### Задачи на самостоятельную работу

**Используя встроенные функции для работы с БД, решить следующие задачи:**

1. Подсчитать количество студентов с одного факультета.
2. Подсчитать количество и общую стипендию студентов одного факультета и одного года рождения.
3. Подсчитать количество и общую стипендию студентов с одного факультета, одного курса и дневной формы обучения.
4. Подсчитать количество и среднюю стипендию студентов с одного факультета, из одной группы, из одного города и одного года рождения.

## 1.4. Сводные таблицы в Excel

Одним из мощных средств Excel по работе с базами данных являются сводные таблицы. Они полезны как для анализа данных, так и для *агрегации (объединения)* данных, хранящихся в базе данных.

Сводная таблица (*Pivot Table*) — таблица, обеспечивающая фильтрацию данных по выбранным столбцам и подведение промежуточных итогов для более удобного анализа больших объемов данных и принятия более обоснованных решений.

Для создания сводной таблицы необходимо выполнить следующие шаги:

1. На вкладке **Вставить** в группе **Таблицы** нужно выбрать пункт **Сводная таблица** или выбрать стрелку под пунктом **Сводная таблица** и запустить команду **Сводная таблица**.

Excel автоматически определяет диапазон для отчета сводной таблицы, но можно заменить его, указав другой диапазон или имя, определенное для диапазона. В разделе **Укажите, куда следует поместить отчет сводной таблицы** нужно задать расположение, выполнив одно из описанных ниже действий:

— чтобы поместить отчет сводной таблицы на новый лист, начиная с ячейки **A1**, выбрать пункт **На новый лист**;

— чтобы поместить отчет сводной таблицы на существующий лист, выбрать пункт **На существующий лист** и в поле **Диапазон** надо указать первую ячейку диапазона, начиная с которой следует поместить отчет сводной таблицы;

— нажать кнопку ОК.

Microsoft Excel добавит пустой отчет сводной таблицы в указанное место и откроет список полей сводной таблицы, с помощью которого можно добавить поля, создать макет и настроить отчет сводной таблицы (рис. 8).

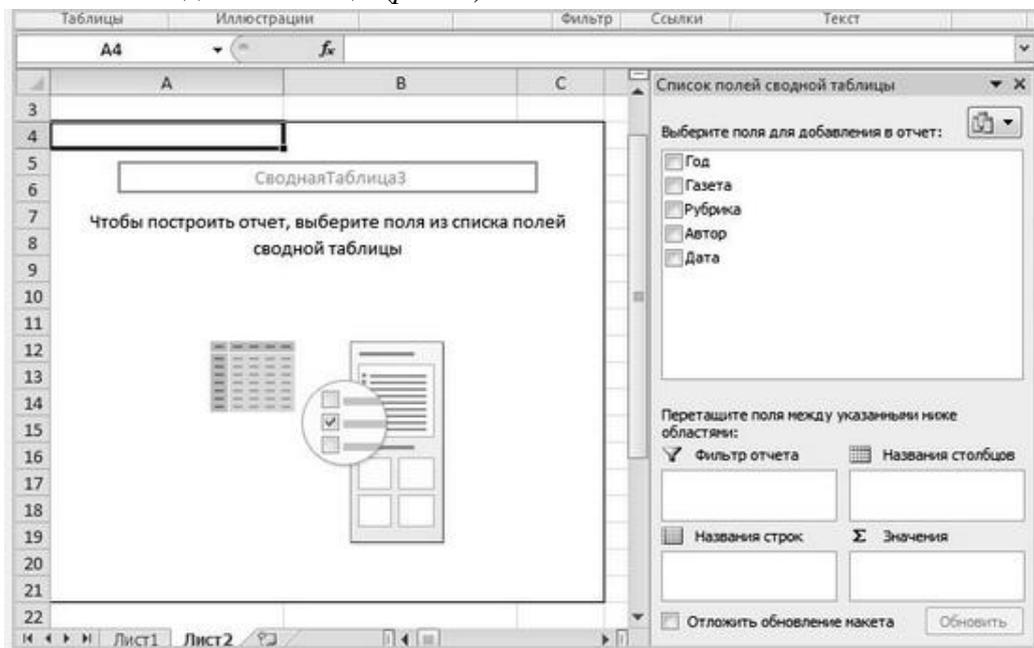


Рис. 8. Макет сводной таблицы и список полей сводной таблицы

2. Для добавления полей к отчету нужно выполнить одно или несколько из действий:

— чтобы добавить поле в область раздела макета, используемую по умолчанию, нужно установить флажок рядом с именем поля в разделе полей. По умолчанию нечисловые поля добавляются в область **Названия строк**, числовые поля — в область **Значения**, а поля даты и времени — в область **Названия столбцов**.

– чтобы поместить поле в определенную область раздела макета нужно щелкнуть его имя в разделе полей правой кнопкой мыши и выбрать пункт *добавить в Фильтр отчета*, *добавить в Названия столбцов*, *добавить в Названия строк* или *добавить в Значения*.

– далее выделить имя поля в разделе полей и, удерживая нажатой левую кнопку мыши, перетащить поле в требуемую область раздела макета.

3. Нажать кнопку **Готово**. Полученный итог работы можно зафиксировать в качестве решения поставленной задачи (рис. 9), либо продолжить работу с меню по настройке результатов сводной таблицы.

Количество	Факультет					
Курс	ИБК	СЗППИ	ФД	ФТО	ФЭМ	Общий итог
1	11		6	4		27
3			1			2
5					2	3
Общий итог	11		7	4	2	32

Рис. 9. Результат работы мастера сводных таблиц

## ЗАДАЧИ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

### *Используя мастер сводных таблиц:*

1. Подсчитать количество студентов по факультетам.
2. Подсчитать количество студентов по курсам и факультетам.
3. Подсчитать среднюю стипендию по факультетам, повышенную среднюю стипендию по факультетам. Используя дополнительный расчет, узнать во сколько раз среднее значение повышенной стипендии по факультетам больше запланированной (базовой), (найти количественное и процентное соотношение).
4. Подсчитать количество студентов, обучающихся на каждой из форм обучения. Используя дополнительный расчет, найти процентное соотношение каждой формы обучения.
5. Подсчитать количество студентов из СПб и других городов. Узнать во сколько раз количество студентов из СПб больше(меньше), чем количество студентов из других городов (результат выразить в количественном и процентом соотношении). *Замечание.* Для решения задачи надо предварительно подготовить данные. Для этого в БД создайте поле Город, используя функцию ЕСЛИ заполнить его значениями по следующему правилу «если студенты не из СПб, значение –Другой город, иначе –СПб».

## ТЕМА2. РАБОТА В СУБД ACCESS

*СУБД – Система Управления Базами Данных – это комплекс программ, предназначенный для создания, ведения и модификации данных в базе данных.*

*СУБД ACCESS – это комплекс программ, для создания **реляционных баз данных**.*

В основе реляционных баз данных лежит понятие *отношения (relation)* – двумерной таблицы особого вида, предназначенной для представления и хранения данных. Набор таблиц (или отношений) может быть использован для моделирования взаимосвязей между объектами реального мира и для хранения данных об этих объектах в базах данных.

## 2.1. Создание БД

Создать базу данных, используя СУБД *MS Access*, достаточно просто. Для этого необходимо запустить приложение, выполнить команду «Создать» и присвоить уникальное имя новой базе данных (рис. 10).

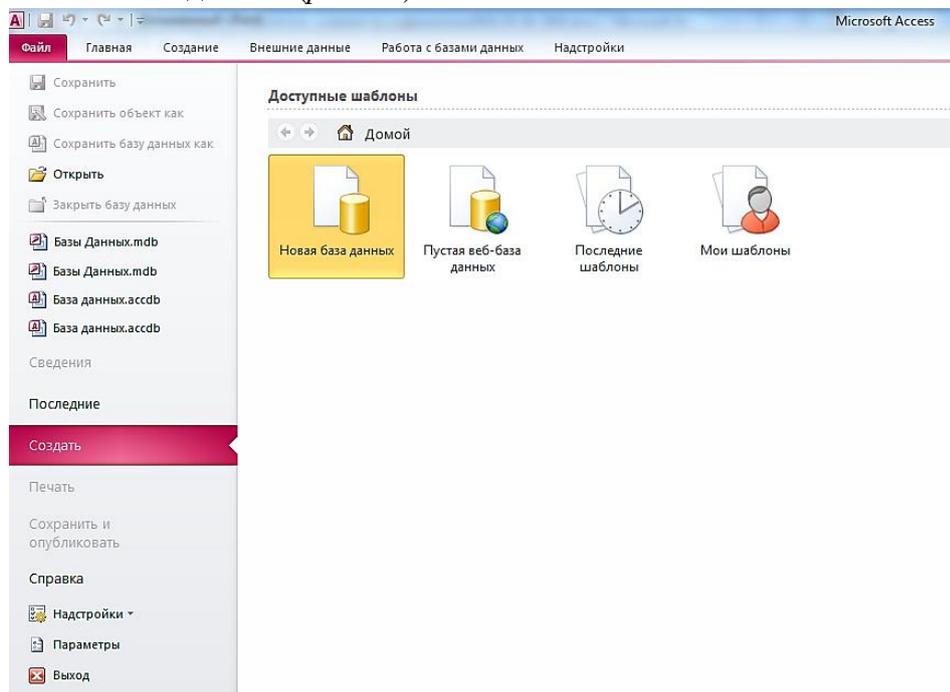


Рис. 10. Создание новой базы данных

## 2.2. Создание таблиц

Таблица – это объект БД, предназначенный для хранения сведений о реальных объектах реального мира в рассматриваемой предметной области.

*Под предметной областью будем понимать часть реального мира, информация о которой подлежит хранению в базе данных.*

Каждая запись таблицы хранит сведения об одном объекте рассматриваемой предметной области. Совокупность записей одной таблицы представляет сведения о классе однотипных объектов, обладающих одинаковыми свойствами. Название таблицы должно соответствовать классу объектов, например, таблица с названием **Заказы** должна содержать сведения о заказах, совершаемых при покупке товаров, а таблица **Товар** – все сведения о товарах, представленных в каком либо магазине.

Для того чтобы создать таблицу в СУБД *MS Access* нужно перейти на вкладку **Создание** и выполнить команду **Конструктор таблиц** (рис. 11).

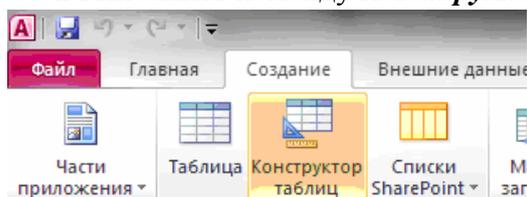


Рис. 11. Вызов конструктора таблиц

В общем виде определение таблицы состоит из двух этапов – определение структуры таблицы в режиме конструктора таблицы и непосредственное заполнение полей таблицы данными в режиме таблицы.

В конструкторе таблицы вводятся название полей, типы данных полей, и в зависимости от назначения поля, определяются его свойства (рис. 12 и рис.13).

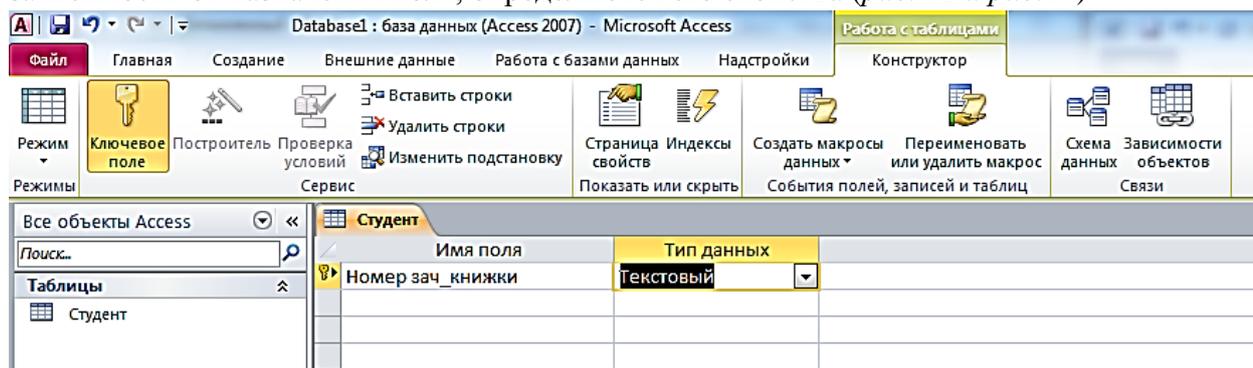


Рис. 12. Определения поля таблицы

Свойства поля	
Общие	Подстановка
Размер поля	255
Формат поля	
Маска ввода	
Подпись	
Значение по умолчанию	
Условие на значение	
Сообщение об ошибке	
Обязательное поле	Нет
Пустые строки	Да
Индексированное поле	Да (Совпадения не допускаются)
Сжатие Юникод	Нет
Режим ИМЕ	Нет контроля
Режим предложений ИМЕ	Нет
Смарт-теги	

Рис.13. Свойства поля для коррекции

Тип данного поля определяет область допустимых значений, которые могут быть представлены в этом поле. Ограничения на ввод допустимых значений могут быть представлены в свойстве **Размер поля** или определены пользователем в поле **Условие на значение**. Проверку на корректность вводимых данных можно определить с помощью простых выражений. В выражении могут использоваться арифметические операторы («+», «-», «^», «/», «\*»), операторы сравнения («>», «<», «>=», «<=», «=», «<>»), логические операторы (**AND** – логическое умножение; **OR** – логическое сложение, **NOT** – логическое отрицание, операторы **BETWEEN ...AND**, **IS**, **LIKE**). Пример определения свойства поля **Дата рождения** таблицы **Студент** представлен на рис. 14.

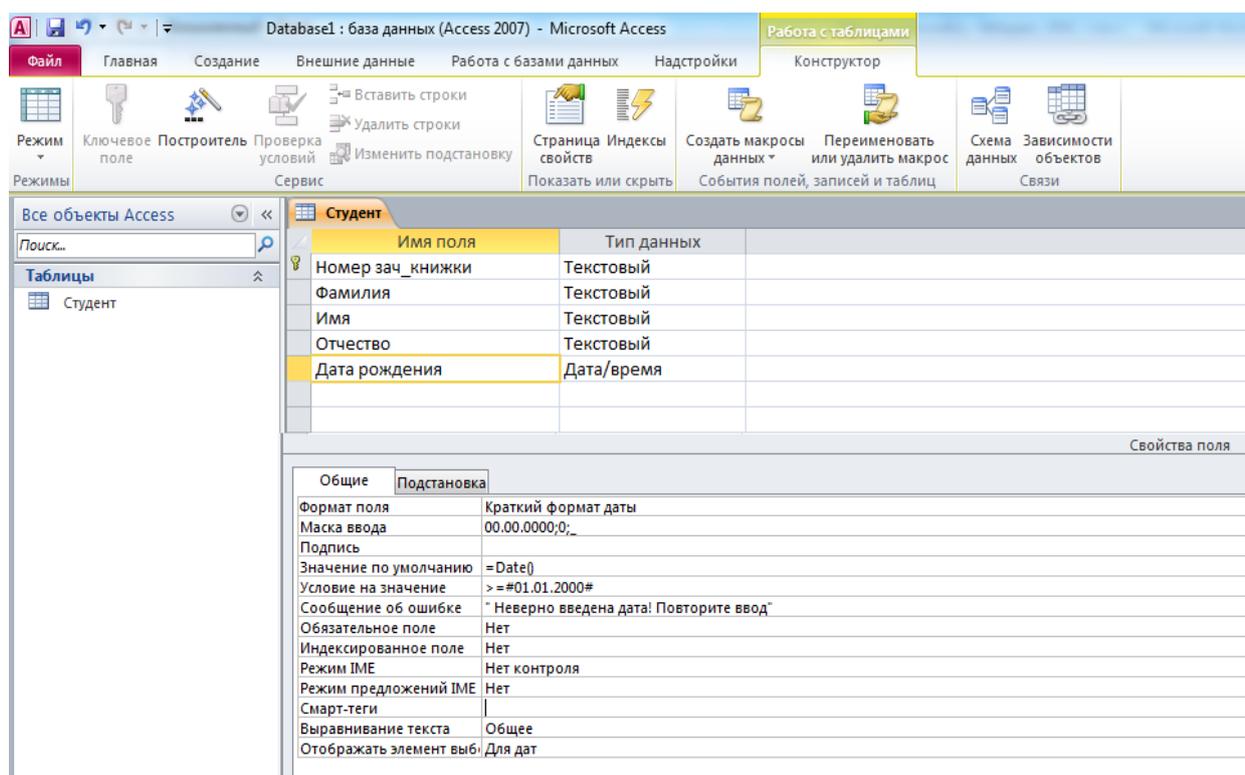


Рис. 14. Заполнение свойств поля

Свойство **Формат поля** определяет вид выводимых представлений даты, для удобства вводимых данных можно определить **Маску ввода** - шаблон заполнения данных. Встроенный *мастер заполнения* позволяет выбрать нужный шаблон.

В свойство **Значение по умолчанию** вводится встроенная функция *Date()*, основным назначением которой является вывод *текущей даты* в поле.

Свойство **Условие на значение** определяется пользователем и содержит ограничение по вводу дат в это поле. Оно определяет наименьший возраст студента, допустимый при приеме в вуз.

Поле **Сообщение об ошибке** заполняется самим пользователем. При неверном вводе значения в поле на экране появится именно это сообщение.

Реляционные БД отличаются от плоских баз данных, созданных в *Excel*, наличием специальных механизмов, поддерживающих принцип **целостности данных**.

За целостность данных в пределах одной таблицы отвечает поле, называемое **первичным ключом** и отвечающее следующим требованиям:

- а)** обеспечивает **уникальность** каждой записи (содержит *неповторяющиеся* значения);
- б)** не может иметь *Null-значений (пустых значений)*.

**Первичный ключ** – это поле или группа полей таблицы однозначно идентифицирующий каждую запись таблицы.

При создании любой таблицы в режиме конструктора, необходимо одно из определяемых полей выбрать **ключевым**. Как правило, ключевое поле располагается вверху списка полей

Для определения поля таблицы **ключевым** необходимо выбрать специальный значок на ленте, вкладка **Работа с таблицами** (рис. 15).

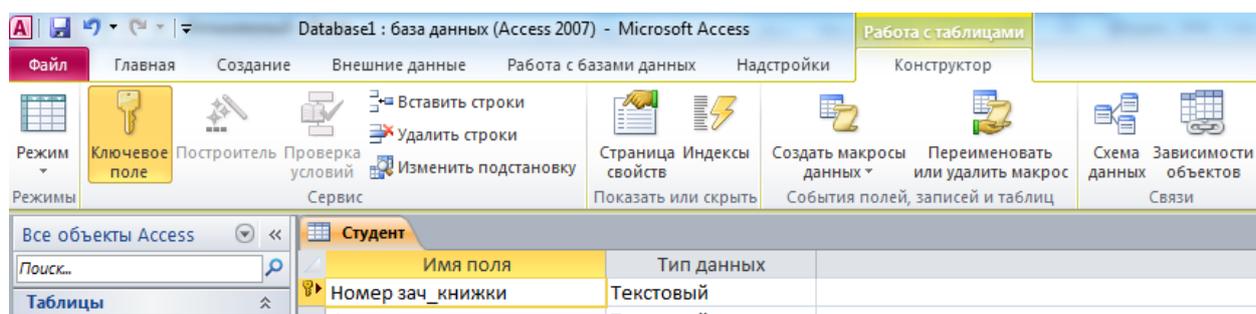


Рис. 15. Определение ключевого поля таблицы.

Базы данных предназначены для хранения информации по некоторой предметной области и содержат сведения об объектах реального мира, свойствах объектах и взаимосвязях между ними. Для того, чтобы построить модель реального мира с помощью набора таблиц, необходимо учитывать взаимоотношения реальных объектов – их связи между собой. В СУБД Access для обеспечения связи между таблицами вводится понятие **внешнего ключа**.

Прежде чем дать понятие *внешнего ключа*, дадим следующие ключевые определения.

*Таблица, содержащая первичный ключ и участвующая в связи с другими таблицами называется главной.*

*Таблица, с которой устанавливается связь, называется **связываемой**.*

***Внешним ключом** связываемой таблицы называется поле или группа полей этой таблицы, значения которых являются значениями первичного ключа главной таблицы.*

***Ограничение, по которому значения данного внешнего ключа должны быть адекватны значению соответствующего первичного ключа, называют ссылочным ограничением или ссылочной целостностью.***

Для того чтобы определить поле таблицы как внешний ключ, необходимо соблюсти ряд требований, а именно:

1. Проиндексировать данное поле – выставить свойство «индексированное поле» - *Да, допускаются совпадения.*
2. Тип данных и формат поля *внешнего ключа* связываемой таблицы должен совпадать с типом данных и форматом поля *первичного ключа* главной таблицы *связей*.
3. Если первичный ключ является составным (содержит несколько полей), то и внешний ключ будет составным.

Пример определения поля таблицы как поле внешнего ключа представлен на *рис. 16*.

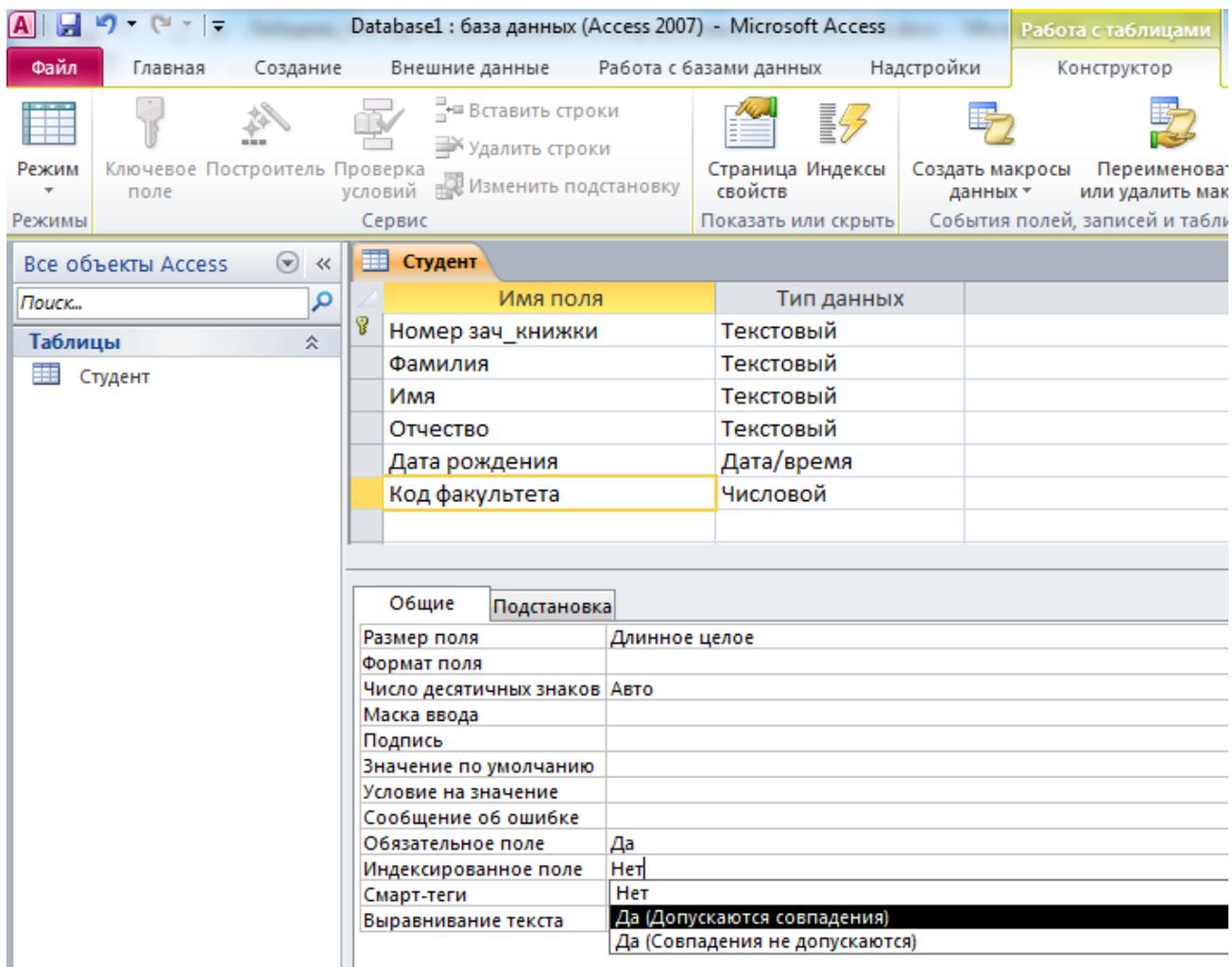


Рис. 16. Определение внешнего ключа в таблице

## ЗАДАЧИ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

1. Создайте свою БД; в качестве имени - своя фамилия.
2. Перенесите в СУБД ACCESS плоскую БД «Студент», созданную в EXCEL. Перейдите в режим конструктора таблицы и выполните следующие задания:
  - 2.1. Добавьте поле – *НомерЗачетки* и поместите его перед всеми остальными полями. Определите его как ключевое поле;
  - 2.2. Измените наименование поля *Факультет* на поле *КодФакультета*. определите его тип данных, как числовой;
  - 2.3. Добавьте поля *Наличие стипендии*, *Наличие детей* (тип данных – логический)
  - 2.4. Сохраните таблицу с именем *Студент*.
3. Создайте таблицу *Факультеты* с полями – *КодФакультета*, *НаименованиеФакультета*;
4. Создайте таблицу *Дисциплины* с полями: (*КодДисциплины* ключевое поле, *НаименованиеДисциплины*);
5. Создайте таблицу *Экзамен* с полями (*НомерЗачКнижки*, *КодДисциплины* – ключевое поле, *ДатаСдачи*, *Оценка*, *КоличествоЧасов*).
6. В таблице *Студент* отредактируйте данные в поле *КодФакультета* – вместо текстовых значений – числовые коды, адекватные значениям поля *КодФакультета* таблицы *Факультет*.

### ТЕМА3. ЛОГИЧЕСКАЯ МОДЕЛЬ БАЗЫ ДАННЫХ

*Логическая модель базы данных – это модель той части реального мира, информацию о которой надлежит хранить и обрабатывать в БД, используя конкретную СУБД.*

Данные о рассматриваемой предметной области структурируются, помещаются и хранятся в таблицах. Ранее рассматривалось, что таблица содержит **класс однотипных объектов** реального мира, обладающих одинаковыми свойствами, которые определяются в конструкторе таблицы, такие как название поля и тип данного поля. Дадим определение этому понятию. Согласно Дейту (*основоположник реляционной теории баз данных*), **класс однотипных объектов** будем называть **сущностями**.

Каждый **экземпляр** сущности содержит сведения о реальном объекте реального мира и представляется строкой таблицы или, иначе, *записью*.

*Логическая модель базы данных отражает сущности рассматриваемой предметной области и взаимосвязи между ними.*

Для того, чтобы отобразить отношения между объектами в рассматриваемой предметной области вводятся специальные понятия взаимоотношений – «*один-к-одному*», «*один-ко-многим*», «*многие-ко-многим*».

1. Связь «*один-к-одному*» (1:1) – это такой тип связи между сущностями А и В, когда каждому экземпляру сущности А соответствует один и только один экземпляр сущности В и, наоборот, каждому экземпляру сущности В соответствует один и только один экземпляр сущности А. Связь уникальна от сущности А к В и наоборот.

2. Связь «*один-ко-многим*» (1:M) – это такой тип связи между сущностями А и В, когда одному экземпляру сущности А может соответствовать 0, 1 или несколько экземпляров сущности В, однако каждому экземпляру сущности В соответствует один и только один экземпляр сущности А. Связь уникальна только в направлении от В к А.

3. Связь «*многие-ко-многим*» определяет такой тип связи между сущностями А и В, при котором каждому экземпляру сущности А может соответствовать 0, 1 или несколько экземпляров сущности В и наоборот. Связь неуникальна в обоих направлениях.

Для того чтобы построить логическую модель надо:

1.Перейти на вкладку **Работа с базами данных** и выбрать команду **Схема данных** (рис. 17).

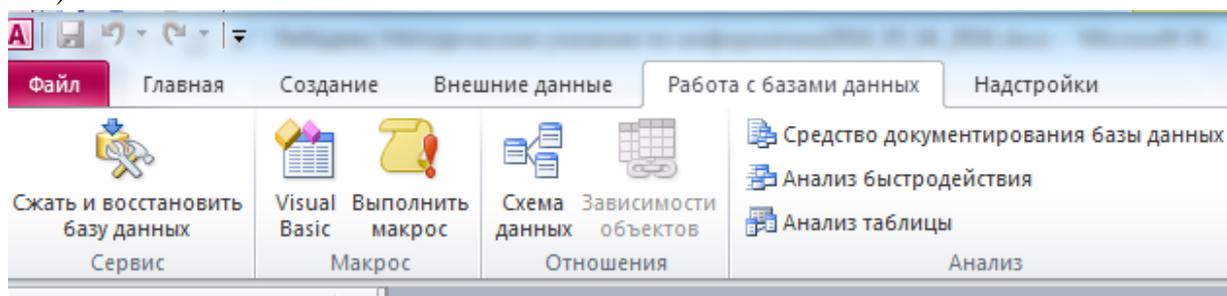


Рис. 17. Вкладка **Работа с базами данных**

2.В диалоговом окне «Добавление таблицы» выбрать все таблицы и нажать кнопку **Добавить**, либо можно добавить каждую из таблиц поочередно;

3.Для создания связи между таблицами надо выделить поле первичного ключа *главной* таблицы, нажать клавишу мыши и, не отпуская нажатую клавишу, перенести это поле на поле внешнего ключа *связываемой* таблицы. После этого обязательно надо выставить ссылочные ограничения, накладываемые на связываемые таблицы **Целостность данных** и **Каскадное обновление связанных записей** (рис.18).

Установка флажка **Целостность данных** означает, что значения внешних ключей связываемых таблиц должны соответствовать значениям первичных ключей главных таблиц.

Установка флажка **Каскадное обновление связанных полей** означает, что при изменении значений в первичных ключах соответствующие значения внешних ключей будут также изменены.

Установка флажка **Каскадное удаление связанных записей** означает, что при удалении значения первичного ключа соответствующие значения внешнего ключа также будут удалены.

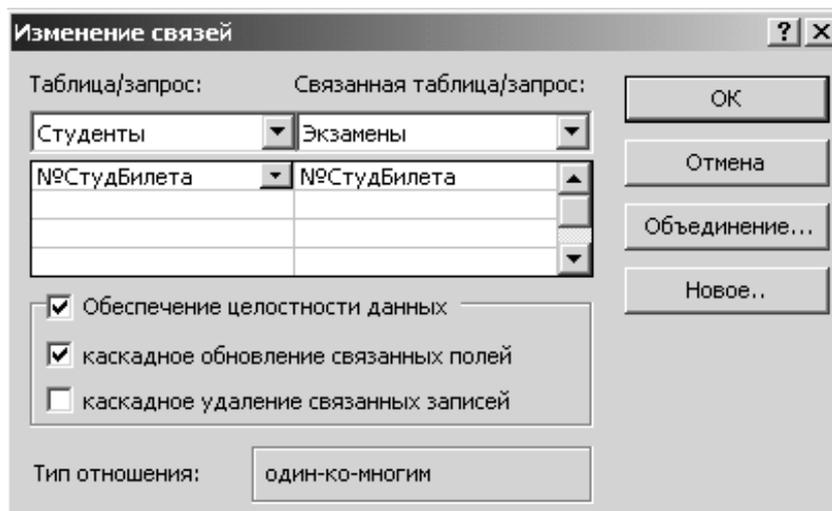


Рис. 18. Создание ссылочных ограничений

## ЗАДАЧИ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

Постройте логическую модель базы данных, исходя из следующих семантических соотношений:

1. В университете существует несколько факультетов, причем наименование каждого факультета уникально;
2. Каждый студент может обучаться только на одном факультете, но на каждом факультете может обучаться несколько студентов;
3. Каждый студент может сдать экзамен по разным дисциплинам, но одна и та же дисциплина может сдаваться разными студентами.

## ТЕМА. 4. ЗАПРОСЫ

Запрос позволяет выбрать необходимые данные из одной или нескольких взаимосвязанных таблиц, произвести вычисления и получить результат в виде *динамической* таблицы. В отличие от базовых (основных) таблиц БД, *таблицы запросов* называются динамическими потому, что данные, содержащиеся в этих таблицах, могут каждый раз меняться в зависимости от условий выборки при выполнении запроса.

Средствами запроса можно выполнить следующие функции:

- выбрать записи, удовлетворяющие условиям отбора;
- включить в результирующую таблицу нужные поля и произвести необходимые вычисления;
- сгруппировать записи с одинаковыми значениями в одном или нескольких полях и выполнить над ними групповые функции по подмножеству (*sum* – сумма, *count* – подсчет количества записей, *average* – вычисление среднего);
- создать новую таблицу БД, используя данные из существующих таблиц и т.д.

В зависимости от выполняемых функций все запросы делятся на две группы: *запросы–выборки* и *запросы–действия*.

**Запросы–выборки** осуществляют выборку данных из одной или нескольких взаимосвязанных таблиц и (или) других запросов в соответствии с заданными условиями. Результатом является таблица, которая существует до закрытия запроса. На основе такого запроса могут строиться запросы других видов. Таблицу с результатами запроса на выборку можно использовать для работы с данными в таблицах, на которых построен запрос. Например, через таблицу запроса можно откорректировать данные сразу в нескольких исходных таблицах базы данных. Запросы могут служить источниками записей для форм, отчетов и страниц доступа к данным

Существует несколько видов запросов этой группы:

- *запрос по условию* – позволяет производить отбор записей в соответствии с заданным критерием поиска;
- *перекрестный запрос* – отображает итоговые данные в виде *кросс–таблицы* (сводная таблица с группировкой данных и подведением итогов по строкам и столбцам);
- *запрос с параметром* отображает свое собственное диалоговое окно, запрашивающее информацию. Критерий отбора для ввода в диалоговое окно задается как условие выборки;
- *запрос с вычисляемым полем* – позволяет создать поле, несуществующее в базе данных (как правило, для произведения дополнительных расчетов);
- *запрос группировкой данных и подведением итогов* – применяется для подведения промежуточных или окончательных итогов на основе одной или нескольких таблиц.

**Запросы–действия** позволяют модифицировать данные в таблицах: *удалять, обновлять, добавлять записи*.

К этой группе запросов относятся следующие:

- *запросы на создание* таблицы применяются для создания новой таблицы на основании данных, содержащихся в одной или нескольких таблиц;

- *запросы на добавление* записей позволяют добавлять записи в таблицу;
- *запросы на обновление* изменяют значения одного или нескольких полей таблицы соответствии с заданным критерием обновления;
- *запросы на удаление* удаляют записи из одной или нескольких таблиц.

## ЗАПРОСЫ НА ВЫБОРКУ

**Выборкой** называется отбор данных из одной или нескольких взаимосвязанных таблиц в зависимости от условий отбора. Результатом выполнения запроса является динамическая таблица, позволяющая изменять данные базовых таблиц.

### Технология создания

Решим следующую задачу «*Отобразить всех студентов с одного факультета*». Для этого:

1. Перейдите на вкладку **Создание** (рис. 19);

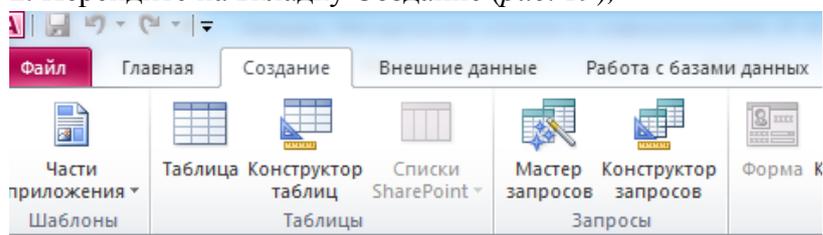


Рис. 19. Создание запроса

2. Выберите команду **Конструктор запросов**;

3. Добавьте таблицу **Студент** в верхний бланк запроса (рис. 20);

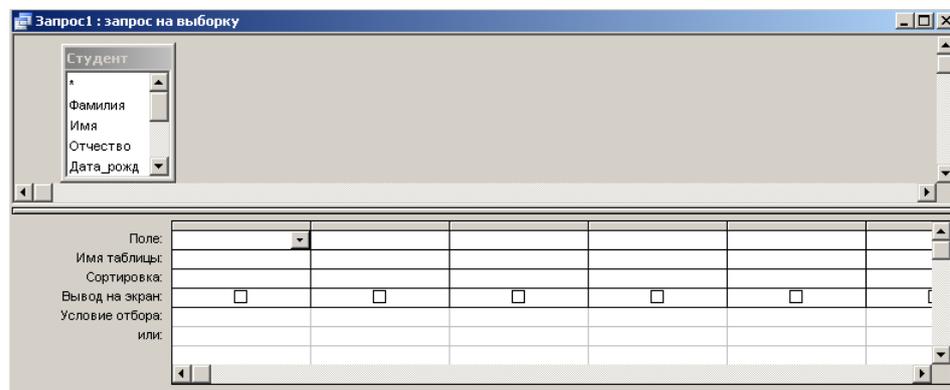


Рис. 20. Конструктор запроса

Конструктор запроса состоит из двух частей: в верхнюю часть добавляются таблицы, определяются свойства запроса, устанавливаются типы соединения таблиц; нижняя часть называется запросом по образцу (*Query By Example*) или *QBE*-запросом. Основная работа происходит именно с *QBE*-запросом.

4. Добавьте в *QBE*-запрос поля «*Фамилия*», «*Имя*», «*Отчество*», «*Факультет*» (либо перетаскивая поля в бланк запроса, либо выделяя их двойным щелчком мыши);

5. Под полем **Факультет** в строку «Условие отбора» введите название факультета «ИБК» (рис. 21).

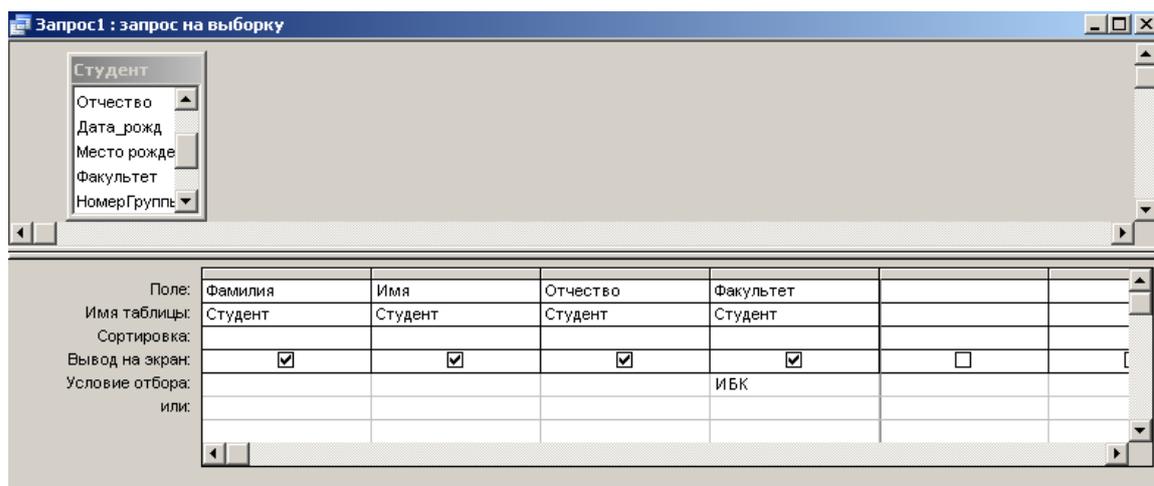


Рис. 21. Создание запроса на выборку

Условия отбора записей могут задаваться для одного или нескольких полей таблицы. Условием отбора является *выражение*, которое состоит из, *операторов сравнения* и *операндов*.

В качестве операндов выражения могут использоваться: *литералы*, *константы*, *идентификаторы (ссылки)*.

**Литералами** являются конкретные значения, воспринимаемые СУБД Access так, как они записаны. В качестве литералов могут быть использованы числа, строковые значения, даты. Текстовые строки заключаются в двойные кавычки, даты – в знаки «#». Например: 567, «Информатика», #1–Января–2011#.

**Константами** являются неизменяющиеся значения, которые определены в Access, например, *Истина (True)*, *Ложь (False)*, *Null*.

**Идентификатор** представляет собой ссылку на значение поля, элемента управления или свойства. Идентификаторами могут быть имена полей таблиц, форм, отчетов и т.д. Они должны заключаться в квадратные скобки. Операторами сравнения и логическими операторами, использование которых допускается в выражении условия отбора, являются: =, <, >, <>, <=, >=, *Between*, *In*, *Like*, *And*, *Or*, *Not*.

Условия отбора записей могут задаваться для одного или нескольких полей таблицы.

– если условия отбора под несколькими полями пишутся в одной строке, в этом случае они объединяются логическим оператором **AND** и запрос выполнится тогда и только тогда когда выполнится каждое из условий;

– если условия отбора под несколькими полями пишутся в разных строках (например, второе условие под первым), в этом случае они объединяются логическим оператором **OR** и запрос выполнится тогда, когда выполнится хотя бы одно из условий.

6. Перейдите в режим просмотра запроса, нажав на кнопку **Режим** на ленте и выбрав

команду **Режим таблицы** (рис. 22), либо выбрать команду **Выполнить**  **Выполнить**.

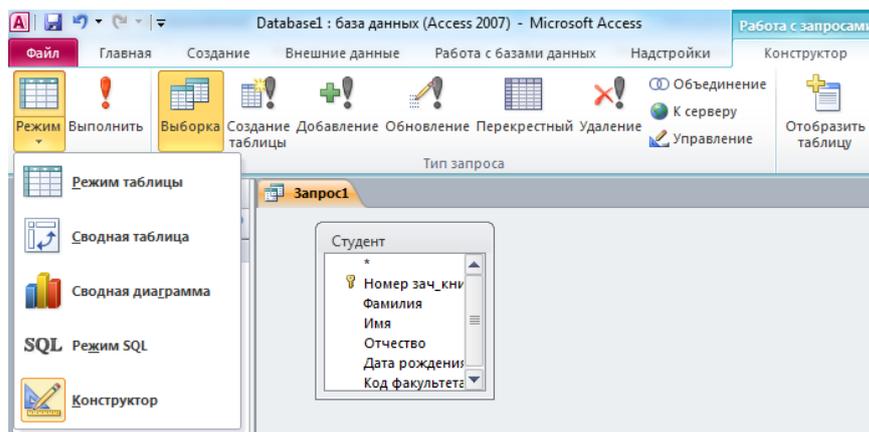


Рис. 22. Запуск запроса для просмотра

Отобранные записи из таблицы представлены на рис. 23.

Фамилия	Имя	Отчество	Факультет
Первакова	Анастасия	Вадимовна	ИБК
Маралев	Кирилл	Александрович	ИБК
Гронская	Анна	Александровна	ИБК
Веселова	Юлия	Сергеевна	ИБК
Погосов	Евгений	Анатольевич	ИБК
Кудрявцев	Павел	Сергеевич	ИБК
Оганина	Елизавета	Владимировна	ИБК
Яковлева	Екатерина	Дмитриевна	ИБК
Эгамбердиева	Эльвира	Дусмирзаевна	ИБК
Прохоренко	Анна	Леонидовна	ИБК
Корогодин	Владимир	Сергеевич	ИБК

Рис. 23. Режим просмотра запроса

7. Сохраните запрос под именем «Студенты ИБК»

## ЗАДАЧИ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

1. Отобрать всех студентов с одного факультета.
  2. Отобрать всех студентов с одного факультета и одного года рождения.
  3. Отобрать всех студентов с одного факультета, одного курса и дневной формы обучения.
  4. Отобрать всех студентов с одного факультета, из одной группы, из одного города и одного года рождения.
  5. Отобрать всех студентов одного года рождения (используйте оператор *Between #data1# and #data2#*).
  6. Отобрать всех студентов одного года рождения (в условие отбора введите функцию *Year(ПолеДаты)=год*).
- Примечание. Функция Year(аргумент) применяется к полям, содержащие даты (формат дата/время) для возвращения года от даты, например, Year(#01.01.01#) вернет значение – 2001.*
7. Отобрать всех студентов, чья фамилия начинается на определенную букву (используйте оператор like “Символ\*”).
  8. Отобрать всех студентов из Санкт–Петербурга или Москвы.

## ТЕМА. 5. ВЫЧИСЛЯЕМЫЕ ПОЛЯ В ЗАПРОСАХ

Иногда в запросах необходимо производить вычисления для одного или нескольких полей. С этой целью создаются **вычисляемые поля**.

Вычисляемое поле – это поле, которое не существует ни в одной из базовых таблиц БД.

Создается для проведения дополнительных расчетов по базе данных. При создании поля необходимо правильно определять его название, оно не должно совпадать ни с одним из названий полей таблиц, включаемых в запрос, иначе может возникнуть ошибка выполнения запроса – циклическая ссылка.

### Технология создания:

- создайте запрос на выборку на основе одной или нескольких взаимосвязанных таблиц;
- добавьте необходимые поля в запрос;
- в пустом поле бланка запроса введите новое имя поля и после него поставьте двоеточие;
- через контекстное меню вызовите команду **Построить**;
- используя интерфейс построителя выражений (рис. 24) построить необходимые для решения задачи выражения.

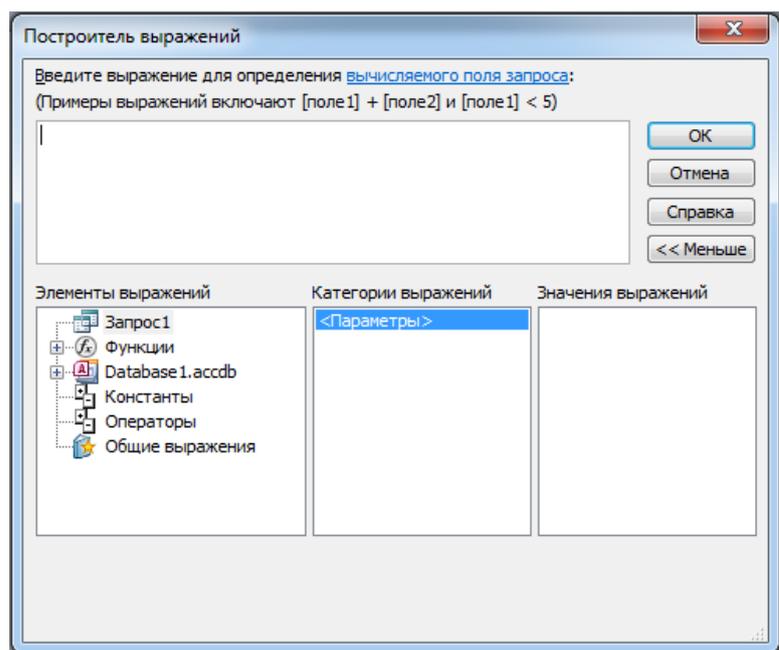


Рис. 24. Окно построителя выражений

## ЗАДАЧИ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

1. Создать вычисляемое поле **ФИО**, объединяющее данные из трех полей – Фамилия, Имя, Отчество.

1.1. Для объединения полей использовать оператор конкатенации - &

1.2. При объединении предусмотреть пробелы (использовать как символ, заключенный в кавычки)

2. Вычислить возраст студентов. (Использовать функции **YEAR** (год) и **Date** (текущая дата))

## ПРИМЕНЕНИЕ ФУНКЦИИ IIF В ВЫЧИСЛЯЕМОМ ПОЛЕ

Функция **IIF** подобна функции ЕСЛИ в *EXCEL*.

**Immediate if** – немедленное вычисление значения ЕСЛИ для каждой строки таблицы.

Синтаксис:

**Iif** («*expression*»; «*truepart*»; «*falsepart*»)

где:

**expression** – любое *выражение отношения* либо выражение, использующее логические *операторы AND, OR, NOT*. И в том и другом случае будет возвращаться логическая константа либо *True (истина)* либо *False (ложь)*;

"«**truepart**» – (*значение если истина*), выражение, выполняющееся по истинности условия;

«**falsepart**» – (*значение если ложь*), выражение, которое выполняется, когда условие не выполняется (ложно).

В качестве примера рассмотрим решение следующей задачи.

**Задача:**

*Увеличить размер стипендии студентам по следующему правилу «Студентам очной формы обучения – в два раза, заочной формы обучения – 500 руб., вечерней формы обучения – в полтора раза».*

**Решение:**

1. Создайте запрос на выборку;
2. Добавьте в бланк запроса следующие поля – **Фамилия, Имя, Отчество, ФормаОбучения, Стипендия**;
3. В первом пустом столбце введите имя **Новая стипендия** и поставьте двоеточие (рис. 25);

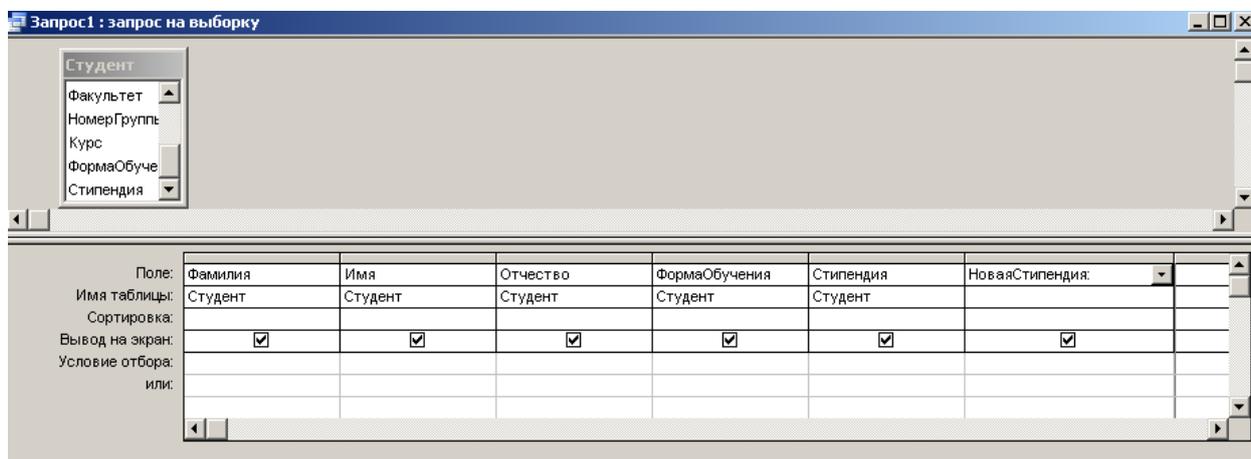


Рис. 25. Создание вычисляемого поля

4. Выберите команду из контекстного меню команду **Построить** (рис. 26);

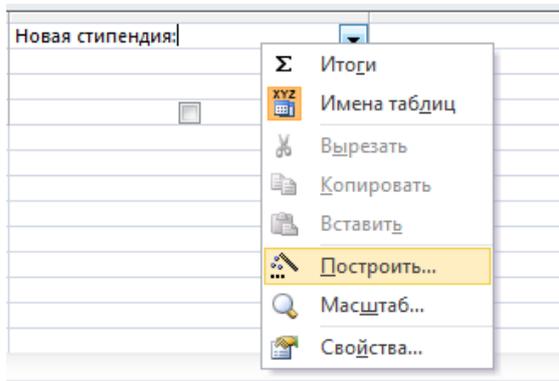


Рис. 26. Вызов построителя выражений

5. В открывшемся построителе выражений (рис. 27) выбрать команду **Встроенные функции**, далее из третьего столбца выбрать функцию **ИФ**.

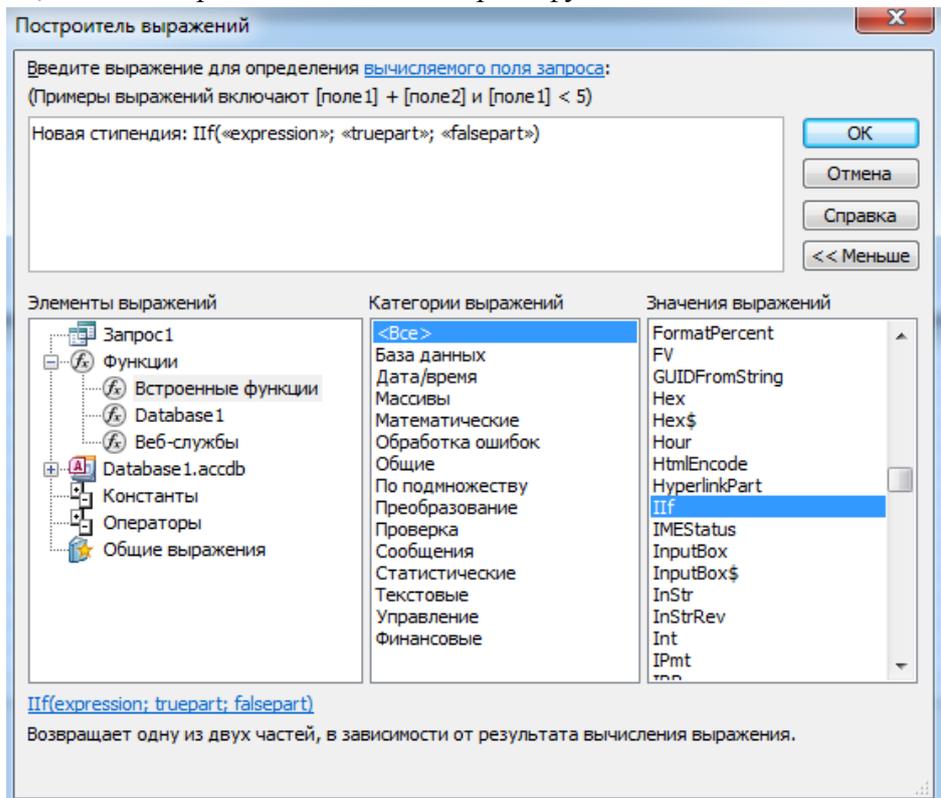


Рис. 27. Построитель выражений

6. Ввести следующее выражение (рис. 28):

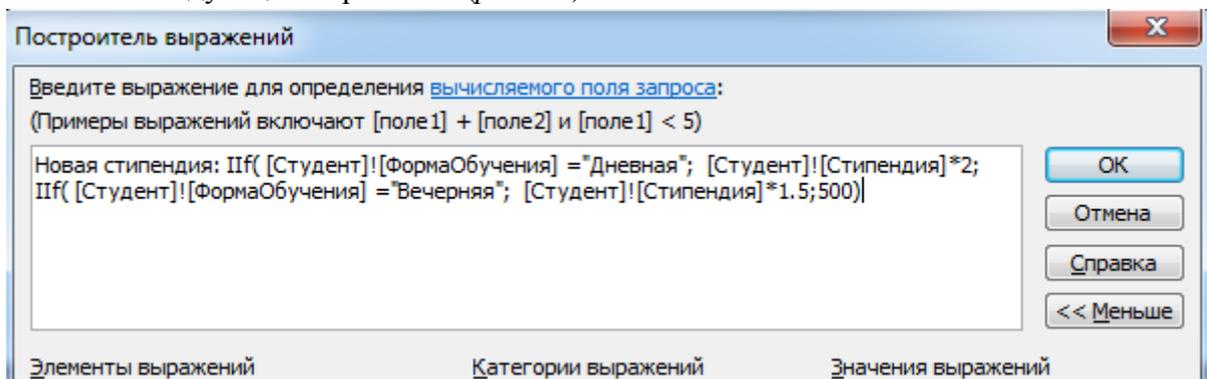


Рис. 28. Расчет стипендии с использованием функции **ИФ**

На рис. 29 представлены результаты работы функции

Запрос 1 : запрос на выборку						
Фамилия	Имя	Отчество	ФормаОбучени	Стипендия	НоваяСтипенд	
▶ Первакова	Анастасия	Вадимовна	Дневная	900	1800	
Маралев	Кирилл	Александрович	Дневная	900	1800	
Гронская	Анна	Александровна	Дневная	900	1800	
Веселова	Юлия	Сергеевна	Дневная	900	1800	
Погосов	Евгений	Анатольевич	Дневная	900	1800	
Кудрявцев	Павел	Сергеевич	Дневная	900	1800	
Оганина	Елизавета	Владимировна	Дневная	900	1800	
Яковлева	Екатерина	Дмитриевна	Заочная	0	500	
Эгамбердиева	Эльвира	Дусмирзаевна	Заочная	0	500	
Прохоренко	Анна	Леонидовна	Дневная	900	1800	
Александров	Алексей	Андреевич	Дневная	900	1800	
Татаринцева	Дарья	Владимировна	Дневная	900	1800	
Лиепиньш	Дана	Эдуардовна	Дневная	900	1800	
Линкевич	Даниил	Сергеевич	Дневная	900	1800	
Бурчалова	Анастасия	Александровна	Дневная	900	1800	
Степанова	Ольга	Олеговна	Дневная	900	1800	
Огурцова	Татьяна	Валентиновна	Дневная	900	1800	
Зинченко	Яна	Сергеевна	Дневная	900	1800	
Матвеева	Юлия	Викторовна	Дневная	900	1800	
Розен	Алевтина	Константиновна	Дневная	900	1800	
Малков	Дмитрий	Владимирович	Заочная	0	500	
Клинина	Дарья	Андреевна	Вечерняя	500	750	
Малышева	Яна	Борисовна	Вечерняя	500	750	
Смирнова	Юлия	Викторовна	Заочная	0	500	
Платонова	Наталья	Михайловна	Вечерняя	500	750	
*						

Запись: 1 из 25

Рис. 29. Результат работы вычисляемого поля

## ЗАДАЧИ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

1. Создать вычисляемое поле, увеличивающее размер стипендии студентам при выполнении следующих условий: *если студент имеет детей и имеет стипендию – стипендию увеличить в 2 раза; если студент не имеет детей, но имеет стипендию – в 1,5 раза; если студент имеет детей, но не имеет стипендию – 1000; если не имеет детей и не имеет стипендии – 500 руб.*

2. Объединить данные из полей Факультет, Курс, ГодРождения.

## ТЕМА. 6. ЗАПРОСЫ НА ГРУППИРОВКУ ДАННЫХ И ПОДВЕДЕНИЕ ИТОГОВ

Групповые запросы служат для группировки данных, содержащихся в одном или нескольких полях запроса и подведения по этим данным промежуточных или окончательных итогов.

Исходные данные в запросе могут быть сгруппированы по одинаковым значениям полей, и к ним можно применить различные агрегатные функции. К агрегирующим функциям относятся **sum** (сумма), **avg** (среднее значение), **max** (максимум), **min** (минимум), **count** (подсчет количества).

При выборе агрегирующей функции по подмножеству следует учитывать следующее:

1. Функции **sum** и **avg** не применяются к полям с текстовым и логическим типами данных.
2. Функция **count** работает с полями любого типа.

### Технология создания группового запроса:

1. Создайте запрос на выборку;
2. Добавьте в него одну или несколько таблиц, в зависимости от условия задачи;

3. Выберите нужные поля, удовлетворяющие условиям отбора, при этом исключите поля с неповторяющимися данными или поля, не участвующие в вычислениях ;

4. Выберите на панели инструментов команду «Групповые операции» -  $\Sigma$  Итоги (рис. 30).

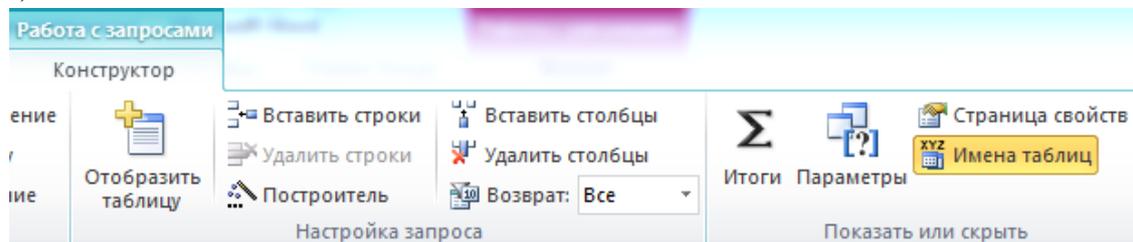


Рис. 30. Выбор команды группировки данных

5. Сгруппируйте записи с одинаковыми значениями в одном или нескольких полях (оставить под полями слово «группировка»);

6. В полях, предназначенных для вычислений, измените слова «Группировка» на групповую функцию по подмножеству, выбрав соответствующую функцию из списка;

7. Выполнить запрос

8. Сохранить запрос под именем задачи.

В качестве примера решим следующую задачу:

*Подсчитать общее количество студентов, общую сумму стипендии и среднюю сумму стипендии студентов одного из факультетов.*

**Решение:**

1. Создадим запрос на выборку;

2. Добавим в бланк запроса таблицу **Студент**;

Отберем из таблицы следующие поля – **Фамилия** – для подсчета количества студентов, **Стипендия** – для суммирования и вычисления среднего (рис. 31).

Результат работы запроса – рис. 32.

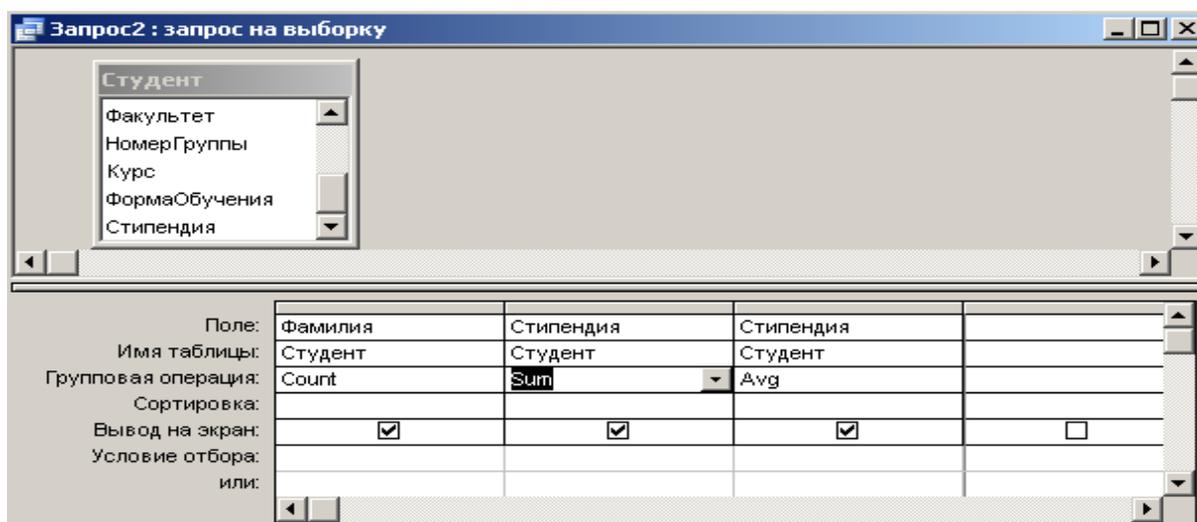


Рис. 31. Создание группового запроса

Запрос2 : запрос на выборку			
	Count-Фамилии	Sum-Стипенди	Avg-Стипендия
▶	25	17700	708

Рис. 32. Результат работы группового запроса

### ЗАДАЧИ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

1. Подсчитать количество студентов с одного факультета.
2. Подсчитать количество и общую стипендию студентов одного факультета и одного года рождения.
3. Подсчитать количество и общую стипендию студентов с одного факультета, одного курса и дневной формы обучения.
4. Подсчитать количество и среднюю стипендию студентов с одного факультета, из одной группы, из одного города и одного года рождения.
5. Подсчитать общую сумму стипендии студентов одного факультета.
6. Вычислить среднюю сумму стипендии у студентов, имеющих детей.
7. Вычислить во сколько раз расчетная стипендия (см. задачу на вычисляемые поля) больше запланированной стипендии (результат предоставить в процентах).
8. Вычислить общую сумму стипендии по всем факультетам и процентную долю стипендии каждого студента по отношению к общей сумме (в запросе работать с вычисляемым полем (см. задачу на вычисляемые поля)).
9. Подсчитать количество студентов, имеющих детей. Если их количество больше пяти человек, увеличить общий размер стипендиального фонда в два раза.
10. Вычислить средний возраст студентов по факультетам; по всей базе данных.

## ТЕМА. 7. ФОРМЫ

Формы используются для просмотра и корректировки данных в базе данных. Источником записей формы служат либо таблица, либо запрос, объединяющий одну или несколько взаимосвязанных таблиц. Если БД предусматривает сетевой вариант работы, то в качестве источника записей формы желательно использовать *sql*-строку, поскольку в этом случае работа с данными формы происходит значительно быстрее.

### 7.1. Разделы формы

Сведения в форме могут быть разбиты на разделы. При создании формы «вручную» (не используя ни мастер форм, ни автоформы), по умолчанию, любая форма будет содержать раздел данных. Помимо этого раздела в форму можно добавить разделы заголовка, верхнего колонтитула, нижнего колонтитула и примечания. Каждый раздел формы имеет свое предназначение и печатается в определенном порядке.

1.1. В заголовке формы отображаются сведения, которые остаются неизменными для каждой записи, такие как название формы. Заголовок формы выводится сверху на экране в режиме формы и сверху на первой странице при печати формы.

1.2. В верхнем колонтитуле отображаются сведения, такие как название формы и заголовки столбцов, которые печатаются сверху на каждой печатной странице. Верхние колонтитулы выводятся только в печатных формах.

1.3. В области данных отображаются записи таблицы или запроса, которые выбраны в качестве источника данных для формы. Существуют три способа отображения данных на экране:

а) форма создана как **одионочная форма** – в этом случае, в режиме просмотра формы пользователь увидит только одну запись (первую), остальные записи можно будет просмотреть осуществляя переход по записям;

б) форма создана как **ленточная форма** – в этом случае на просмотр будут выведены все записи таблицы или запроса. Заголовки столбцов формы будут помещены в раздел заголовка формы, поля с данными будут помещены в область данных. Размеры столбцов(ширина и высота) будут определяться из типов данных, заданных пользователем;

в) форма создана как **табличная форма** – данные будут отображены в виде таблицы с фиксированной шириной столбцов.

1.4. В верхнем колонтитуле могут быть отображены такие сведения, как дата или номера страниц.

1.5. В примечании формы, как правило, выводятся комментарии для пользователя или инструкции по работе с формой. Очень удобно использовать этот раздел для создания вычисляемых полей с целью подведения итогов по таблице или запросу.

### 7.2. Способы создания формы

Существуют три способа создания формы:

- с использованием «Автоформы»;
- с использованием «Мастера форм»;
- «вручную», используя «Конструктор форм».

1. **Автоформа** – это «минимастер». Она автоматически строит простые формы на основании имеющихся в базе данных таблиц. Для создания формы таким способом необходимо:

1. Выделить объект, для которого нужно создать форму;
2. Нажать кнопку Форма вкладки Создание (рис. 37)

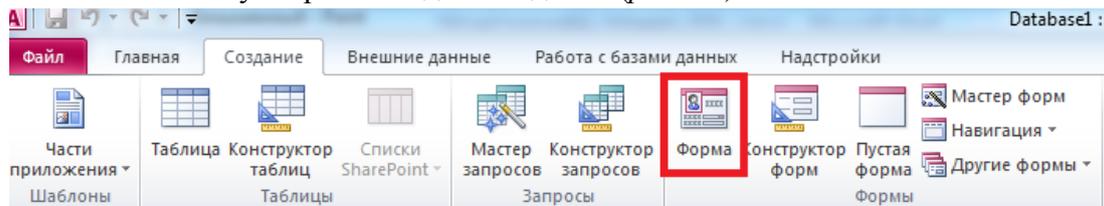


Рис. 33. Запуск Автоформы

3. Автоматически будет создан макет формы, который впоследствии можно доработать (рис. 34);

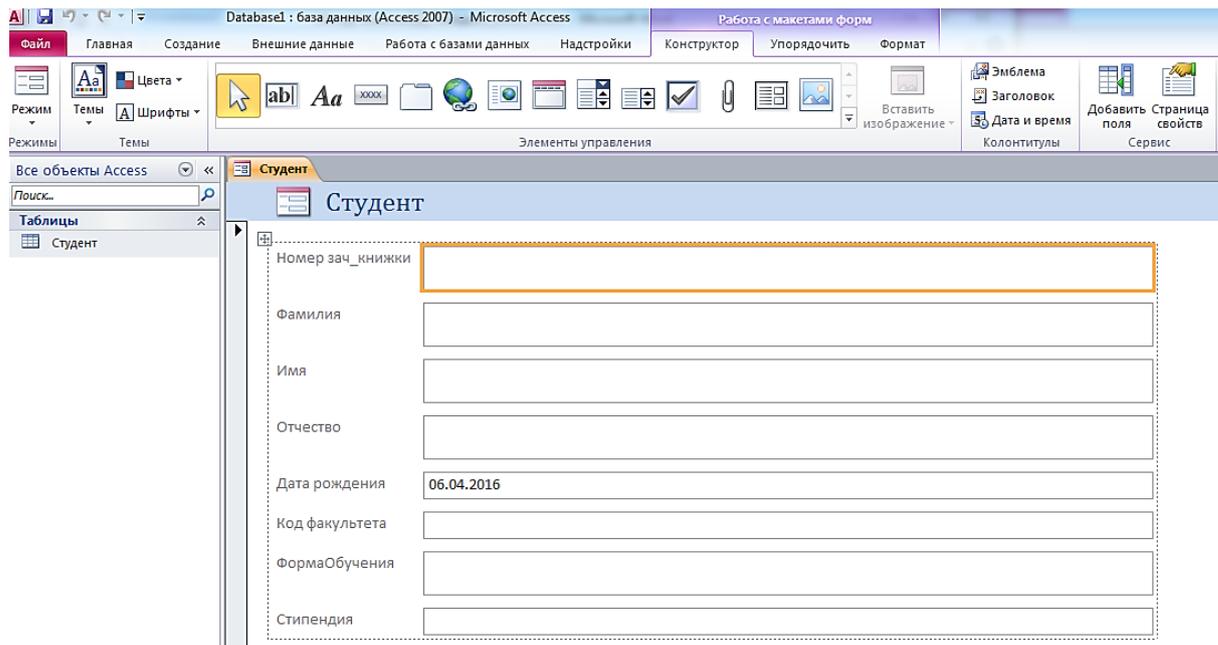


Рис. 34. Создание формы командой *Автоформа*

## 2. Создание формы с помощью *мастера форм*.

Мастер форм предлагает создание формы по шаблону либо с использованием различных стилей оформления формы. Для создания формы таким способом необходимо:

1. Запустить команду *Мастер форм* вкладки *Создание* рис.35;

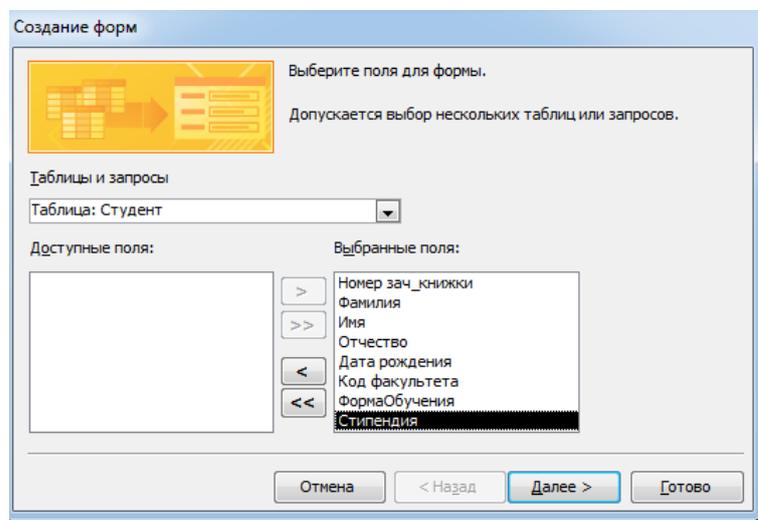


Рис. 35. Мастер форм, шаг первый

2. На следующем шаге определяется выбор представления формы (рис. 36);

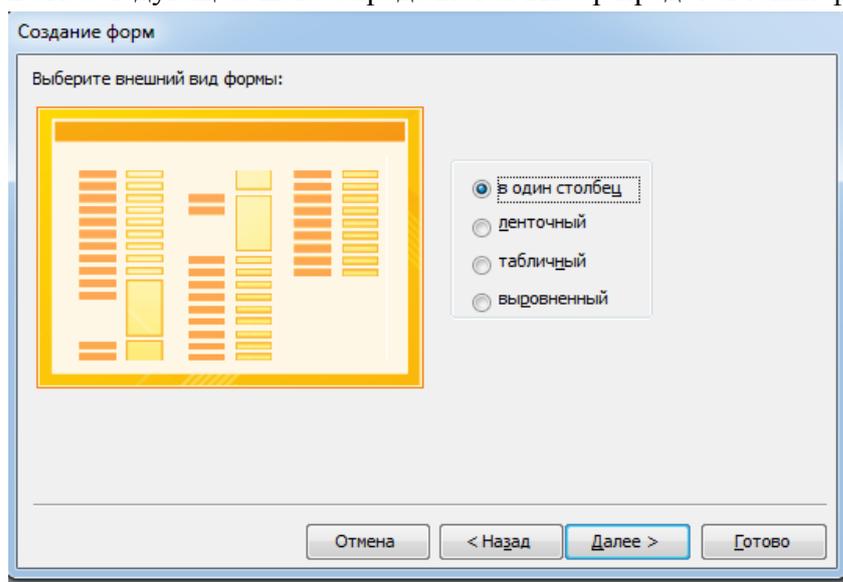


Рис. 36. Выбор вида представления формы

3. На последнем шаге мастера вводится имя формы и выбирается дальнейший режим работы с ней, либо открыть для просмотра, либо продолжить работу в режиме конструктора.

### 3. Создание формы в *режиме конструктора*.

Для этого:

1. Запустить команду **Конструктор форм** вкладки **Создание**;
2. Используя свойства формы (рис. 37), заполнить источник записей формы, настроить вид представления формы (простая, ленточная, табличная);

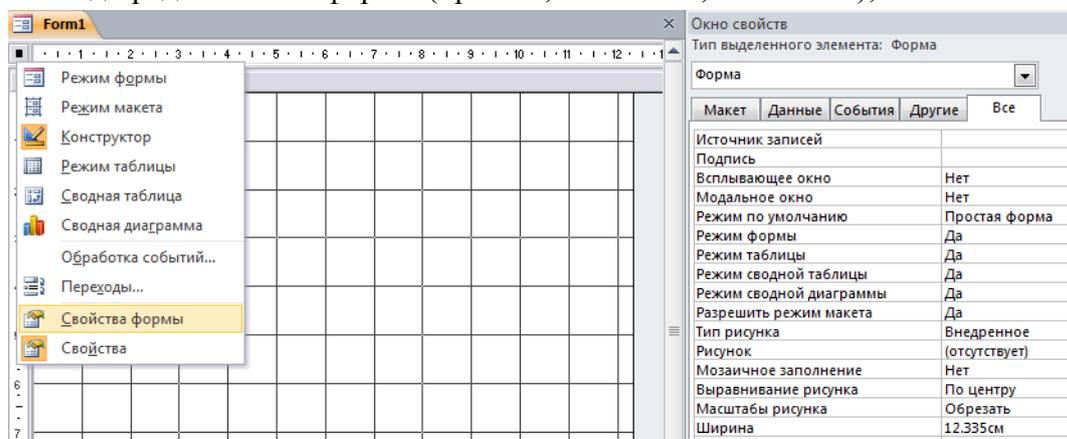


Рис. 37. Вызов свойств формы

3. Используя панель элементов, добавить необходимые элементы в форму.

### 3. Общие сведения об элементах управления

*Элементы управления* – это объекты формы или отчета, которые служат для вывода данных на экран, выполнения макрокоманд или оформления формы или отчета. Например, поле можно использовать для вывода данных на экран в форме или отчете, кнопку – для открытия другой формы или отчета, а линию или прямоугольник – для разделения и группировки элементов управления с тем, чтобы они лучше воспринимались пользователем.

В *Microsoft Access* существуют следующие типы элементов управления, которые содержатся на панели *элементов управления* вкладки *Инструменты конструктора форм* (рис.38).

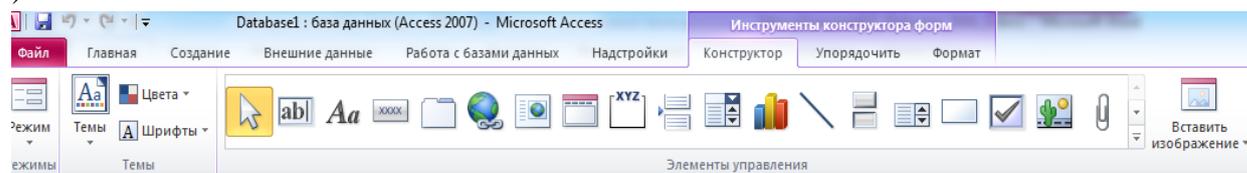


Рис. 38. Элементы управления формы

Это – *поле, надпись, группа, переключатель, флажок, выключатель, поле со списком, список, кнопка, рисунок, присоединенная рамка объекта, свободная рамка объекта, набор вкладок, подчиненная форма/отчет, разрыв страницы, линия, прямоугольник и дополнительные элементы ActiveX.*

Элементы управления могут быть связанными, свободными или вычисляемыми.

- *Связанный элемент управления* присоединен к полю базовой таблицы или запроса. Такие элементы управления используются для отображения, ввода или обновления значений из полей базы данных.

- Для *вычисляемого элемента управления* в качестве источника данных используется выражение. В выражении могут быть использованы данные из поля базовой таблицы или запроса для формы или отчета, а также данные другого элемента управления формы или отчета.

- Для *свободного элемента управления* источника данных не существует. Свободные элементы управления используются для вывода на экран данных, линий, прямоугольников и рисунков.

#### 7.4. Создание формы на базе нескольких таблиц

Использование мастера форм – это наиболее простой и быстрый способ создания формы, содержащей данные из нескольких таблиц. Мастер форм ускоряет процесс создания форм, выполняя автоматически всю основную работу. В первом окне мастера форм можно выбрать поля, которые должны быть включены в форму. Это могут быть поля из одной или нескольких таблиц. При использовании мастера форм для создания формы на основе нескольких таблиц создается инструкция *SQL*. Эта инструкция содержит сведения о том, какие таблицы и поля нужно использовать.

С помощью мастера можно создать обычную или составную форму, отображающую данные из нескольких таблиц. Составной является форма, содержащая одну или несколько *подчиненных форм*. Подчиненные формы используются для показа данных из таблиц, связанных друг с другом отношением «*один–ко–многим*».

Имеется возможность представления данных из связанных таблиц без использования подчиненной формы. Например, если в форме много элементов управления, для подчиненной формы может не остаться места. В этом случае можно использовать мастер форм для создания связанных форм. При нажатии кнопки в форме открывается другая форма, связанная с записью в главной форме отношением «*один–ко–многим*».

## ЗАДАНИЕ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

1. Создать справочники «Факультеты», «Предметы(Дисциплины)», используя команду «Автоформа»;
2. Добавить разделы «Заголовок» и «Примечание формы»;
3. Используя элемент управления «Надпись» ввести заголовки справочников;
4. В заголовках форм создать кнопку закрытия формы;
5. В примечании форм создать вычисляемые поля для подсчета записей в формах.  
Чтобы сделать это на панели элементов нужно:

- 5.1 На панели элементов выбрать элемент управления – поле – .
- 5.2. Отрисовать поле в примечании формы (сверху вниз, слева направо).
- 5.3. Выделить поле одним щелчком мыши;
- 5.4. Нажать правую кнопку мыши не снимая фокус с поля;
- 5.5. В открывшемся контекстном меню выбрать свойства поля.
- 5.6. В свойствах перейти на вкладку – **Все**, выбрать свойство – *Данные*;
- 5.6. Запустить построитель выражения .
- 5.7. Ввести выражение для подсчета количества записей (*количество считается с помощью той же функции, что и в запросе*). Для введения выражения в поле использовать только построитель выражений(!).

6. В примечании формы также создать кнопки перехода по записям – первая, последняя, предыдущая, последующая, используя элемент управления – *кнопка*.

7. Создать составную форму, для этого в качестве главной формы выбрать справочник «Факультеты», для подчиненной формы выбрать таблицу «Студенты». Для этого:

- 7.1. Запустить мастера форм, на первом шаге выбрать все поля из таблиц «Факультеты» и «Студенты»;
- 7.2. Следующий шаг – вид представления данных – подчиненная форма (щелкнуть по главной таблице);
- 7.3. Следующий шаг – внешний вид подчиненной формы – табличный;
- 7.4. Следующий шаг – выбрать стиль оформления главной формы;
- 7.5. Завершить создание составной формы с помощью мастера;

8. Войти в режим конструктора формы и отредактировать форму, задав заголовок формы, необходимые кнопки управления, отформатировать поля и надписи полей, придав им единый размер.

## ТЕМА. 8. ОТЧЕТЫ

Отчет – это гибкое и эффективное средство для организации данных при выводе на печать. С помощью отчета имеется возможность вывести необходимые сведения в том виде, в котором требуется. В отличие от форм, отчеты не позволяют вводить данные.

### *Разделы отчета*

**1.1. Заголовок отчета** печатается один раз в начале отчета и может содержать эмблему, название и дату печати отчета. Заголовок печатается на первой странице отчета перед верхним колонтитулом.

**1.2. Верхний колонтитул** печатается в верхней части каждой страницы отчета и может содержать заголовки столбцов.

**1.3. Заголовок группы** печатается в начале каждой новой группы записей и может содержать имя группы и другие сведения, применимые ко всей группе целиком.

**1.4. Область данных** содержит основную часть данных, которые требуется представить в отчете. Этот раздел повторяется для каждой записи из базового источника данных отчета.

**1.5. Примечание группы** печатается в конце группы записей и обычно содержит результаты групповых вычислений.

**1.6. Нижний колонтитул** печатается в нижней части каждой страницы отчета и может содержать номера страниц.

**1.7. Примечание отчета** печатается один раз в конце отчета (на последней странице перед нижним колонтитулом) и может содержать результаты вычислений над всеми записями отчета. Этот раздел является самым последним в отчете.

При формировании отчета для печати или предварительного просмотра Access начинает печать с заголовка отчета и верхнего колонтитула. Затем до конца страницы печатается область данных, после которой Access помещает нижний колонтитул, разделитель страниц и верхний колонтитул следующей страницы. Каждая страница содержит верхний колонтитул, столько областей данных, сколько помещается на странице, и нижний колонтитул. На последнюю страницу отчета, перед нижним колонтитулом, помещается область примечаний отчета.

### *Элементы управления, используемые в отчетах*

Элементы управления определяют, как будет выглядеть отчет.

- **Подпись** добавляет текстовые данные, которые не меняются при переходе от одной записи к другой. Подписи можно использовать для добавления фиксированной информации (например, название компании, отдела) или как пояснение к отображаемым данным.

- **Поле** используется для ввода или отображения текста, который меняется при переходе от одной записи к другой. Этот тип элементов управления обычно используется для отображения и ввода данных.

### *Вычисляемые поля в отчетах*

Вычисляемые поля используются в отчетах для объединения данных, либо для проведения дополнительных расчетов по отчету.

1. В вычисляемых элементах управления перед каждым выражением необходимо ставить знак равенства (=).

2. Имя вычисляемого элемента управления, заданное в свойстве *Имя (Name)*, должно быть уникальным. Нельзя использовать имя одного из элементов управления, использованных в выражении.

3. Если в выражении используются арифметические операторы (+, -, \*, /), и один из элементов управления выражения имеет значение *Null*, результатом всего выражения будет значение *Null*

4. Если некоторые записи в одном из элементов управления, используемых в выражении, могут иметь значения *Null*, эти значения следует преобразовать в нулевые с помощью функции *Nz*; например:

=Nz([Стипендия])+Nz([Социальная выплата])

#### 4. Группировка записей в отчете

Группировка записей в отчетах используется для подведения промежуточных и окончательных итогов по отчету.

В отчетах допускается группировка по 10 и менее полям или выражениям.

##### Технология создания группы

1. Откройте отчет в режиме конструктора;
2. Нажмите кнопку **Группировка** вкладки **Инструментов конструктора отчетов**, чтобы открыть соответствующее окно настройки параметров *рис 38* и *рис. 39*;

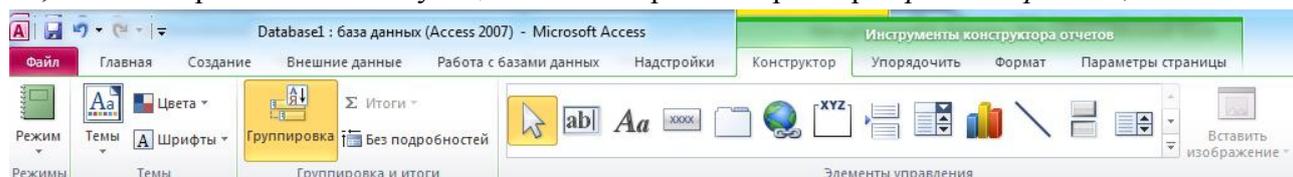


Рис. 38. Команда Группировка на ленте

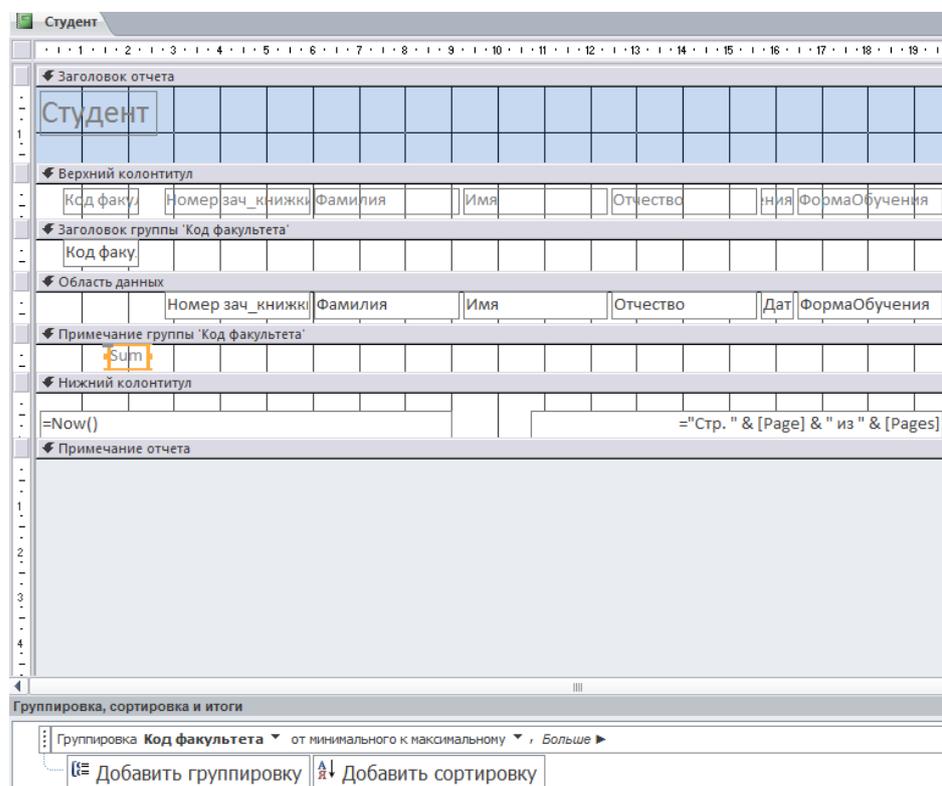


Рис. 39. Добавление группировки в отчет

3. Укажите порядок сортировки данных в отчете;
4. Выберите поле или выражение, для которого задаются параметры группировки;
5. Задайте значения свойств группы, перечисленных в следующем списке. Чтобы создать уровень группировки и определить другие свойства группы, необходимо задать значение *Да* по крайней мере для одного из свойств *Заголовок группы* или *Примечание группы*.

– *заголовок группы* – добавляет или удаляет заголовок группы, определяемой полем или выражением;

– *примечание группы* – добавляет или удаляет область примечаний группы, определяемой полем или выражением;

– *группировка* – определяет способ группировки значений. Список доступных значений этого свойства зависит от типа данных поля, по которому проводится группировка. При группировке по выражению выводится полный список значений данного свойства;

– *интервал* – определяет любой допустимый интервал значений поля или выражения, по которому проводится группировка;

– *не разрывать* – задает или отменяет обязательную печать группы на одной странице.

#### **Подсчет количества записей в каждой группе отчета**

1. Откройте отчет в режиме конструктора;
2. Добавьте в область данных вычисляемое поле;
3. Чтобы открыть окно свойств, убедитесь, что нужное поле выбрано, и нажмите кнопку *Свойства* на панели инструментов;
4. . Задайте указанные значения для следующих свойств;

Свойство	Значение
Имя	ЧислоЗаписей
Данные	=1
Сумма с накоплением	Для группы

5. Добавьте вычисляемое поле в примечание группы;
6. Чтобы открыть окно свойств, убедитесь, что нужное поле выбрано, и нажмите кнопку *Свойства* на панели инструментов.
7. Задайте в качестве значения свойства *Данные* имя элемента управления из области данных, в котором рассчитывается сумма с накоплением, например: =[ЧислоЗаписей].

#### **Подсчет количества записей во всем отчете**

1. Откройте отчет в режиме конструктора.
2. Добавьте вычисляемое поле в примечание отчета.
3. Чтобы открыть окно свойств, убедитесь, что нужное поле выбрано, и нажмите кнопку *Свойства* на панели инструментов.
4. Введите в ячейку свойства *Данные* вычисляемого поля выражение =Count(\*). Это выражение использует функцию *Count* для подсчета всех записей отчета включая записи, имеющие пустые значения в некоторых полях.

## ЗАДАНИЯ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

1. Создайте простой отчет на основе таблицы «Факультеты»;
2. Создайте составной отчет с помощью мастера на основе таблиц «Студенты» и «Факультеты»;
3. Создайте отчет с помощью мастера на основе таблиц «Студенты» и «Факультеты», в отчете должны быть представлены следующие поля – ФИО студента, номер группы, сумма стипендии, наличие детей, наименование факультета, причем поле ФИО – вычисляемое поле отчета (а не запроса);
  - 3.1. Сгруппируйте записи по наименованию факультета,
  - 3.2. Вычислите для каждого факультета количество студентов и общую сумму стипендии (создайте вычисляемые поля в примечании группы «Наименование факультета»);
  - 3.3. Увеличьте размер стипендии для студентов, имеющих детей на 30%, подсчитайте по факультетам увеличенную стипендию;
4. Подведите общие итоги по всему отчету – количество студентов, сумму стипендии, сумму увеличенной стипендии (создайте вычисляемые поля в примечании отчета);
5. Отформатируйте отчет и подготовьте его к сдаче.

**УЧЕБНИКИ, УЧЕБНЫЕ ПОСОБИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ С  
ГРИФОМ МИНОБРАЗОВАНИЯ РФ**

№ п.п.	Наименование	Автор	Год издания	Издательство	Объем (п.л. или а.л.)
1.	Введение в систему баз данных	Дейт, К.Дж	2006	М: Издательский дом «Вильямс»	1328 с.
2	Разработка базы данных	С.В. Лебедева	2011	ИПЦ ФГБОУВПО «СПГУТД»	38 с.