**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ,**
**МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

МОО ВО Кыргызско-Российский Славянский университет
имени первого Президента Российской Федерации Б.Н. Ельцина

УТВЕРЖДАЮ

**Директор Социо-Гуманитарного Института**
**Голобородько И. Л.**

2026 г.

# Базы данных

## рабочая программа дисциплины (модуля)

Закреплена за школой **Международная бизнес-школа**

Учебный план Направление 38.03.01 - РФ, 580100 - КР Экономика
Профиль "Математические методы в экономике"

Квалификация **бакалавр**

Форма обучения **очная**

Общая трудоемкость **4 ЗЕТ**

Часов по учебному плану 144

в том числе:

аудиторные занятия 64

самостоятельная работа 79,8

Виды контроля в семестрах:

зачет с оценкой 6

**Распределение часов дисциплины по семестрам**

| Семестр (<Курс>.<Семестр на курсе>) | 6 (3.2) | | Итого | |
|---|---|---|---|---|
| Недель | 18 | | | |
| Вид занятий | УП | РП | УП | РП |
| Лекции | 32 | 32 | 32 | 32 |
| Практические | 32 | 32 | 32 | 32 |
| Контактная работа в период теоретического обучения | 0,2 | 0,2 | 0,2 | 0,2 |
| В том числе инт. | 8 | 8 | 8 | 8 |
| В том числе в форме практ.подготовки | 32 | 32 | 32 | 32 |
| Итого ауд. | 64 | 64 | 64 | 64 |
| Контактная работа | 64,2 | 64,2 | 64,2 | 64,2 |
| Сам. работа | 79,8 | 79,8 | 79,8 | 79,8 |
| Итого | 144 | 144 | 144 | 144 |

Программу составил(и):
*Доцент, к.т.н., Мусакулова Жылдыз Абдыманаповна* _____

Согласовано РОП:
*Мокроусов Николай Васильевич* _____

Рабочая программа дисциплины
**Базы данных**

разработана в соответствии с ФГОС 3++:
Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 38.03.01 Экономика (приказ Минобрнауки России от 12.08.2020 г. № 954)

составлена на основании учебного плана:

Направление 38.03.01 - РФ, 580100 - КР Экономика
Профиль "Математические методы в экономике"
утвержденного учёным советом вуза от 26.06.2026 протокол № 11.

Рабочая программа одобрена на заседании Международной бизнес-школы

Протокол от 03.09.2026 г. № 1
Срок действия программы: 2026-2030 уч.г.
Руководитель подразделения: Кочербаева Айнура Анатольевна

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ
# к рабочей программе дисциплины (модуля)

| № п/п | Основание внесения изменения | Раздел/ пункт | Содержание изменения |
|---|---|---|---|
| 1. | | | |
| 2. | | | |
| 3. | | | |
| 4. | | | |
| 5. | | | |

| 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ | |
|---|---|
| 1.1 | Целью дисциплины является формирование у студентов теоретических знаний и практических навыков проектирования, создания и использования баз данных, а также умения применять реляционные СУБД для решения экономических и аналитических задач с использованием MS Access и PostgreSQL. |

| 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП | |
|---|---|
| Цикл (раздел) ООП: | Б1.В |
| **2.1** | **Требования к предварительной подготовке обучающегося:** |
| 2.1.1 | Изучение дисциплины опирается на знания, навыки и умения, полученные при освоении курсов: Введение в информационные технологии, Прикладное программное обеспечение, Информационные технологии в профессиональной деятельности. |
| **2.2** | **Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:** |
| 2.2.1 | Анализ данных, Нейронные сети, Преддипломная практика. |

| 3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ) | |
|---|---|
| **ПК-1: Способен системно анализировать связи между объектами и процессами в экономике и бизнесе, выявлять проблемы, сравнивать альтернативы с учетом рисков и выбирать наиболее оптимальный вариант решения.** | |
| **Знать:** | |
| Уровень 1 | основные понятия и модели баз данных, используемые для представления экономических и бизнес-процессов; |
| Уровень 2 | принципы проектирования реляционных баз данных и установления связей между таблицами; |
| Уровень 3 | основы языка SQL для извлечения и анализа данных; |
| Уровень 4 | возможности MS Access и PostgreSQL для анализа экономической информации. |
| **Уметь:** | |
| Уровень 1 | моделировать экономические и бизнес-процессы в виде структурированных баз данных; |
| Уровень 2 | выявлять связи между объектами предметной области с использованием реляционной модели; |
| Уровень 3 | формировать SQL-запросы (в том числе с соединениями и агрегатными функциями) для анализа данных; |
| **Владеть:** | |
| Уровень 1 | навыками проектирования и реализации баз данных в MS Access и PostgreSQL; |
| Уровень 2 | инструментами анализа данных с использованием SQL-запросов, представлений и отчётов; |

**В результате освоения дисциплины обучающийся должен**

| **3.1** | **Знать:** |
|---|---|
| 3.1.1 | основные понятия и модели баз данных, используемые для представления экономических и бизнес-процессов; |
| 3.1.2 | принципы проектирования реляционных баз данных и установления связей между таблицами; |
| 3.1.3 | средства обеспечения целостности и согласованности данных; |
| 3.1.4 | основы языка SQL для извлечения и анализа данных; |
| 3.1.5 | возможности MS Access и PostgreSQL для анализа экономической информации. |
| **3.2** | **Уметь:** |
| 3.2.1 | моделировать экономические и бизнес-процессы в виде структурированных баз данных; |
| 3.2.2 | выявлять связи между объектами предметной области с использованием реляционной модели; |
| 3.2.3 | формировать SQL-запросы (в том числе с соединениями и агрегатными функциями) для анализа данных; |
| 3.2.4 | сравнивать альтернативные управленческие решения на основе результатов запросов к базе данных. |
| **3.3** | **Владеть:** |
| 3.3.1 | навыками проектирования и реализации баз данных в MS Access и PostgreSQL; |
| 3.3.2 | инструментами анализа данных с использованием SQL-запросов, представлений и отчётов; |
| 3.3.3 | приёмами системного анализа экономических данных, хранимых в базах данных. |

**4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**

| Код занятия | Наименование разделов и тем /вид занятия/ | Семестр / Курс | Часов | Компетен-ции | Литература | Инте ракт. | Пр. подг. | Примечание |
|---|---|---|---|---|---|---|---|---|
| | **Раздел 1. Объекты БД. СУБД MS Access** | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1.1 | Проектирование базы данных. /Лек/ | 6 | 2 | ПК-1 | Л1.2 Э1 | | | |
| 1.2 | Создание таблиц. Связи. /Пр/ | 6 | 2 | ПК-1 | Л3.1 Э1 | | 2 | |
| 1.3 | Функциональны зависимости. Связывание таблиц. Виды связей. Индексирование. Каскадирование /Ср/ | 6 | 4 | ПК-1 | Л1.2 Э1 | | | |
| 1.4 | Отбор данных с помощью запросов /Лек/ | 6 | 2 | ПК-1 | Л1.2Л2.1 Э1 | | | |
| 1.5 | Отбор данных с помощью запросов. /Пр/ | 6 | 2 | ПК-1 | Л1.1Л2.2Л3.1 Э1 | | 2 | |
| 1.6 | Разработка запросов в БД (Тема выбирается в зависимости от варианта) /Ср/ | 6 | 4 | ПК-1 | Л1.1Л2.1Л3.1 Э1 | | | |
| 1.7 | Запросы действия /Лек/ | 6 | 2 | ПК-1 | Л1.2Л2.2Л3.1 Э1 Э4 | | | |
| 1.8 | Запросы с вычисляемыми полями /Пр/ | 6 | 2 | ПК-1 | Л2.1Л3.1 Э1 | | 2 | |
| 1.9 | Запросы SQL /Ср/ | 6 | 4 | ПК-1 | Л1.1Л2.1 Э1 | | | |
| 1.10 | Формы(Интерфейс пользователя) /Лек/ | 6 | 2 | ПК-1 | Л1.2Л3.1 Э1 | 1 | | Лекция в форме конференции |
| 1.11 | Использование форм в базе данных. /Пр/ | 6 | 2 | ПК-1 | Л2.2Л3.1 Э1 | | 2 | |
| 1.12 | Свойства формы и ее разделов. /Ср/ | 6 | 4 | ПК-1 | Л1.1Л2.2Л3.1 Э1 | | | |
| 1.13 | Составные формы /Пр/ | 6 | 2 | ПК-1 | Л3.1 Э1 | | 2 | |
| 1.14 | Многотабличные формы. Рекомендации по оформлению. /Ср/ | 6 | 4 | ПК-1 | Л1.1Л2.1 Э1 Э4 | | | |
| 1.15 | Контрольная работа /ЗачётСОц/ | 6 | 10 | ПК-1 | Э1 | | | |
| | **Раздел 2. Создание пользовательского интерфейса** | | | | | | | |
| 2.1 | Отчеты. Печать данных /Лек/ | 6 | 2 | ПК-1 | Л1.2Л3.1 Э1 | 2 | | Лекция в форме конференции |
| 2.2 | Создание отчетов. /Пр/ | 6 | 2 | ПК-1 | Л2.1Л3.1 Э1 | | 2 | |
| 2.3 | Разделы отчета. Использование вычисляемых значений. /Ср/ | 6 | 4 | ПК-1 | Л1.1Л2.2 Э1 | | | |
| 2.4 | Кнопочные формы /Лек/ | 6 | 2 | ПК-1 | Л1.2 Э1 Э4 | 0,5 | | Лекция в форме конференции |
| 2.5 | Разработка кнопочных форм. /Пр/ | 6 | 2 | ПК-1 | Л1.2Л2.2Л3.1 Э1 | | 2 | |
| 2.6 | Разработка навинационной формы приложения /Ср/ | 6 | 4 | ПК-1 | Л1.1 Э1 Э4 | | | |
| 2.7 | Макросы.Модули. /Лек/ | 6 | 2 | ПК-1 | Л1.1 Л1.2Л2.1 Л2.2Л3.1 Э1 | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2.8 | Программирование в Access. /Пр/ | 6 | 2 | ПК-1 | Л2.2Л3.1 Э1 | | 2 | |
| 2.9 | Использование макросов для автоматизации приложения /Ср/ | 6 | 4 | ПК-1 | Л1.1Л2.1 Л2.2Л3.1 Э1 | | | |
| 2.10 | Концептуальное и логическое проектирования базы данных. /Лек/ | 6 | 2 | ПК-1 | Л2.2 Э1 | 0,5 | | Лекция в форме конференции |
| 2.11 | Экспорт и импорт данных в Access /Пр/ | 6 | 2 | ПК-1 | Л2.1 Л2.2 Э1 Э4 | | | |
| 2.12 | Програмирование на VBA в приложении Access /Ср/ | 6 | 4 | ПК-1 | Л1.1Л2.1 Л2.2 Э1 | | | |
| 2.13 | Защита данных /Лек/ | 6 | 2 | ПК-1 | Л1.2Л2.1 Э1 | | | |
| 2.14 | Защита проекта /КрЭк/ | 6 | 0,3 | ПК-1 | Э1 | | | |
| 2.15 | Контрольная работа /ЗачётСОц/ | 6 | 10 | ПК-1 | Э1 | | | |
| | **Раздел 3. Язык SQL и работа с PostgreSQL** | | | | | | | |
| 3.1 | Основы языка SQL. /Лек/ | 6 | 2 | ПК-1 | Л1.3 Л1.4 Л1.5Л2.3Л3.2 Э2 | | | |
| 3.2 | Запросы SELECT: фильтрация, сортировка, агрегатные функции. /Лек/ | 6 | 2 | ПК-1 | Л1.4Л2.3 Э3 | | | |
| 3.3 | Соединения таблиц (JOIN). /Лек/ | 6 | 2 | ПК-1 | Л1.3Л2.3Л3.2 Э2 | 2 | | |
| 3.4 | Установка и настройка PostgreSQL. /Пр/ | 6 | 2 | ПК-1 | Л1.5Л2.3Л3.2 Э3 | | 2 | |
| 3.5 | Создание базы данных и таблиц. /Пр/ | 6 | 2 | ПК-1 | Л1.5Л2.3Л3.2 Э2 | | 2 | |
| 3.6 | Выполнение SQL-запросов различной сложности. /Пр/ | 6 | 2 | ПК-1 | Л1.3 Л1.5Л2.3Л3.2 Э2 Э3 | | 2 | |
| 3.7 | Использование агрегатных функций для анализа данных. /Пр/ | 6 | 2 | ПК-1 | Л1.3 Л1.5 Л1.4Л2.3 Э3 | | 2 | |
| 3.8 | Изучение синтаксиса SQL (DDL и DML) по учебной и электронной литературе. /Ср/ | 6 | 4 | ПК-1 | Л1.3 Л1.5Л2.3Л3.2 Э2 Э3 | | | |
| 3.9 | Самостоятельная разработка реляционной схемы базы данных для экономической задачи. /Ср/ | 6 | 4 | ПК-1 | Л1.3 Л1.5 Л1.4Л2.3Л3.2 Э2 Э3 | | | |
| 3.10 | Подготовка SQL-скриптов для создания таблиц и заполнения их тестовыми данными. /Ср/ | 6 | 4 | ПК-1 | Л1.5Л3.2 Э2 Э3 | | | |
| 3.11 | Выполнение индивидуальных заданий на составление SELECT-запросов с использованием JOIN и агрегатных функций. /Пр/ | 6 | 2 | ПК-1 | Л1.4 Л1.5Л2.3 Э2 Э3 | 2 | 2 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3.12 | Анализ полученных результатов запросов и формулирование выводов для принятия управленческих решений. /Ср/ | 6 | 4 | ПК-1 | Л1.4 Л1.5Л2.3Л3.2 Э2 Э3 | | | |
| 3.13 | Самостоятельная разработка сложных SQL-запросов с использованием подзапросов и агрегатных функций для анализа экономических данных. /Ср/ | 6 | 4 | ПК-1 | Л1.5 Л1.4Л2.3Л3.2 Э3 | | | |
| 3.14 | Контрольная работа /ЗачётСОц/ | 6 | 10 | ПК-1 | Л1.5 Л1.4Л2.3Л3.2 Э2 Э3 | | | |
| | **Раздел 4. Возможности PostgreSQL и аналитические запросы** | | | | | | | |
| 4.1 | Создание представлений и индексов. /Пр/ | 6 | 2 | ПК-1 | Л1.5Л2.3Л3.2 Э2 Э3 | | | |
| 4.2 | Работа с транзакциями. /Пр/ | 6 | 2 | ПК-1 | Л1.5Л2.3Л3.2 Э2 Э3 | | 2 | |
| 4.3 | Ограничения целостности и транзакции. /Ср/ | 6 | 2 | ПК-1 | Л1.5Л2.3Л3.2 Э3 | | | |
| 4.4 | Реализация сложных аналитических запросов. /Ср/ | 6 | 2 | ПК-1 | Л1.5Л2.3Л3.2 Э3 | | | |
| 4.5 | Мини-проект: анализ экономических данных с использованием PostgreSQL. /Ср/ | 6 | 6 | ПК-1 | Л1.5Л2.3 Э2 Э3 | | | |
| 4.6 | Анализ альтернативных управленческих решений на основе результатов аналитических запросов. /Ср/ | 6 | 2 | ПК-1 | Л1.5Л2.3Л3.2 Э1 | | | |
| 4.7 | Контрольная работа /ЗачётСОц/ | 6 | 5,7 | ПК-1 | Л1.5Л2.3Л3.2 Э1 | | | |

## 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### 5.1. Контрольные вопросы и задания

Контрольные вопросы. Приложение 1

### 5.2. Темы курсовых работ (проектов)

Дисциплина не предусматривает написание курсовой работы

### 5.3. Фонд оценочных средств

Задания для лабораторных работ и методические указания. Приложение 2
Исследовательский проект. Приложение 3
Тесты. Приложение 4

### 5.4. Перечень видов оценочных средств

Лабораторные работы
Исследовательский проект
Тесты
Виды работ и шкалы оценок. Приложение 5

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 6.1. Рекомендуемая литература

#### 6.1.1. Основная литература

| | Авторы, составители | Заглавие | Издательство, год |
|---|---|---|---|
| Л1.1 | Бекаревич Ю., Пушкина Н. | Самоучитель Microsoft Access 2013 | БХВ-Петербург 2013 |
| Л1.2 | Мария Волик | Разработка базы данных в Access. Учебное пособие: В учебном пособии представлены теоретические основы разработки баз данных с использованием Ms Access, а также ситуационные задания с описанием последовательности их выполнения и практикоориентированные задания для самостоятельного выполнения | Прометей 2021 |
| Л1.3 | Кузнецов С. Д. | Введение в модель данных SQL: Учебное пособие | Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа 2021 |
| Л1.4 | Полякова Л. Н. | Основы SQL: Учебное пособие | Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа 2020 |
| Л1.5 | Гутман Г. Н. | Объектно-реляционная СУБД PostgreSQL: Учебное пособие | Самара: Самарский государственный технический университет, ЭБС АСВ 2016 |
| **6.1.2. Дополнительная литература** | | | |
| | Авторы, составители | Заглавие | Издательство, год |
| Л2.1 | Гурвиц Г.А. | Microsoft Access 2010. Разработка приложений | БХВ-Петербург 2010 |
| Л2.2 | Букунов С.В., Букунова О.В. | Применение СУБД MS Access для создания бизнес-приложений: Содержит основные сведения о работе в системе управления базами данных Microsoft Access. | СПб. : ЭБС АСВ, 2017 |
| Л2.3 | Крис Фиайли, Хаванов А. В. | SQL | Саратов: Профобразование 2019 |
| **6.1.3. Методические разработки** | | | |
| | Авторы, составители | Заглавие | Издательство, год |
| Л3.1 | Мусакулова Ж.А. | Методические указания к выполнению лабораторных работ по курсу "Базы данных и знаний" | Бишкек : КРСУ 2012 |
| Л3.2 | Фролова Е. А. | Программирование на PL/SQL: Учебно-методическое пособие по дисциплине Базы данных | Москва: Московский технический университет связи и информатики 2016 |
| **6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"** | | | |
| Э1 | Справка и обучение по Access | | https://support.microsoft.com/ru-ru/access |
| Э2 | Официальная документация PostgreSQL | | https://www.postgresql.org/docs/ |
| Э3 | Работа с PostgreSQL | | https://proglib.io/p/learn-postgresql |
| Э4 | Создание баз данных в Microsoft Access | | https://izd-mn.com/PDF/35MNNPU19. |
| **6.3. Перечень информационных и образовательных технологий** | | | |
| **6.3.1 Компетентностно-ориентированные образовательные технологии** | | | |
| 6.3.1.1 | Традиционные: Лекции, лабораторные работы. | | |
| 6.3.1.2 | Инновационные образовательные технологии - дискуссионный разбор выполненных заданий на практических занятиях. | | |
| **6.3.2 Перечень информационных справочных систем и программного обеспечения** | | | |
| 6.3.2.1 | Microsoft Office(Access, Word, Excel, Power Point). | | |
| 6.3.2.2 | PostgreSQL, pgAdmin | | |
| 6.3.2.3 | Облачное хранилище. | | |
| 6.3.2.4 | Тестирующие системы ЭММ-тест и https://onlinetestpad.com/ru. | | |

| **7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)** | |
|---|---|
| 7.1 | Лекции проводятся в виде компьютерных презентаций с использованием мультимедийных средств. |

| | |
|---|---|
| 7.2 | Лабораторные занятия проводятся в компьютерном классе, оснащенном персональными компьютерами с необходимыми параметрами и с установленным профессиональным программным обеспечением. |

| 8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ) |
|---|
| Технологическая карта, виды работ и их вес в оценке дисциплины. Приложение 6<br>Методические указания для выполнения лабораторных работ. Приложение 2 |

**Приложение 1**

## Контрольные вопросы
## по курсу Базы данных

1. 1. Назначение и основные компоненты системы базы данных.
2. Современные СУБД.
3. Базы данных и их модели.
4. Язык манипулирования данными для реляционной модели.
5. Таблица. Способы создания таблиц.
6. Первичный и внешний ключ.
7. Индексирование базы данных.
8. Схема данных. Связи между таблицами. Виды связей.
9. Ограничение и каскадирование.
10. Типы данных.
11. Свойства поля.
12. Ограничения на значения для полей таблицы.
13. Использование маски ввода.
14. Поиск и сортировка данных.
15. Мастер создания запросов.
16. Создание запроса в режиме конструктора.
17. Бланк запроса. Условие отбора в запросах.
18. Группировка и сортировка данных. Статистические функции.
19. Построитель выражений.
20. Использование встроенных функций в вычисляемых полях.
21. Запрос на обновление. Запрос на удаление. Запрос на добавление.
22. Параметрический запрос. Окно параметров.
23. Перекрестный запрос. Заголовки строк. Заголовки столбцов.
24. Формы. Назначение форм.
25. Режимы форм.
26. Способы создания форм.
27. Мастер форм.
28. Конструктор форм.
29. Свойства формы.
30. Элементы управления.
31. Простые и сложные формы.
32. Главная и подчиненная форма.
33. Многотабличные формы.
34. Отчеты. Назначение отчетов.
35. Разделы отчета.
36. Оформление отчета и свойства элементов управления.
37. Создание простых отчетов.
38. Группировка и сортировка данных.
39. Составные отчеты.
40. Внедрение объектов.
41. Вычисления в отчетах.
42. Кнопочная форма.
43. Разделы кнопочной формы.
44. Макрос. Назначения макроса.
45. Основные макрокоманды.
46. Назначение и структура языка SQL.

47. Команды DDL и их назначение.
48. Команды DML и их назначение.
49. Синтаксис оператора SELECT.
50. Условия фильтрации данных (WHERE).
51. Сортировка и ограничение выборки.
52. Агрегатные функции и группировка данных.
53. Назначение соединений таблиц (JOIN).
54. Отличия PostgreSQL от других СУБД.
55. Что такое представления (VIEW) и для чего они используются?
56. Назначение индексов и их влияние на производительность.
57. Понятие транзакции и свойства ACID.
58. Команды управления транзакциями.
59. Подзапросы и их виды.
60. Использование SQL для анализа экономических данных.

**Приложение 2**

**Лабораторные работы и методические указания к лабораторным работам по курсу Базы данных**

## Лабораторная работа №1. Создание таблиц. Связи.

Таблица базы данных — это обычная таблица из строк и столбцов. Строки таблицы называются записями. Каждая запись описывает один объект, запись содержит несколько ячеек таблицы, хранящих определенные сведения об объектах. Ячейки таблицы, составляющие запись, по терминологии баз данных называются полями. В ячейках хранятся атрибуты объекта. Связь между двумя таблицами организуется посредством общих полей этих таблиц. Если одна таблица ссылается на другую, то первая таблица называется дочерней (как таблица Студенты), а вторая — родительской (как таблица Группы). Если одна запись ссылается на другую, они также называются дочерней и родительской.

В реляционной базе данных существует два вида ключевых полей: первичный ключ и вторичный ключ. Для идентификации записей в таблице используется первичный ключ - поле или набор полей таблицы, чьи значения однозначно определяют запись. Соответственно значения первичного ключа должны быть уникальны, то есть в таблице не должно быть двух строк с одинаковыми значениями первичного ключа. Вторичным ключом называется поле дочерней таблицы, содержащее значения первичного ключа родительской таблицы. Связи между таблицами организуются с помощью ключевых полей.

Для создания и редактирования таблиц предназначены два режима.

- Режим таблицы предназначен для ввода, просмотра и редактирования записей. При вводе данных программа Access автоматически добавляет недостающие поля и определяет тип данных для этих полей.
- Режим конструктора таблицы предназначен для настройки свойств полей, таких как тип данных, значение по умолчанию, список возможных значений, маска значений и многих других.

**Основные типы данных.**

- Текстовый. Поля этого типа содержат текст, объем которого не должен превышать 255 символов. Длина поля устанавливается посредством параметра «Размер поля», где необходимо указать максимальное количество символов, которые можно ввести в это поле.
- Поле MEMO. В MEMO-полях хранятся текстовые либо текстовые и числовые данные объемом не более 65535 символов.
- Числовой. Поля этого типа содержат числовые значения, диапазон которых определяется параметром размер поля.
- Дата время. Поля данного типа содержат значения даты и времени и допускают ввод дат от 100 года по 9999.
- Денежный. В полях этого типа хранятся числа, содержащие 15 разрядов слева от десятичной точки и 4 разряда справа от нее. Такие поля следует использовать при выполнении денежных расчетов, поскольку в процессе вычислений в этом случае можно избежать округления.

- Счетчик. Поле этого типа содержит уникальное значение, которое автоматически увеличивается на единицу, когда в таблицу добавляется новая запись. Значение поля типа счетчик нельзя обновлять.
- Логический. В таких полях сохраняются значения Да и Нет (Истина и Ложь, Вкл и Выкл). Access использует величину -1 для представления значения ДА, и величину 0 для представления значения Нет.
- Поле объекта OLE. В полях данного типа помещаются объекты, обработанные OLE-сервером. В них можно хранить произвольные данные. OLE – это способ обмена и совместного использование данных.
- Гиперссылка. В полях этого типа хранятся гиперссылки, представляющие собой путь к файлу на жестком диске, либо адрес в сети Интернет.

Необходимо заметить, что изменение типа данных поля после ввода данных в таблицу иногда сопровождается очень длительной процедурой их преобразования при сохранении таблицы. Если Access не сможет выполнить преобразование без ошибок, то часть данных будет утеряна или искажена.

**Свойства полей.**

- Размер поля. Определяет максимальную длину текстового или числового поля (при создании полей слишком большого размера неэкономно расходуется память, а полей слишком маленького размера – искажается содержимое поля).
- Формат поля. Устанавливает формат отображения данных из поля в режиме таблицы.
- Число десятичных знаков. Определяет количество разрядов в дробной части десятичного числа.
- Маска ввода. Определяет маску данных при вводе. Эта маска позволяет автоматизировать проверку ввода символов в поле и применяется к полям, содержащим дату, номер телефона и т.д.
- Подпись. Содержит надпись, которая выводится рядом с полем в форме или отчете.
- Значение по умолчанию. Содержит значение, устанавливаемое в соответствующем поле таблицы, по умолчанию.
- Условие на значение. Определяет множество значений, которые можно вводить в это поле при заполнении таблицы. В ситуации несоответствия вводимых данных указанному условию выдается сообщение об ошибке.
- Индексированное поле. Определяет простые индексы для ускорения поиска (поле первичного ключа индексируется автоматически). Существует два режима индексирования: режим, при котором совпадения допускаются, и режим, когда совпадения не допускаются. В первом случае поле может содержать повторяющиеся значения, во втором – нет. Также вариант «нет» означает, что поле не является индексированным.
- Сообщение об ошибке. Определяет сообщение, которое появляется на экране при вводе недопустимого значения.
- Обязательное поле. Указывает на то, что данное поле обязательно следует заполнить.
- Пустые строки. Определяет, допускается ли ввод в данное поле пустых строк « ».
- Сжатие Юникод. Указывает, используется ли кодировка Unicode. Определяет, выполняется ли сжатие данных при сохранении для полей текстового типа.
- Режим IME. Input Method Edition. Позволяет распознавать вводимые с помощью специальных кодов символы азиатских алфавитов. Это свойство дает возможность указывать режим преобразования, которое выполняется при переходе к этому полю.

- Режим предложений IME. Определяет режим предложений IME, который применяется при переходе к этому полю.
- Смарт-теги. Предоставляет возможность присоединения смарт-тегов.
- Новые значения. Используется для полей типа «Счетчик» и задает способ формирования следующего значения счетчика. Здесь возможен выбор одного из двух вариантов: последовательные и случайные.

**Создание таблицы.**

Для создания новой таблицы в режиме конструктора необходимо выполнить команду «Конструктор таблиц» (Рис. 1).

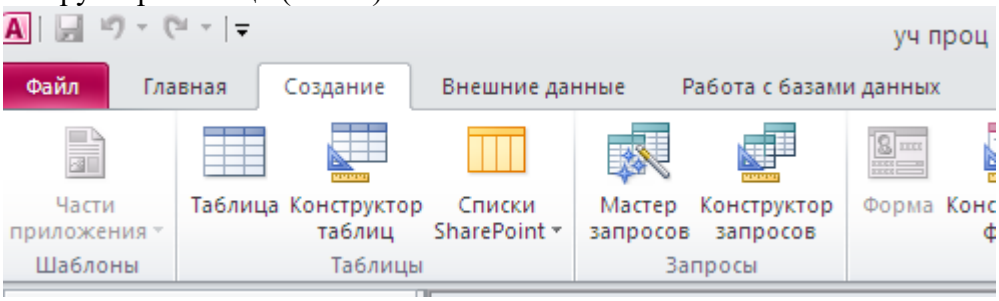



Рис. 1. Лента. Вкладка «Создание»

В открывшемся окне конструктора таблиц можно вводить имена полей новой таблицы. Преимуществом создания полей в конструкторе является возможность задать свойства полей, ускоряющие ввод данных (например, список значений поля, значение по умолчанию) и уменьшающие количество случайных ошибок ввода (маска ввода, условия на значение и др.).

В верхней части конструктора перечислены поля таблицы. Для добавления нового поля необходимо установить курсор в первую пустую запись, ввести название поля и нажать клавишу Enter. Курсор перейдет в следующую ячейку, и в правой части ячейки появится кнопка. С ее помощью выбирается из списка (рис. 2) тип данных поля.

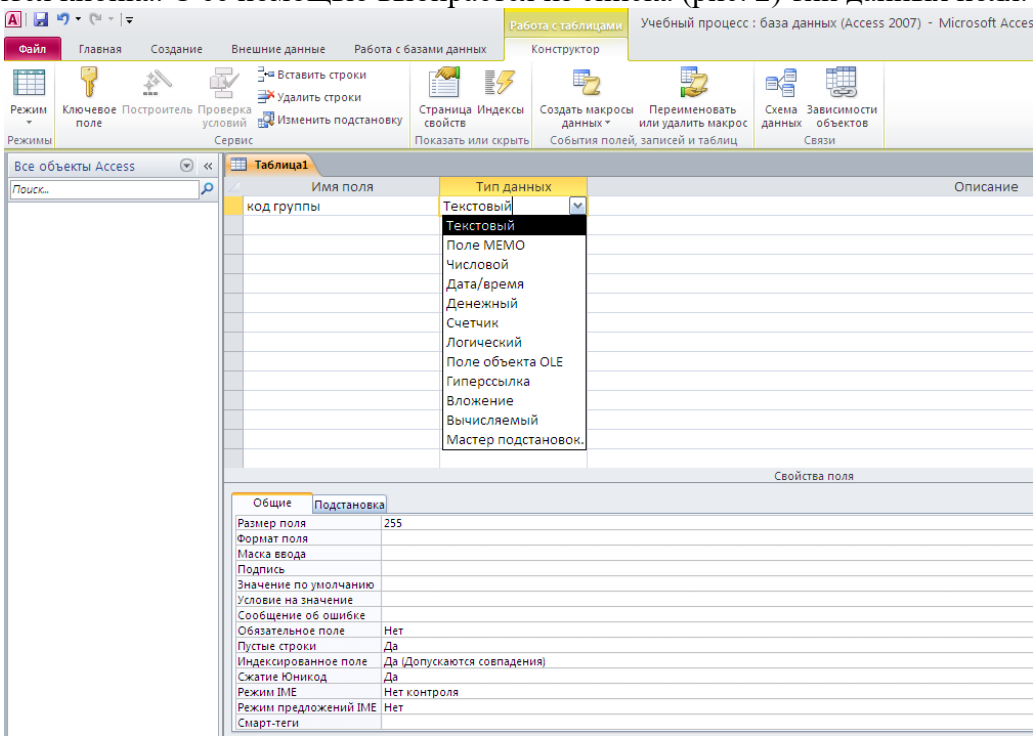



Рис. 2. Выбор типа данных поля.

**Создание ключей.**

Для создания первичного ключа необходимо выбрать требуемое поле, выделить его, щелкнув область выделения строки нужного поля. Затем, необходимо выполнить команду «Ключевое поле» на вкладке «Конструктор». (Рис. 3).

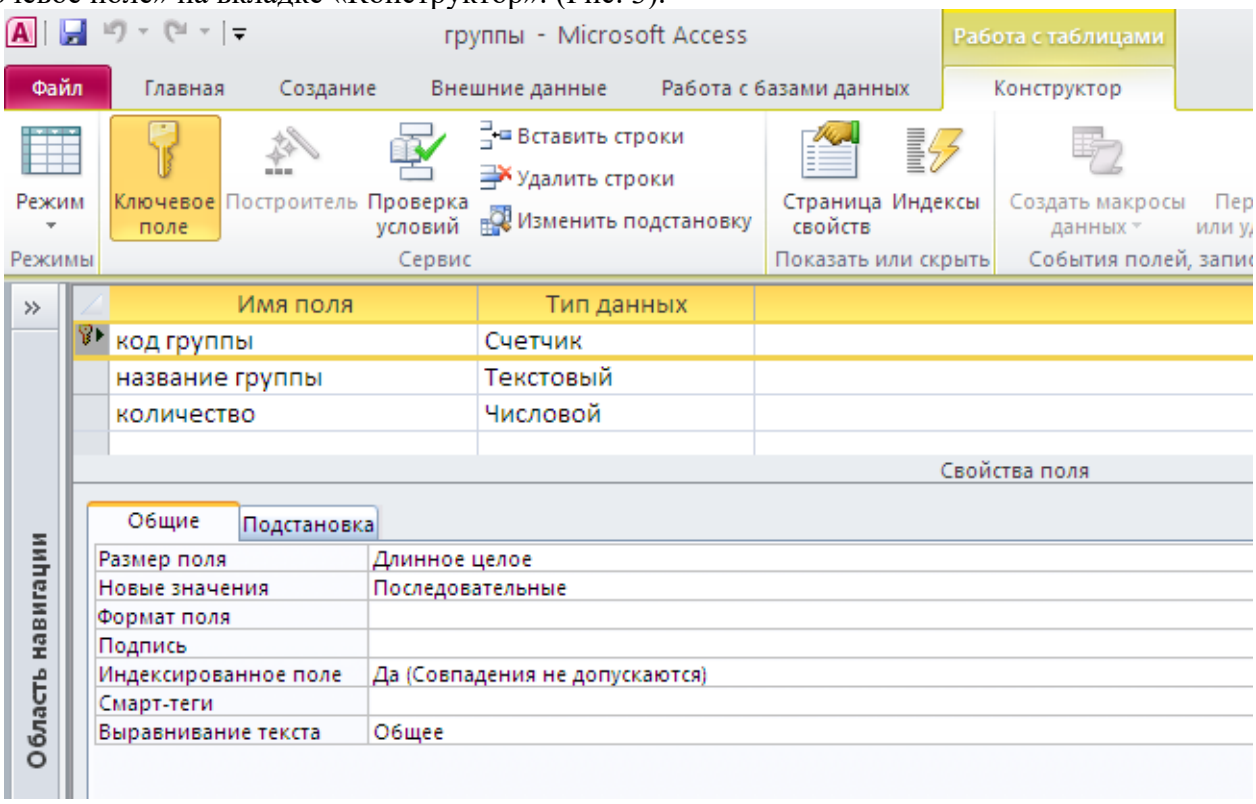



Рис. 3. Установка первичного ключа.

Необходимо обратить внимание, что в свойстве «Индексированное поле» для поля первичного ключа всегда устанавливается значение «Да (Совпадения не допускаются)».

При создании внешнего ключа для поля в свойстве «Индексированное поле» устанавливается значение «Да (Совпадения допускаются)». Важно помнить, что типы данных первичного и внешнего ключа обязательно должны совпадать.

После определения структуры таблицы необходимо ее сохранить.

**Создание связей между таблицами.**

Следует отметить, что для установки связи между таблицами используются общие поля этих таблиц. Вторичный ключ дочерней таблицы содержит значения первичного ключа родительской таблицы, и таким образом запись дочерней таблицы ссылается на одну из записей родительской таблицы. Просмотр, создание, редактирование и удаление связей осуществляются на схеме данных (рис. 4). Чтобы открыть схему данных, необходимо вызвать команду «Схема данных» на вкладке «Работа с базами данных».

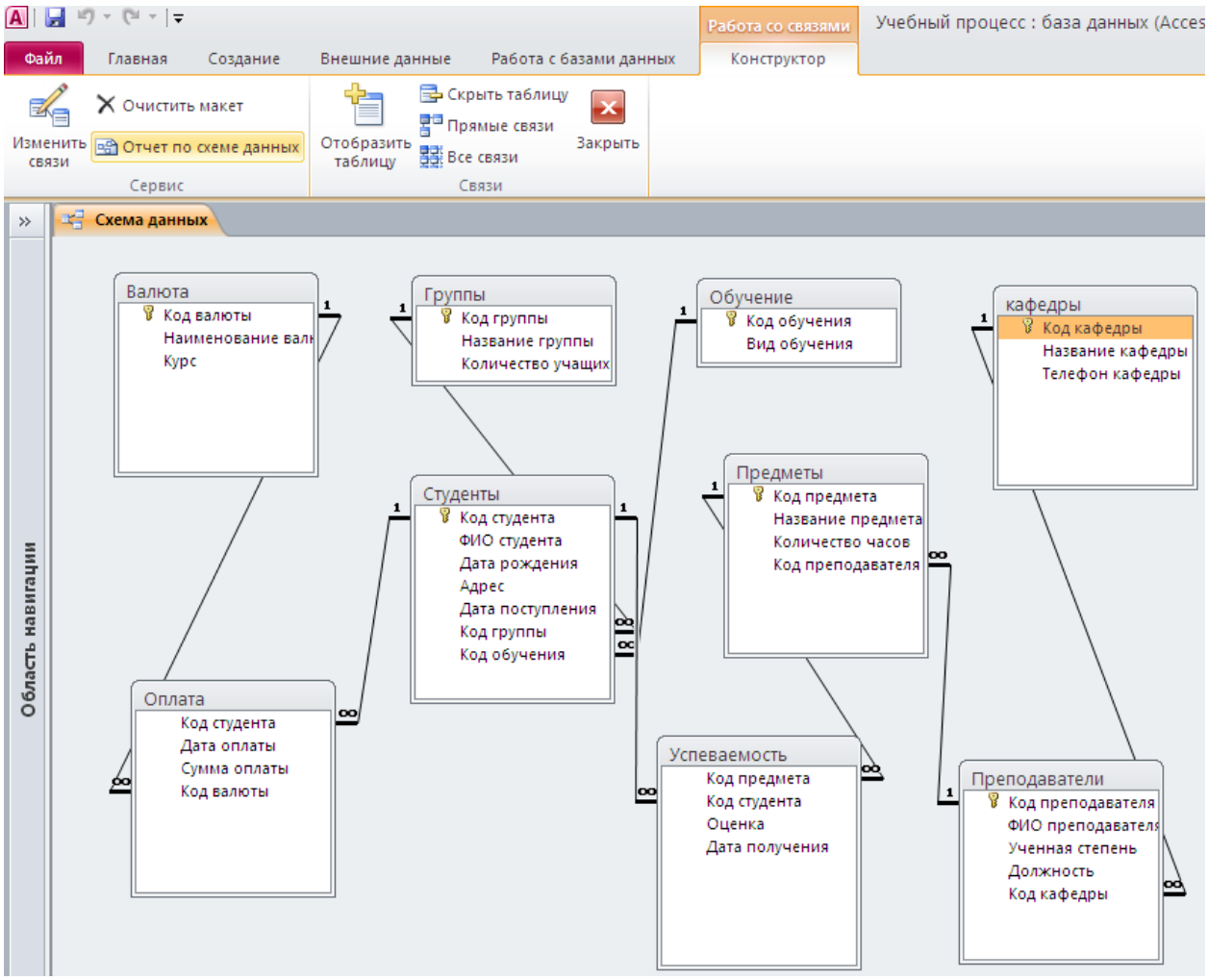



Рис. 4. Схема данных.

Для создания связи между таблицами необходимо щелкнуть кнопкой мыши на названии поля, которое является первичным ключом в родительской таблице. Удерживая нажатой кнопку мыши, перетащить поле первичного ключа, навести указатель на название того поля дочерней таблицы, которое должно стать вторичным ключом и отпустить кнопку мыши. На экране появится окно «Изменение связей» (рис. 5). В этом окне указаны названия связываемых полей. Необходимо убедиться, что поля для создания связи выбраны, верно. В случае обнаружения ошибки нужно выбрать другие поля.

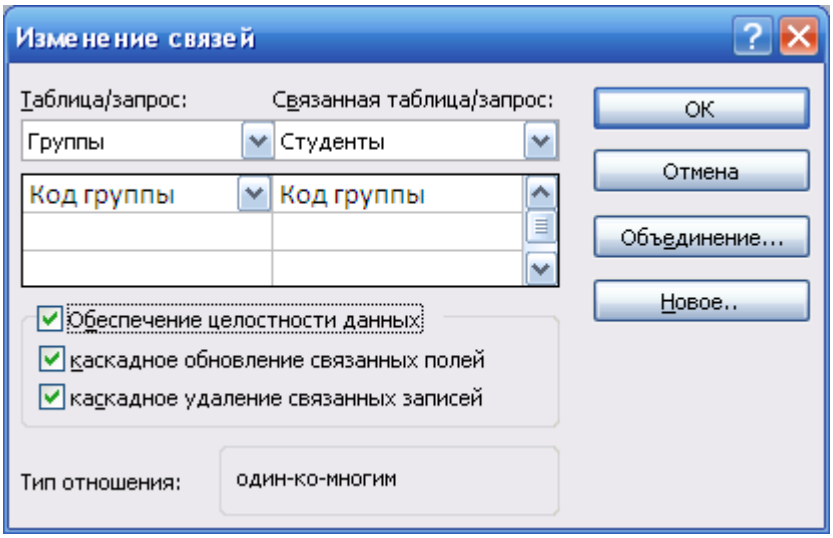



Рис. 5. Изменение связей.

Рекомендуется установить флажок Обеспечение целостности данных. Целостностью данных в Microsoft Access называется ссылочная целостность — состояние отдельной связи или всей базы данных, при котором отсутствуют некорректные ссылки. Иными словами, целостность данных означает, что в дочерней таблице нет записей, ссылающихся «в никуда», вторичный ключ каждой записи содержит значение, существующее в столбце первичного ключа родительской таблицы.

Целостность данных может нарушиться в следующих ситуациях:

- в дочернюю таблицу добавляется новая запись, содержащая некорректное (отсутствующее в первичном ключе родительской таблицы) значение вторичного ключа, или в существующей записи дочерней таблицы значение вторичного ключа изменяется на некорректное;
- из родительской таблицы удаляется запись, в результате чего ссылки на эту запись становятся некорректными;
- в родительской записи изменяется значение первичного ключа, следовательно, ссылки на эту запись также становятся некорректными.

Исключить все эти ситуации позволяет режим автоматического обеспечения целостности данных. Этот режим включается отдельно для каждой связи в окне «Изменение связи» (рис. 5). Если установить флажок «Обеспечение целостности данных», программа Access не допустит выполнения операций, ведущих к нарушению ссылочной целостности. Так, запрещается:

- сохранение записи с некорректным значением вторичного ключа;
- удаление родительской записи, если для связи не установлен флажок каскадное удаление связанных записей; либо вместе с родительской записью автоматически удаляются дочерние записи, если флажок каскадное удаление связанных записей установлен;
- изменение значения первичного ключа родительской записи, если для связи не установлен флажок каскадное обновление связанных полей; либо при изменении значения первичного ключа в родительской записи автоматически меняются значения вторичного ключа в дочерних записях, если флажок каскадное обновление связанных полей установлен.

Если вторичный ключ дочерней таблицы содержит некорректные значения, то включить режим обеспечения целостности невозможно: необходимо вначале удалить или изменить значения, нарушающие целостность данных.

**Контрольные вопросы:**

1. Таблица. Способы создания таблиц.
2. Первичный и внешний ключ.
3. Индексирование базы данных.
4. Схема данных. Связи между таблицами. Виды связей.
5. Ограничение и каскадирование.
6. Типы данных.
7. Свойства поля.
8. Ограничения на значения для полей таблицы.
9. Использование маски ввода.

**Задания к лабораторной работе:**

Разработать базу данных «Учебный процесс» (Файл назвать своей фамилией), состоящую из следующих таблиц и полей:

Группы:

- код группы (счетчик) – первичный ключ

- название группы (текстовый)
- количество учащихся (числовой)

Обучение:
- код обучения (счетчик) – первичный ключ
- вид обучения (текстовый)

Студенты:
- код студента (счетчик) – первичный ключ
- код группы (числовой) – внешний ключ
- код обучения (числовой) – внешний ключ
- ФИО студента (текстовый)
- дата поступления (дата/время)
- адрес (тестовый)
- дата рождения (дата/время)

Кафедры:
- код кафедры (счетчик) – первичный ключ
- название кафедры (текстовый)
- ФИО зав. кафедры (текстовый)
- телефон кафедры (текстовый)

Преподаватели:
- код преподавателя (счетчик) – первичный ключ
- код кафедры (числовой) – внешний ключ
- ФИО преподавателя (текстовый)
- ученая степень (текстовый)

Предметы:
- код предмета (счетчик) – первичный ключ
- код преподавателя (числовой) – внешний ключ
- название предмета (текстовый)
- количество часов (числовой)
- количество лекций (числовой)
- количество практик (числовой)

Успеваемость:
- код записи (счетчик) – первичный ключ
- код студента (числовой) – внешний ключ
- код предмета (числовой) – внешний ключ
- оценка (числовой)
- дата получения оценки (дата/время)

Валюта:
- код валюты (счетчик) – первичный ключ
- наименование валюты (текстовый)

Оплата:
- код платежа (счетчик) – первичный ключ
- код студента (числовой) – внешний ключ
- дата платежа (дата/время)
- сумма платежа (числовой)
- код валюты (числовой) – внешний ключ

**Связать таблицы** (рис. 4)

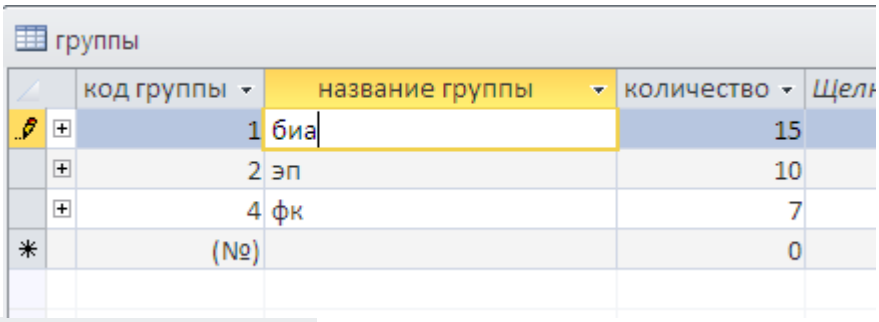
**Заполнить созданную базу информацией в следующем порядке.**

Группы: 3 записи

Например:

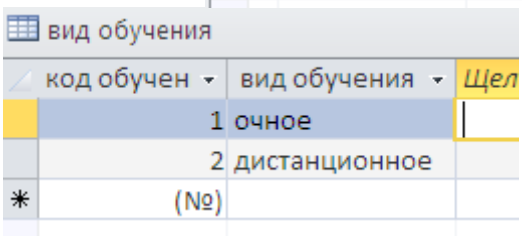
Обучение: 2 записи

Например:



группы

| код группы | название группы | количество | Щелк |
|---|---|---|---|
| 1 | биа | 15 | |
| 2 | эп | 10 | |
| 4 | фк | 7 | |
| (№) | | 0 | |

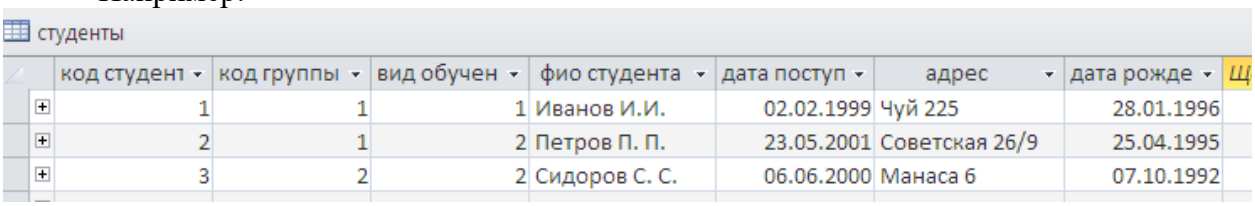


вид обучения

| код обучен | вид обучения | Щел |
|---|---|---|
| 1 | очное | |
| 2 | дистанционное | |
| (№) | | |

Студенты: 15 записей

Например:



студенты

| код студент | код группы | вид обучен | фио студента | дата поступ | адрес | дата рожде | Щ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | Иванов И.И. | 02.02.1999 | Чуй 225 | 28.01.1996 | |
| 2 | 1 | 2 | Петров П. П. | 23.05.2001 | Советская 26/9 | 25.04.1995 | |
| 3 | 2 | 2 | Сидоров С. С. | 06.06.2000 | Манаса 6 | 07.10.1992 | |

Кафедры: 3 записи

Преподаватели: 5 записей

Предметы: 5 записей

Успеваемость: 70 записей

Валюта: 3 записи

Оплата: 10 записей

## Лабораторная работа №2. Отбор данных с помощью запросов.

Запрос — это обращение к данным для получения информации, либо выполнения действий над ними. С помощью запросов можно просматривать, анализировать, редактировать данные из одной или нескольких таблиц, сортировать определенные данные, делать выборку по желаемым условиям, выполнять запрос из таблиц, который может пригодиться для составления форм и отчетов.

В MS Access запросы делятся: на QBE-запросы и SQL-запросы

QBE-запросы (Query By Example – запрос по образцу) создаются с помощью стандартных средств MS Access. Параметры QBE-запроса устанавливаются в режиме конструктора запросов.

При создании SQL-запроса (Structured Query Language – Язык структурированных запросов) применяются операторы и функции языка SQL. Этот язык является стандартизованным. Access легко преобразует QBE-запрос в SQL-запрос и наоборот.

QBE-запросы делятся на следующие типы:

- Запрос на выборку
- Перекрестный запрос
- Запросы действия

В свою очередь запросы действия делятся:

- Запрос на удаление данных
- Запрос на добавление данных
- Запрос на изменение данных
- Запрос на создание новой таблицы

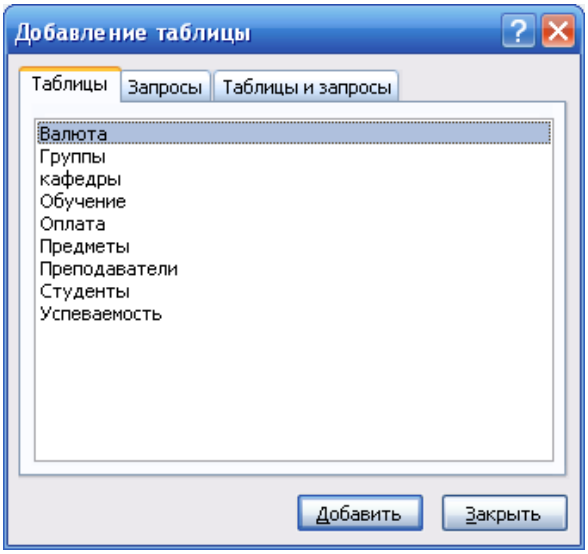
Запрос на выборку может быть с параметрами, с вычисляемыми полями или с группировкой данных.

**Создание запроса на выборку.**

Создание нового запроса с помощью конструктора запросов позволяет получить гораздо больше возможностей. При использовании конструктора запросов можно не только создавать новые запросы, но также изменять уже имеющиеся запросы в базе данных.

Рис. 1. Добавление таблицы.

Для создания запроса с использованием конструктора необходимо перейти на вкладку




«Создание». И нажать кнопку «Конструктор запросов». Откроется окно добавление таблицы (рис. 1). В появившемся окне есть три вкладки: «Таблицы», «Запросы», «Таблицы и запросы». На вкладке «Таблицы» необходимо выбрать нужные таблицы и нажать кнопку «Добавить». Затем нажать кнопку «Закрыть», после чего на экране появится пустой бланк запроса на основании выбранной таблицы (рис. 2).

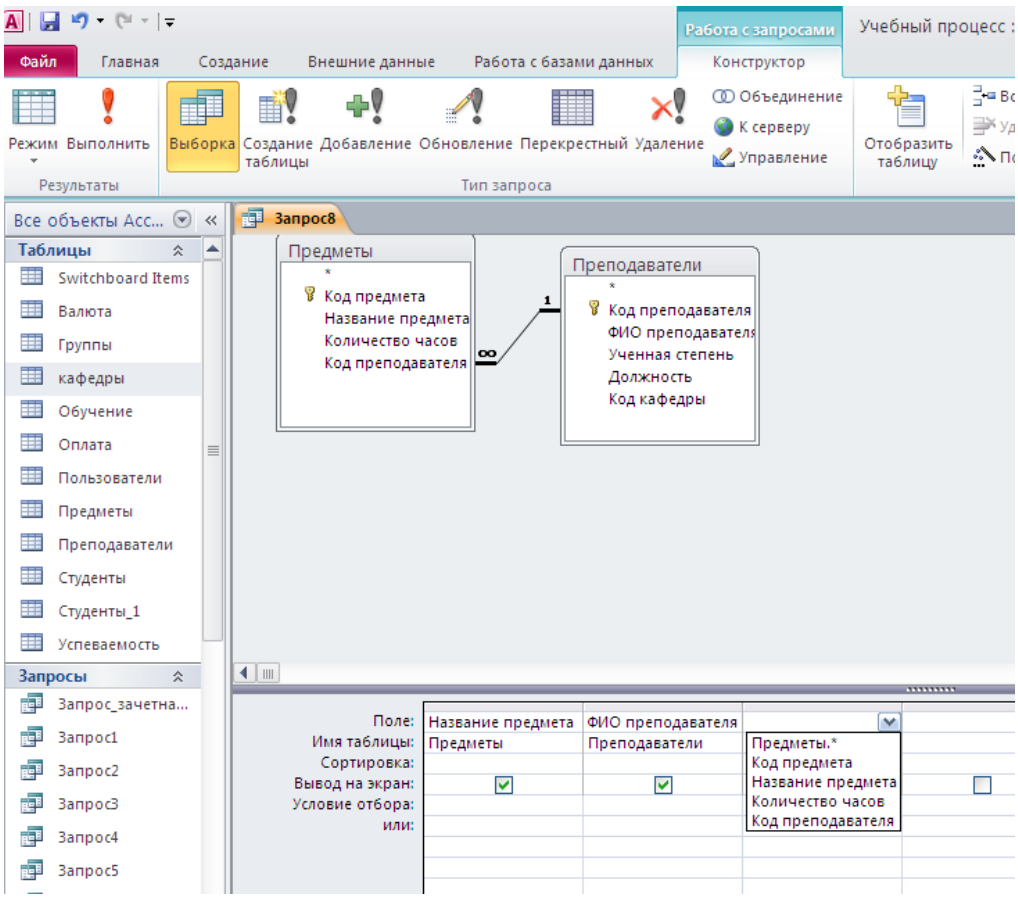



Рис. 2. Бланк запроса.

В нижней части окна в открывающемся списке «Поле» необходимо выбрать нужные поля таблицы. Далее необходимо сохранить запрос. Для просмотра запроса необходимо нажать кнопку «Выполнить» на вкладке «Конструктор». Результат выполнения запроса просматривается в режиме таблице (рис. 3).

Используя бланк запроса в режиме конструктора, модифицировать существующий запрос можно следующими способами.

1. Удалить таблицу из запроса можно, если выделить ее в запросе и нажать клавишу «Delete».
2. Удалить поле таблицы можно, выделив желаемый столбец с помощью области выделения столбцов (серая полоска над столбцами) и затем нажать клавишу «Delete».
3. Изменить порядок следования полей можно следующим образом: необходимо выделить поле или несколько полей и, удерживая нажатой кнопку мыши, перетащить их на нужное место, затем отпустить кнопку мыши.
4. Условия отбора записей из базовой таблицы указываются в строках «Условие отбора» и «Или». Данными условиями могут являться любые логические выражения. Существуют условия, которые находятся в одной строке, но в разных столбцах бланка, такие условия объединяются по логическому оператору AND (И). Чтобы объединить условия по логическому оператору OR (ИЛИ), нужно объединить данные условия в разных строках бланка запроса.
5. Строка «Сортировка» помогает установить порядок сортировки записей в запросе. Для поля, которое необходимо сортировать, нужно выбрать из списка порядок сортировки: «По убыванию» или «По возрастанию». Если в запросе необходимо отсортировать записи по нескольким полям, нужно разместить эти поля слева направо.

6. В бланке запросов можно изменить ширину столбцов. Это можно сделать, перетаскивая границу столбца или два раза щелкнув кнопкой мыши на линии границы.

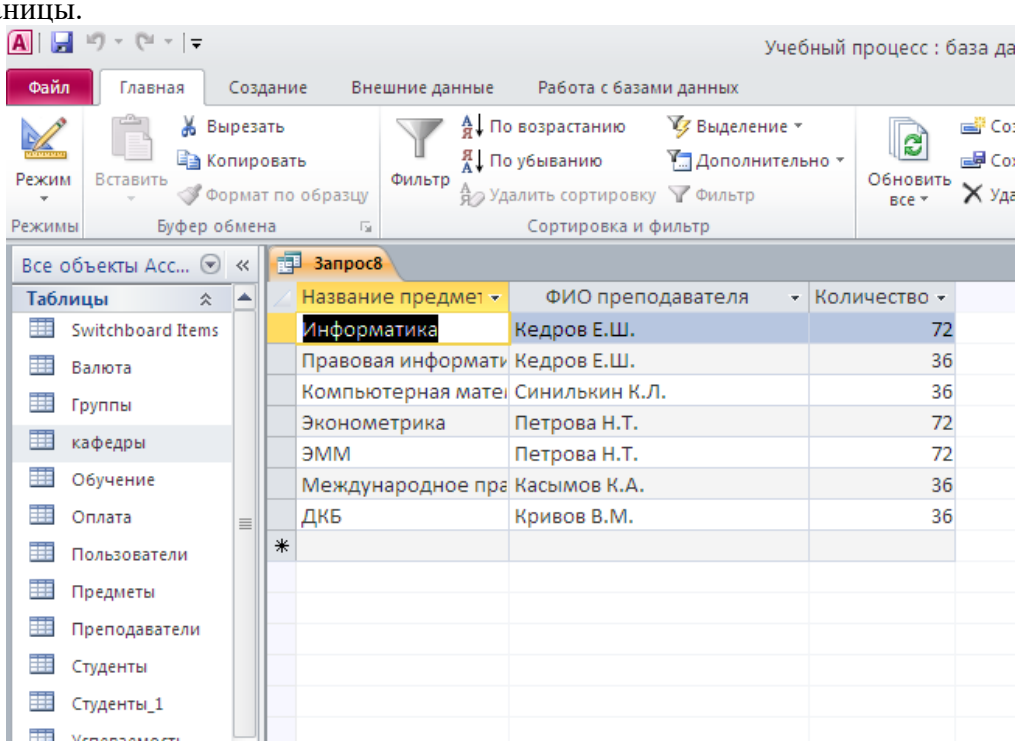



Рис. 3. Результат запроса.

**Условие отбора.**

Добавление в запрос условия отбора позволяет выбирать из таблицы не все записи, а лишь те, которые удовлетворяют определенным критериям. Например, нам нужны записи, приходящиеся на октябрь 2011 года. Создаем новый запрос в режиме конструктора на базе двух таблиц: «Студенты», «Оплата». И вводим условие: Between DateValue("01.10.2010") And DateValue("31.10.2010") в столбец «Дата оплаты», в строку «Условие отбора» (рис. 4.). При создании условий отбора можно пользоваться специальными символами. ? – один неизвестный символ, * – любое число произвольных символов

Например, можно вывести список студентов оплативших в октябре 2011 года, фамилии которых начинаются на букву «А». (Like "А*")

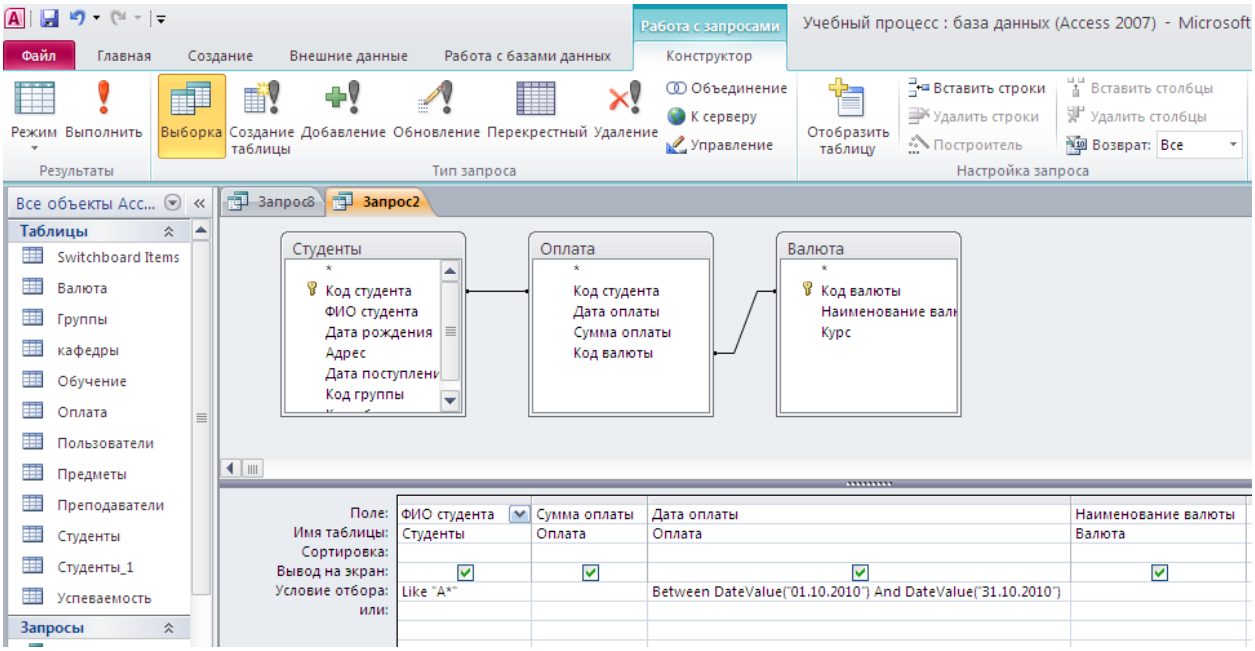



Рис. 4. Условие отбора.

**Примеры условий запроса.**

Таблица 1 . Условия для полей с типом данных «Текстовый».

| **Записи** | **Условие** | **Результат запроса** |
|---|---|---|
| Точно соответствуют определенному значению, например, «Кыргызстан» | "Кыргызстан" | Записи, в которых поле «Страна» содержит значение «Китай». |
| Не соответствуют определенному значению, например, «Мексика» | Not "Мексика" | Записи, в которых значением поля «Страна» не является «Мексика». |
| Начинаются с заданной строки символов, например «С» | Like С* | Записи, в которых название страны начинается с буквы «С», например, Словакия, США и т.д. |
| Не начинаются с заданной строки символов, например «С» | Not Like С* | Записи, в которых названия стран не начинаются с буквы «С». |
| Содержат заданную строку, например «Корея» | Like "*Корея*" | Записи с названиями стран, которые содержат слово «Корея». |
| Не содержат заданную строку, например «Корея» | Not Like "*Корея*" | Записи с названиями стран, которые не содержат слово «Корея». |
| Заканчиваются заданной строкой, например «стан» | Like "*стан" | Записи, в которых названия стран заканчиваются на «стан», например «Казахстан» или «Кыргызстан». |
| Не заканчиваются заданной строкой, например «ина» | Not Like "*ина" | Записи, в которых названия стран не заканчиваются на «ина», например «Украина» или «Аргентина». |
| Содержат пустые значения (значение отсутствует) | Is Null | Записи, в которых данное поле не содержит значения. |

| | | |
|---|---|---|
| Не содержат пустых значений | Is Not Null | Записи, в которых данное поле содержит ненулевое значение. |
| Входят в определенный диапазон, например от А до Г | Like "[А-Г]*" | Записи, в которых название страны начинается с одной из букв от «А» до «Г». |
| Соответствуют заданному шаблону | Like "Лив??" | Записи для стран, таких как Ливия и Ливан, названия которых включают пять знаков, первые три из которых — «Лив». |

Таблица 2. Условия для полей типа «Числовой», «Денежный» и «Счетчик».

| **Записи** | **Условие** | **Результат запроса** |
|---|---|---|
| Точно соответствуют определенному значению, например 100 | 100 | Записи, в которых цена единицы продукта составляет 100 сомов. |
| Не соответствуют значению, например, 1000 | Not 1000 | Записи, в которых цена единицы продукта не равна 1000 сом. |
| Содержат значение, которое меньше заданного, например 100 | < 100<br><= 100 | Записи, в которых указана цена единицы меньше 100 сом (<100). Второе выражение (<=100) позволяет отобразить записи, в которых цена единицы меньше либо равна 100. |
| Содержат значение, которое больше заданного, например 99,99 | >99,99<br>>=99,99 | Записи, в которых указана цена единицы больше 99,99 рублей (<99,99). Второе выражение позволяет отобразить записи, в которых цена единицы больше либо равна 99,99. |
| Содержат одно из двух значений, например 20 или 25 | 20 or 25 | Записи, в которых цена единицы равна 20 или 25 сом. |
| Содержат значение, которое входит в определенный диапазон | >49.99 and <99.99<br>-или-<br>Between 50 and 100 | Записи, в которых указана цена единицы в диапазоне между 49,99 и 99,99 сом (сами эти значения не включаются в результаты). |
| Содержат значение, которое не входит в определенный диапазон | <50 or >100 | Записи, в которых цена единицы не входит в диапазон от 50 до 100 рублей. |
| Содержат пустые (или отсутствующие) значения | Is Null | Записи, в которых поле «Цена» не заполнено. |
| Содержат непустые значения | Is Not Null | Записи, в которых поле «Цена» заполнено. |

Таблица 3. Условия для полей типа «Числовой», «Денежный» и «Счетчик».

| **Записи** | **Условие** | **Результат запроса** |
|---|---|---|

| | | |
|---|---|---|
| Точно соответствуют определенному значению, например 02.02.2006 | #02.02.2006# | Записи об операциях, совершенных 2 февраля 2006 г. Значения даты должны быть окружены знаками #, чтобы Access мог отличить значения даты от текстовых строк. |
| Содержат значения, которые входят в определенный диапазон дат | >#02.02.2006# and <#04.02.2006# | Записи об операциях, совершенных между 2 и 4 февраля 2006 г.<br><br>Для фильтрации по диапазону значений можно также использовать оператор **Between**. Например, выражение Between #02.02.2006# and #04.02.2006# идентично выражению >#02.02.2006# and <#04.02.2006#. |
| Содержат любой день заданного месяца (вне зависимости от года), например, «декабрь» | DatePart("M", [ДатаПродажи]) = 12 | Записи об операциях, совершенных в декабре любого года. |
| Содержат любую дату в пределах заданного, (например, первого) квартала вне зависимости от года | DatePart("q", [ДатаПродажи]) = 1 | Записи об операциях, совершенных в первом квартале любого года. |
| Содержат текущую дату | Date() | Записи об операциях, совершенных на текущую дату. Если текущая дата — 02.02.2006, отображаются записи, в которых значению поля «ДатаЗаказа» соответствует 2 февраля 2006 г. |
| Содержат вчерашнюю дату | Date()-1 | Записи об операциях, совершенных за один день до текущей даты. Если текущая дата — 02.02.2006, отображаются записи с датой 1 февраля 2006 г. |
| Содержат один из последних 7 дней | Between Date() and Date()-6 | Записи об операциях, совершенных в течение последних 7 дней. Если текущая дата — 02.02.2006, отображаются записи за период с 24 января 2006 по 2 февраля 2006 г. |
| Содержат дату, которая приходится на текущий месяц | Year([ДатаПродажи]) = Year(Now()) And Month([ДатаПродажи]) = Month(Now()) | Записи за текущий месяц. Если текущая дата — 02.02.2006, отображаются записи за февраль 2006 г. |
| Содержат дату, которая приходится на текущий год | Year([ДатаПродажи]) = | Записи за текущий год. Если текущая дата — 02.02.2006, отображаются |

| | | |
|---|---|---|
| | Year(Date()) | записи за 2006 год. |
| Содержат дату, которая приходится на следующий год | Year([ДатаПродажи]) = Year(Date()) + 1 | Записи об операциях за следующий год. Если текущая дата — 02.02.2006, отображаются записи за 2007 год. |

**Запросы с параметрами.**

Запрос с параметрами — это запрос, позволяющий выполнять один и тот же запрос, но с разными значениями в условии отбора. Он способствует упрощению схемы действий, так как не надо каждый раз повторять запрос с разными условиями, а достаточно ввести новое значение параметра.

Для создания параметрического запроса необходимо создать запрос на выборку в режиме конструктора. В строке «Условия» поля, для которого нужно применить параметр, введите текст, который должен отображаться в диалоговом окне, заключив его в квадратные скобки, например: [Введите название группы:] (рис. 5). При запуске запроса с параметрами текст отображается в диалоговом окне без скобок (рис. 6). В результате мы получим только фамилии студентов обучающихся в группе, название которой введено в окне параметров.

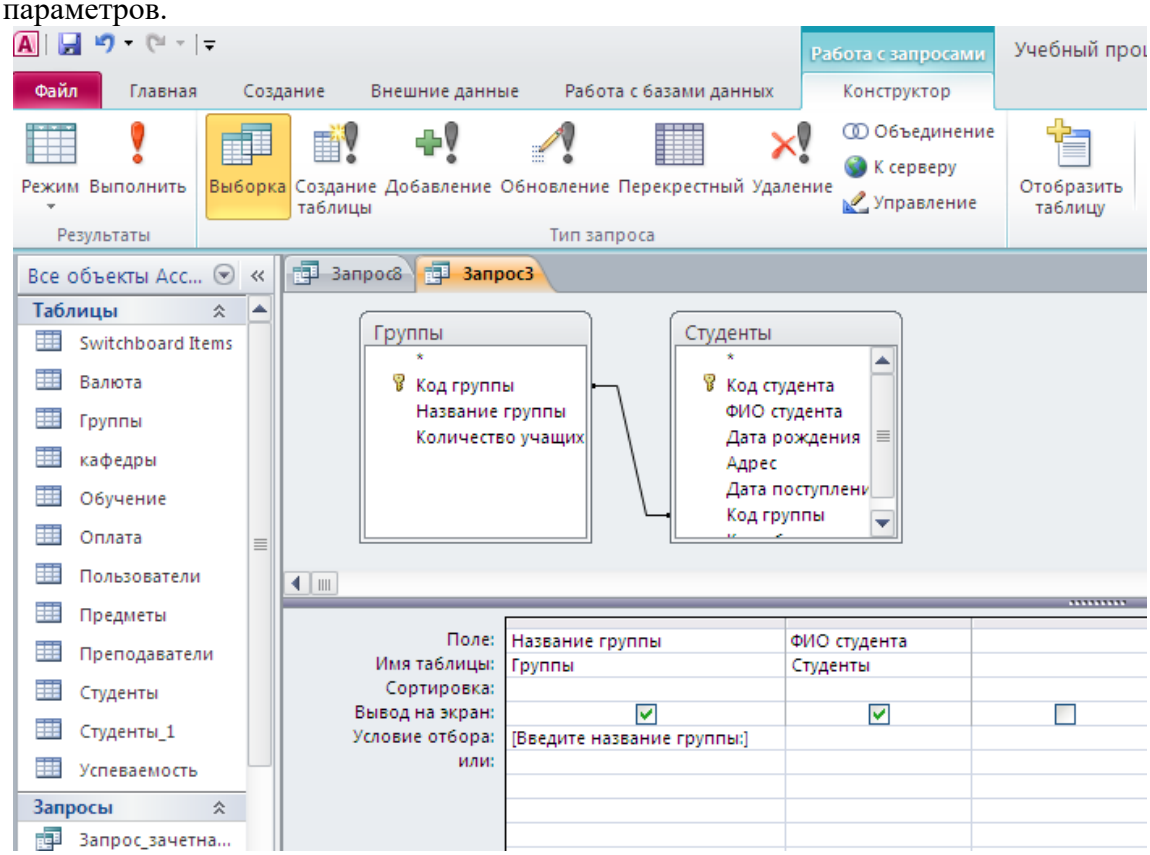



Рис. 5. Параметрический запрос.

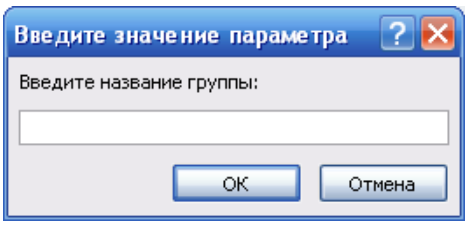



Рис. 6. Окно параметров.

Иногда необходимо немного изменить способ применения параметра запросом. Например, может потребоваться, чтобы запрос получал текстовую строку и сравнивал ее с частью поля. Здесь пригодится ключевое слово Like в сочетании с подстановочными знаками. Предположим, необходимо, чтобы запрос предлагал ввести страну

происхождения и при этом сравнение выполнялось, когда значение соответствующего поля содержит строку параметра. Для такого запроса в строке «Условия отбора» поля, для которого нужно применить параметр, необходимо ввести Like "*"&[, текст, который должен отображаться в сообщении, а затем ]&"*". При выполнении запроса с параметрами на экран будет выведено диалоговое окно, текст сообщения в котором будет отображаться без квадратных скобок, ключевого слова Like и подстановочных знаков. Когда запрос получит значение параметра, он сравнит его со значениями, которые содержат строку параметра. Например, для строки параметра «ст» будут найдены строки, в которых поле параметра имеет значение «Кыргызстан» и «Казахстан».

**Запросы с группировкой.**

Разрабатывая запрос в режиме конструктора, можно вычислить итоги, для отображаемых записей для этого в бланке запроса используется строка групповые операции.

Для вычисления итогов создается обычный запрос на выборку, выбираются нужные поля. Щелкнув на бланке запроса правой кнопкой мыши, в контекстном меню можно выбрать команду «Итоги». Или на вкладке «Конструктор выполнить команду «Итоги». В бланке запроса появится строка «Групповые операции», а в строке для каждого поля запроса появится элемент «Группировка». В поле, по которому следует провести подсчет, необходимо выбрать нужную статистическую функцию.

Например, если необходимо посчитать количество студентов в каждой группе необходимо создать запрос в режиме конструктора, выбрать таблицы «Группы», «Студенты». Из таблиц выбрать поля «Название группы», «ФИО студента». Выполнить команду «Итоги». В поле «ФИО студента» вместо «Группировка» необходимо указать статистическую функцию «Count», так как нам необходимо найти количество студентов (рис. 7).

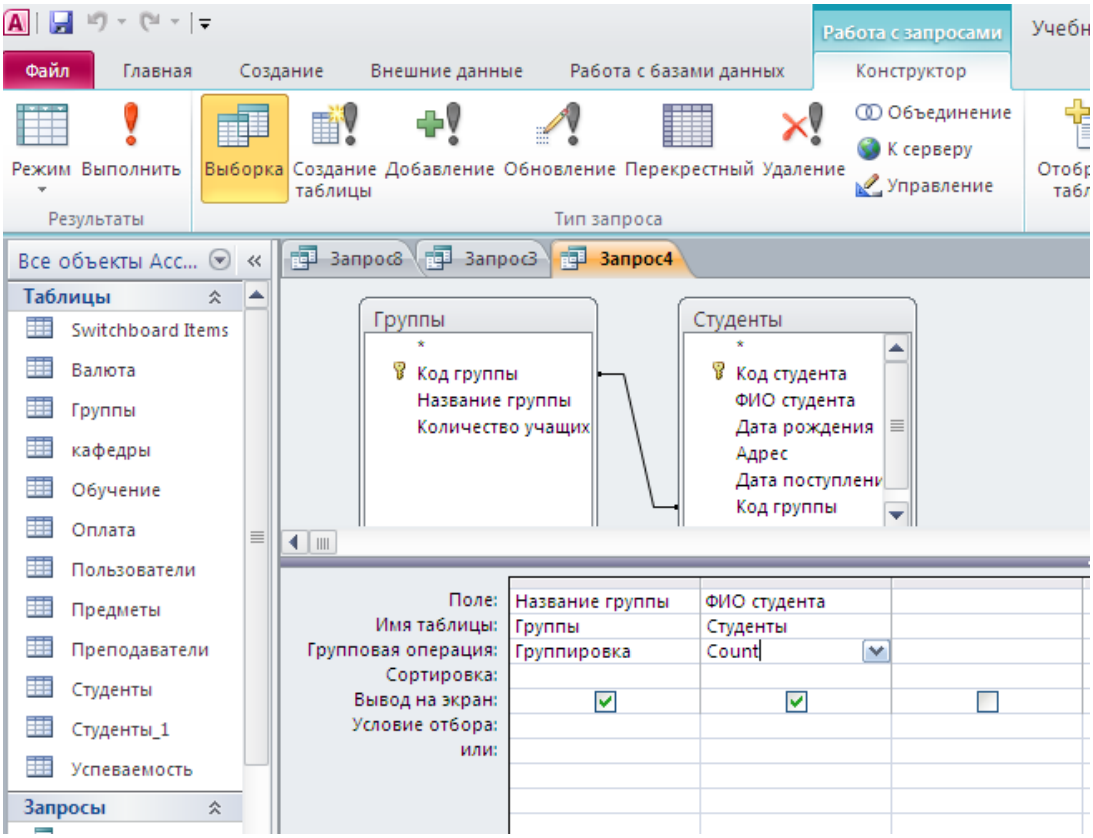



Рис. 7. Запрос с группировкой.

В запросах с группировкой используются следующие статистические функции:

- Avg- среднее арифметическое значение

- Count – количество записей
- First- значение в первой записи
- Last- значение в последней записи
- Max- максимальное значение
- Min- минимальное значение
- Sum-сумма
- StDev-оценка стандартного отклонения
- Var-оценка дисперсии

**Запросы с вычисляемыми полями.**

Вычисляемые поля - это поля, которые содержат информацию, основанную на данных, хранящихся в таблицах.

Вычисляемые поля могут содержать любые арифметические, логические операции, операции сравнения и различные встроенные функции.

Вычисляемое поле начинается со знака «двоеточие». После выполнения запроса перед двоеточием появляется текст «Выражение1», который можно заменить на любое соответствующее название поля.

Для создания вычисляемых полей можно воспользоваться построителем выражений (рис. 8).

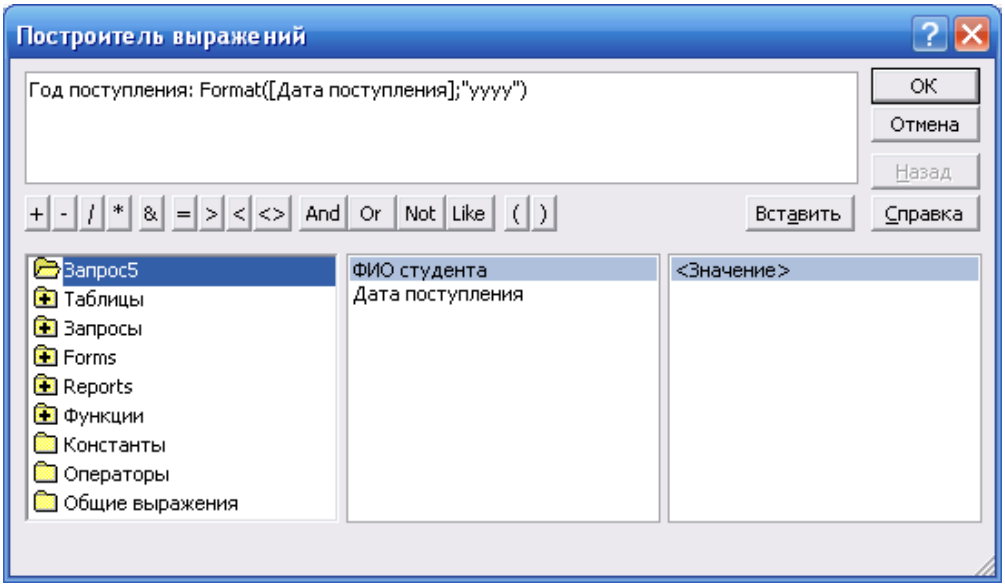



Рис. 8. Построитель выражений.

Выражения используются для создания вычисляемых полей в запросе. Предположим, необходимо отобразить год, в котором поступил студент, как часть запроса. Чтобы создать вычисляемое поле, помещаем следующее выражение в пустую ячейку строки «Поле» в запросе: Год Поставки: Format([ДатаПоставки],"гггг")

В этом случае выражение использует функцию Format для извлечения года из значений поля «Дата поступления». Функция Format представляет год в виде четырех цифр. Начало выражения — Год поступления: — присваивает конечному столбцу имя Год поступления.

Например для того чтобы вывести статус успеваемости студента, по среднему значению успеваемости студента необходимо:

1) Создать запрос с группировкой, и с использованием статистической функции Avg найти среднюю успеваемость (рис. 9). Сохранить запрос.

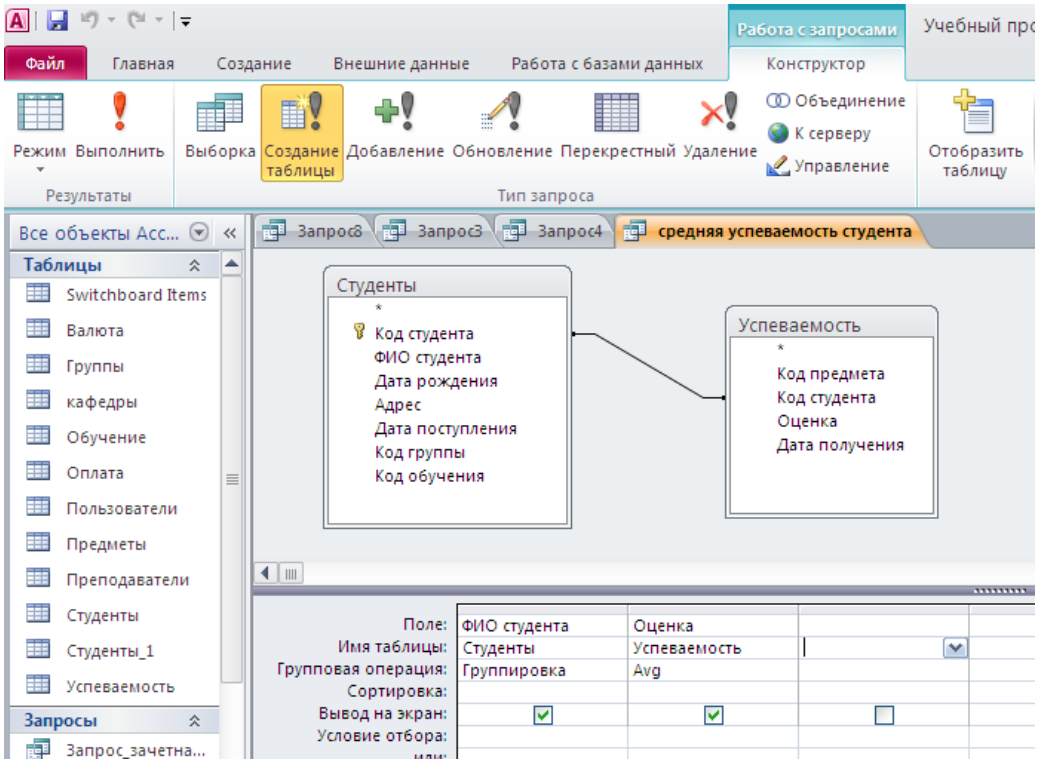



Рис. 9. Средняя успеваемость студента.

2) На базе созданного запроса в п. 1. Необходимо создать новый запрос. Добавить новое вычисляемое поле (рис. 10) с использованием условной конструкции. Если средняя оценка больше чем 4,5 – статус «отличник», больше чем 3,5 – «ударник», больше чем 2,5 – «успевающий», в противном случае «неуспевающий».

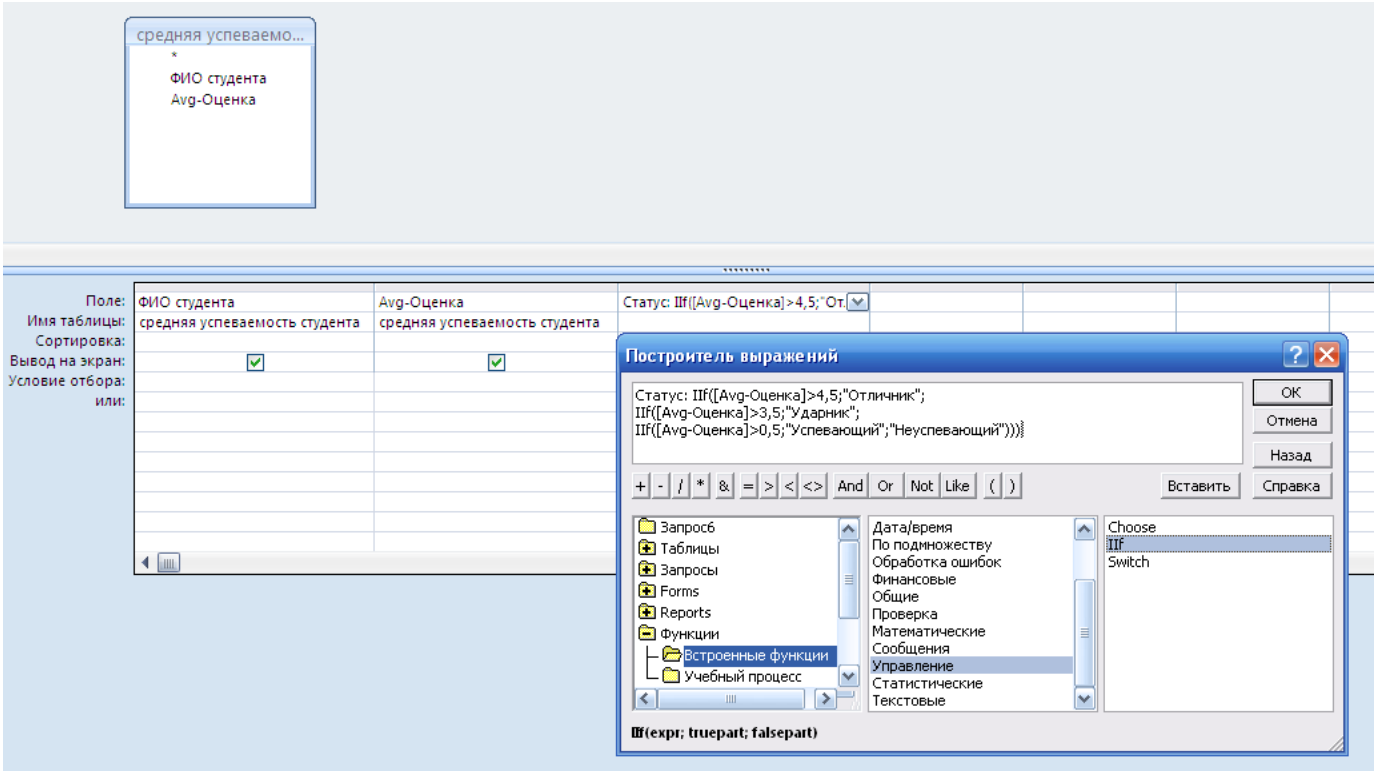
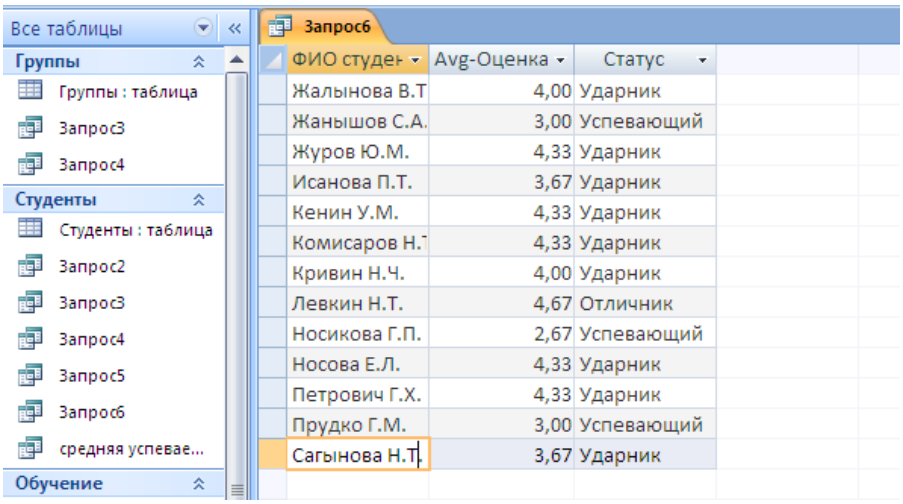



Рис. 10. Запрос с вычисляемым полем.

Выражение нового вычисляемого поля выглядит следующим образом:
*Статус:IIf([Avg-Оценка]>4,5;"Отличник";IIf([Avg-Оценка]>3,5;"Ударник";IIf([Avg-Оценка]>0,5;"Успевающий";"Неуспевающий")))*

Результат работы запроса выглядит следующим образом (рис. 11):

Рис. 11. Результат выполнения запроса с вычисляемым полем.




**Перекрестные запросы.**

С помощью перекрестных запросов, можно компактно отобразить выбранные данные и объединить информацию. Они служат для обобщения информации из двух и более таблиц или запросов.

Перекрестный запрос выполняет сведение данных по двум наборам значений, один из которых отображается в левой части таблицы, а другой – в верхней ее части.

Для создания перекрестного запроса необходимо создать обычный запрос на выборку в режиме конструктора, выбрать необходимые поля и выполнить команду «Перекрестный» на вкладке «Конструктор» (рис. 12). В бланке запроса, в строке «Перекрестная таблица» необходимо задать одно или несколько полей со значениями для заголовков строк, одно поле со значениями для заголовков столбцов и одно поле значений.

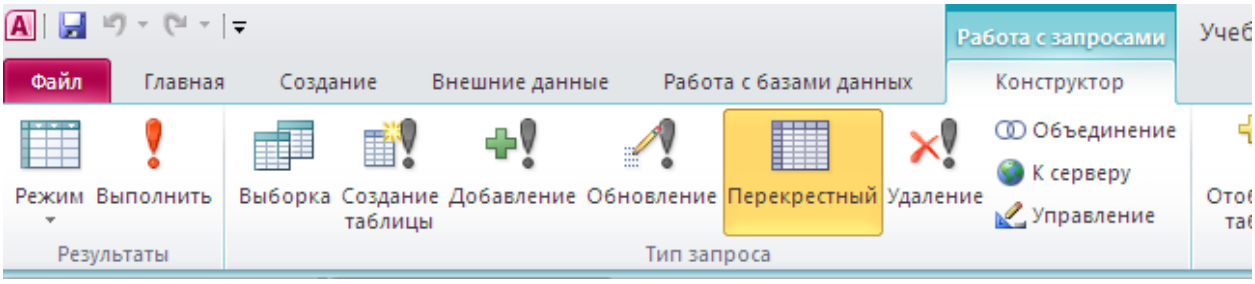



Рис. 12. Вкладка «Конструктор».

Например, для создания перекрестного запроса, для отображения успеваемости студентов необходимо:

1) Создать обычный запрос на выборку в режиме конструктора на основе таблиц «Студенты», «Предметы», «Успеваемость».
2) Выбрать поля «ФИО студента», «Название предмета», «Оценка».
3) Выполнить команду «Перекрестный» на вкладке «Конструктор».
4) В строке «Групповая операция» для поля «ФИО студента» указать «Группировка», для поля «Название предмета» указать «Группировка», для поля «Оценка» указать «Avg».
5) В строке «Перекрестная таблица» для поля «ФИО студента» указать «Заголовки столбцов», для поля «Название предмета» указать «Заголовки строк», для поля «Оценка» указать «Значение» (рис. 13)
6) Сохранить и выполнить запрос (рис. 14).

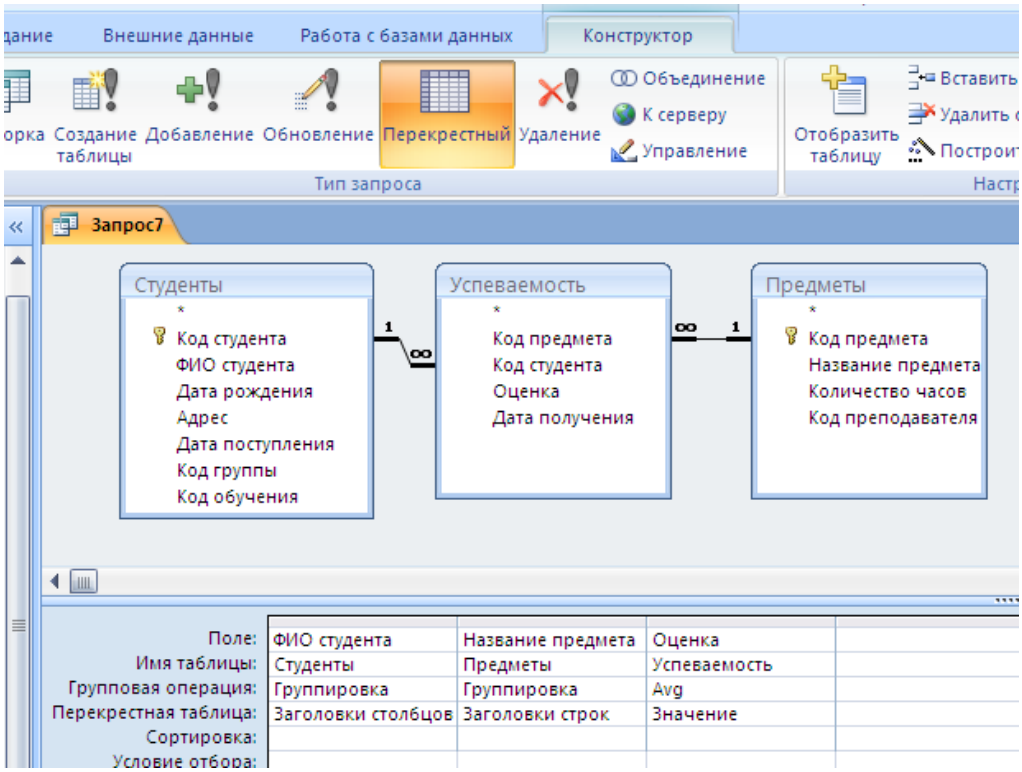



Рис. 13. Перекрестный запрос.

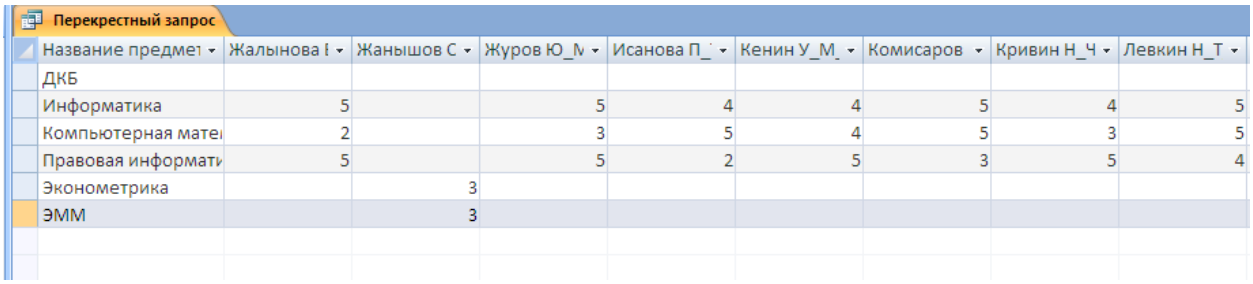



Рис. 14. Результат выполнения перекрестного запроса.

**Запросы действия.**

Существует четыре вида запросов действия:

- запрос на создание таблицы — создает новую таблицу на основе данных, которые уже размещены в одной или нескольких таблицах;
- запрос на удаление — удаляет записи из одной или нескольких таблиц на основании критериев, заданных в запросе;
- запрос на добавление — добавляет записи целиком или только указанные поля в одну или несколько таблиц;
- запрос на обновление — изменяет данные в существующих таблицах, обновляя записи с использованием критериев.

Необходимо помнить, что если открывается база данных, которая находится не в надежном расположении, и при этом ей не предоставлено состояние доверенной, то приложение Access по умолчанию не допускает выполнения запросов действия, то есть запросов на добавление, обновление, удаление или на создание таблицы. Если при попытке выполнения запросов действия ничего не происходит, проверьте, не появляется ли в строке состояния следующее сообщение: ***Данное действие или событие заблокировано в режиме отключения***. Если оно отображается, необходимо включить заблокированное содержимое. Для этого выполните следующие действия. На панели сообщений (рис. 15) нажмите кнопку «Включить содержимое».

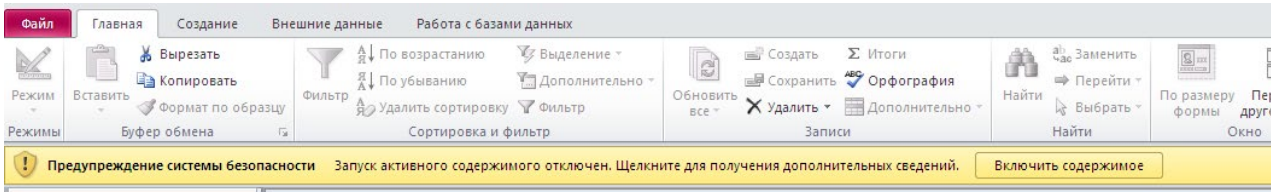


Рис. 15. Предупреждение системы безопасности.

Запросы действия, как и запросы на выборку, создаются с помощью конструктора запросов и могут редактироваться с использованием языка SQL. Для любого типа запроса на изменение сначала создается запрос на выборку, в котором разрабатываются критерии выбора записей, участвующих в запросе. Далее запрос на выборку преобразуется в один из запросов действия.

**Запрос на создание таблицы.**

Источником данных при создании новой таблицы являются уже существующие одна или несколько таблиц. Новая таблица может находиться в той же базе данных, где находится источник, или в любой другой. Создание таблиц с помощью запроса используется в основном для архивации данных. Предположим, у вас есть запрос на выборку, использующий несколько таблиц. При этом запрос используется часто, а данные в таблицах остаются неизменными. В таком случае проще и быстрее один раз поместить результаты выборки в новую таблицу и использовать ее вместо сложного запроса на выборку.

Запрос на создание таблицы создается путем преобразования запроса на выборку. Необходимо создать нужный запрос, а затем на вкладке «Конструктор» в группе «Тип запроса» выбрать команду «Создание таблицы». Откроется окно «Создание таблицы» (рис. 16). В раскрывающемся списке «имя таблицы» выбрать название новой таблицы. Установить переключатель в положение «в текущей базе данных». Нажать кнопку ОК. Запустить запрос на выполнение.

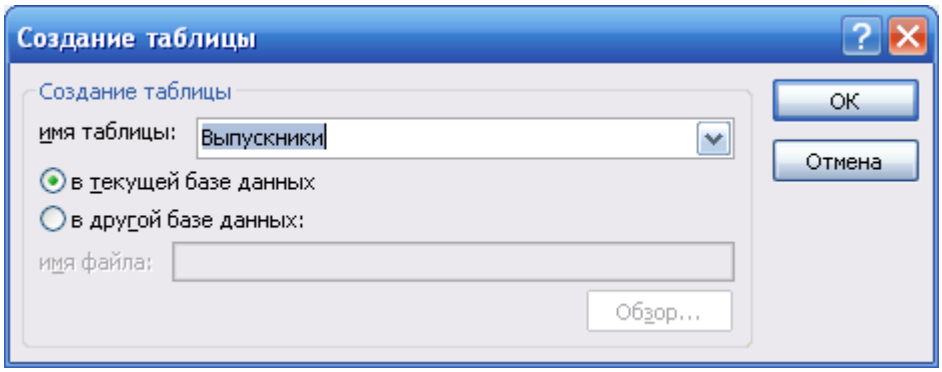


Рис. 16. Создание таблицы на базе запроса.

**Запрос на добавление данных.**

Запрос на добавление используется, когда необходимо добавить к имеющейся таблице новые записи. Исходных таблиц может быть одна или несколько, при этом они могут находиться в той же самой базе данных, что и результирующая, или в другой.

Как и в случае с запросом на создание таблицы, запрос на добавление данных создается из запроса на выборку. Необходимо создать нужный запрос на основе той таблицы, откуда будут извлечены данные, далее на вкладке «Конструктор» в группе «Тип запроса» выбрать команду «Добавление». Откроется окно «Добавление». В списке имя таблицы выбрать название результирующей таблицы. Установить переключатель в положение в текущей базе данных. Нажать кнопку ОК. В бланке запроса появиться новая строка «Добавление». Для каждого поля выборки указать поле назначения

результирующей таблицы, выбирая названия полей из раскрывающегося списка. Результирующий бланк запроса на добавление показан на рис. 17.

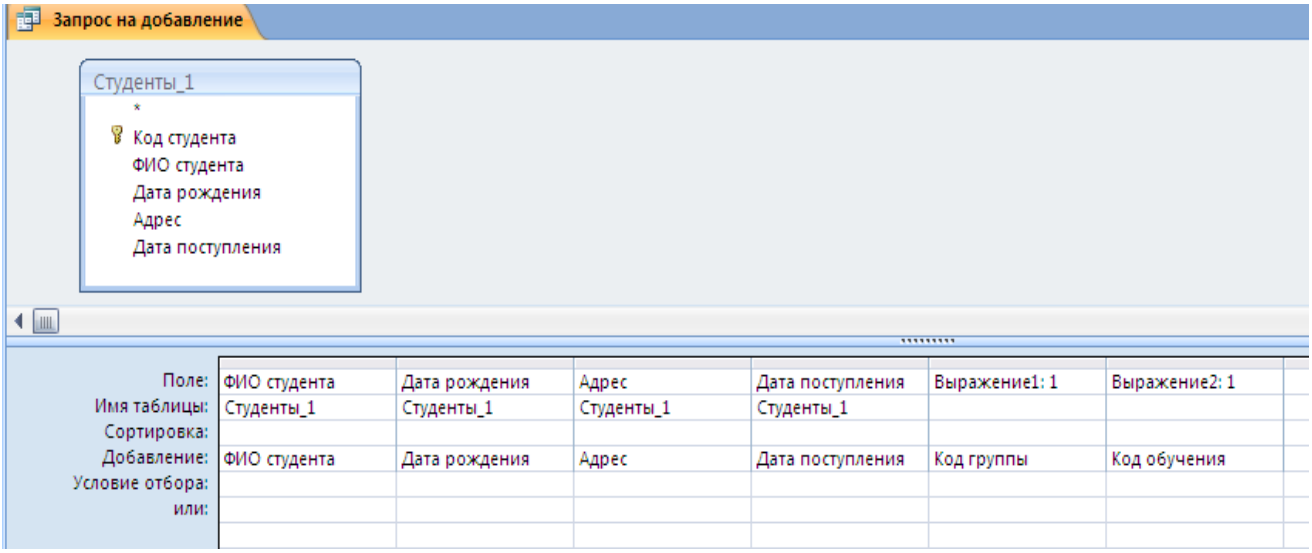



Рис. 17.Запрос на добавление.

**Запрос на обновление.**

Рассмотрим пример запроса на обновление. Допустим, что необходимо поменять адрес студента Жанышова С.А. Для этого необходимо создать обычный запрос на выборку, на основе таблицы «Студенты». Выбрать поля «ФИО студента» и «Адрес». Далее на вкладке «Конструктор» в группе «Тип запроса» выберите команду «Обновление». При такой операции Access добавляет строку «Обновление» в бланк запроса. Далее необходимо в столбце «ФИО студента» в строке «Условие отбора» ввести необходимое условие для обновления, в нашем случае это «Жанышов С.А.» . А в столбце «Адрес» в строке «Обновление» необходимо ввести новый адрес студента (рис. 18).

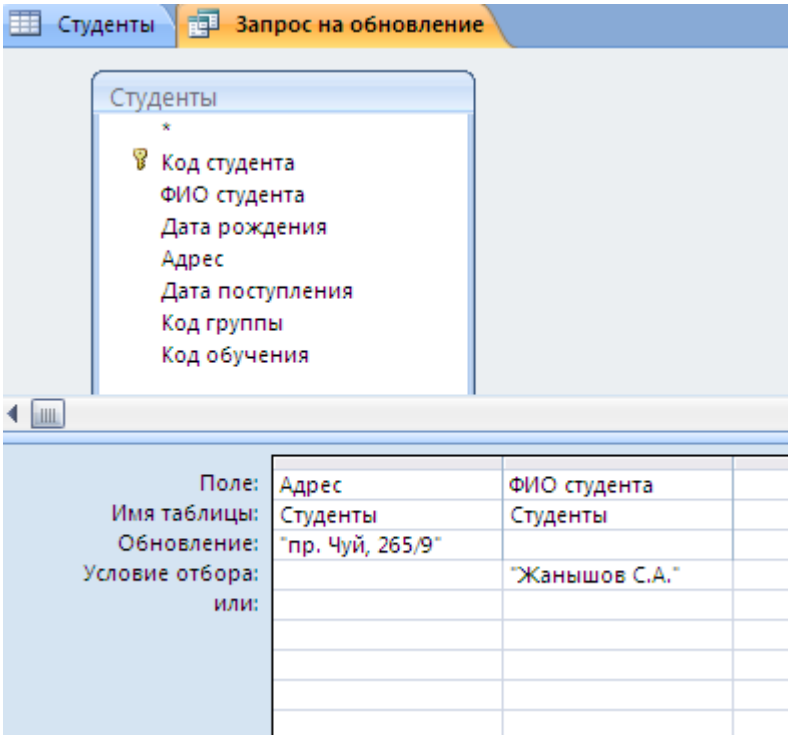



Рис. 18. Запрос на обновление.

**Запрос на удаление.**

Запрос на удаление записей используется, если требуется удалить запись целиком. Такой запрос удаляет все данные каждого поля записи вместе со значением ключа, который делает эту запись уникальной.

Для создания запроса на удаления, необходимо вначале создать нужный запрос на выборку, а далее на вкладке «Конструктор» в группе «Тип запроса» выбрать команду «Удаление». Далее необходимо в строке «Условие отбора» для нужного поля указать условие для удаления данных.

Важно помнить, что все изменения в базе данных, полученные в результате работы запросов действия, необратимы.

**Контрольные вопросы:**

1. Поиск и сортировка данных.
2. Мастер создания запросов.
3. Создание запроса в режиме конструктора.
4. Бланк запроса.
5. Условие отбора в запросах.
6. Группировка и сортировка данных.
7. Статистические функции.
8. Построитель выражений.
9. Использование встроенных функций в вычисляемых полях.
10. Запрос на обновление.
11. Запрос на удаление.
12. Запрос на добавление.
13. Параметрический запрос. Окно параметров.
14. Перекрестный запрос. Заголовки строк. Заголовки столбцов.

**Задания к лабораторной работе:**

База данных «Учебный процесс».

1. Отобрать всех студентов, фамилии которых начинаются на букву К.
2. Вывести фамилии студентов, внесших оплату в течении октября 2015 года.
3. Отобрать всех студентов, родившихся в зимние месяцы 1996г.
4. Отобрать информацию по конкретному преподавателю (ФИО преподавателя, название кафедры, название предмета, количество часов). ФИО преподавателя вводится пользователем в окне параметров.
5. Вывести список преподавателей, имеющих степень кандидата или доктора наук.
6. Отобрать информацию: ФИО, название группы, дата рождения, адрес, телефон, успеваемость; по конкретному студенту. ФИО вводится пользователем в окне параметров.
7. Создать зачетную книжку (предмет, количество часов, ФИО преподавателя, оценка, ФИОстудента).
8. Вывести студентов и их среднюю успеваемость.
9. Создать перекрестный запрос - строки – ФИО студента, столбцы-предметы, на пересечении –оценки.
10. Создать перекрестный запрос - строки –ФИО преподавателя, столбцы-название предмета, на пересечении –количество часов.
11. Создать запрос с условием в вычисляемом поле. Необходимо вывести всех студентов, их успеваемость (средняя оценка по всем предметам). А также их статус: двоечник (средняя оценка <2.7); троечник (средняя оценка >=2.7 И <3.5); ударник (средняя оценка >=3.5 И <4.5); отличник (средняя оценка >=4.5). (Рис. 10)

12. Создать запрос с вычисляемым полем. Необходимо перевести все суммы платежа в сомы.
    :iif([код валюты]=2;[сумма платежа]*61; iif([код валюты]=3;[сумма платежа]*1.1;[сумма платежа]*))
    *(если 2 - доллар и 3 - рубль)*
13. Отобрать всех студентов (фио, группа), которые внесли оплату в диапазоне от 200 до 300 $.
14. Вывести информацию о сумме оплаты, внесенной студентами каждой группы и наименование группы. Например, студенты группы ЭП-1 внесли 45620 сом, студенты группы БИА-2 внесли 4566 сом…..
15. Создать запрос на обновление. Преподавателю присвоили новую ученую степень. Необходимо обновить данные. (Например, Сидоров был старшим преподавателем, стал доцентом).
16. Создать запрос на добавление. Из колледжа на 3-ий курс перевелись 5 студентов в определенную группу, информация по этим студентам хранится в новой таблице Студенты1. Необходимо добавить их в таблицу студенты.
17. Создать запрос на удаление. Какая-то группа была расформирована по некоторым причинам. Необходимо удалить ее.
18. Создать запрос, который вычисляет стоимость за обучение для отличников и ударников со скидкой (для отличников – скидка 40 %, для ударников – скидка 20 %) .
19. Создать запрос с вычисляемым полем, позволяющий посчитать возраст студента.
    :Year(Date())-Year([дата рождения])
    *Year() – Извлекает год из даты*
    *Date() – Текущая дата*
20. Создать запрос, который позволяет посчитать зарплату преподавателя. Зарплата преподавателя = количество часов (по всем предметам, которые он ведет) *60.
21. Создать запрос, позволяющий вывести ФИО студента, дату рождения и название дня недели, в который он родился.
    :WeekdayName(Weekday([дата рождения]);0;1)
    *WeekdayName() – возвращает название дня недели*
    *Weekday() – возвращает число соответствующее дню недели*

22. Создать запрос, позволяющий вывести ФИО студента, дату его рождения, и текст «С днем рождения», если у него сегодня день рождения.
    :IIf(Month(Date())=Month([дата рождения]) And Day(Date())=Day([дата рождения]);"С днем рождения")
    *Month() – Извлекает месяц из даты*
    *Day() – Извлекает число из даты*
23. Создать диаграмму в Access (сводную диаграмму) в которой отображается количество студентов рожденных в понедельник, вторник, … .

## Лабораторная работа №3. Использование форм в базе данных.

Формой называется объект базы данных, используемый для просмотра, ввода или изменения данных. По сравнению с ручным вводом данных, интерфейс, основанный на формах, позволяет автоматизировать процесс путем добавления в форму функциональных элементов, ввести проверку данных, настроить представление данных для просмотра.

**Создание формы с помощью инструмента «Форма».**

Самый простой способ создания формы — это использование инструмента «Форма». Необходимо открыть базу данных «Учебный процесс». В области переходов щелкните кнопкой мыши дважды на названии таблицы «Группы», чтобы открыть таблицу. Далее перейдите на вкладку «Создание» и в группе «Формы» щелкните кнопкой мыши на инструменте «Форма». Автоматически созданная форма отобразится в режиме макета (рис. 1).

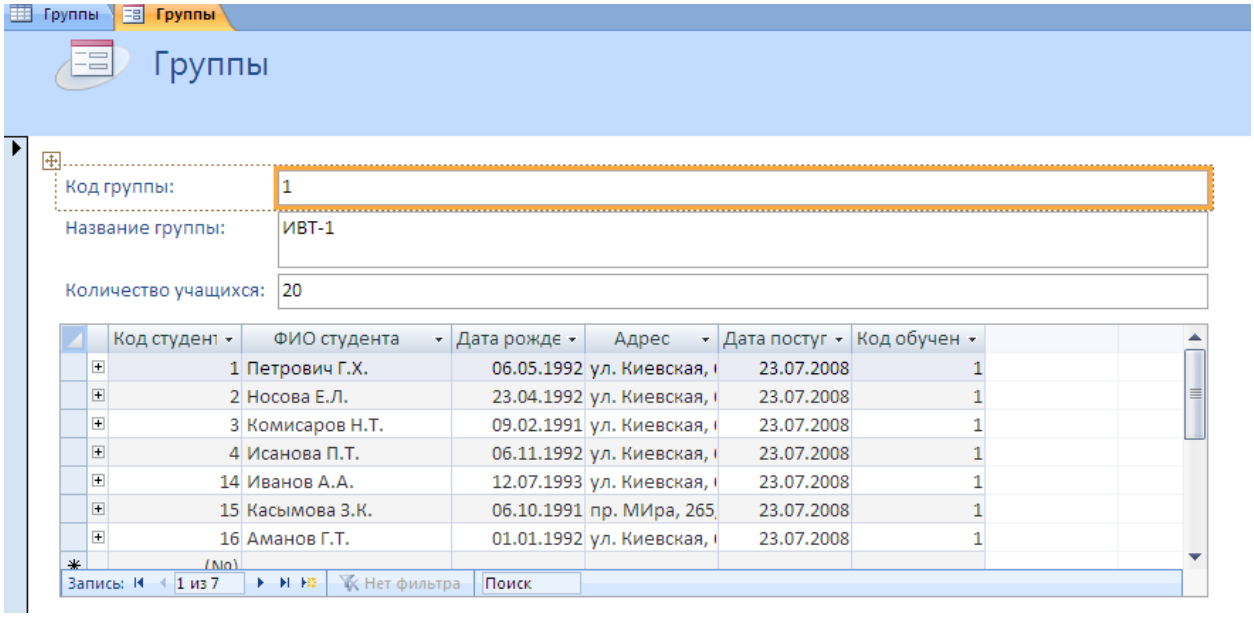



Рис. 1. Автоматически созданная форма.

Форма содержит все поля из таблицы «Группы» и отображает первую запись из таблицы. Кроме того, так как существует таблица «Студенты», связанная с таблицей «Группы» отношением «один ко многим», в форму добавлена та часть таблицы «Группы», записи которой относятся к выбранной группе. Однако, в случае существования нескольких таблиц, связанных с выбранной отношением «один ко многим», они не добавляются на форму, чтобы не загромождать ее.

Созданная форма, обладает основными функциями формы, позволяющими вводить и редактировать данные. Кроме того, любая форма, даже созданная таким простым путем, может быть доработана в режиме конструктора.

**Создание формы с нуля.**

Такой вариант создания формы предполагает максимальную гибкость и функциональность и используется, если требуемого результата нельзя достичь с помощью автоматизированных средств создания форм. Сначала необходимо создать пустую форму. Далее необходимо перейти на вкладку «Создание» и в группе «Формы» выбрать инструмент «Пустая форма». Откроется пустой макет формы, а в правой части окна откроется список доступных полей (рис. 2). Далее необходимо щелкнуть дважды кнопкой мыши на названиях полей «ФИО преподавателя», «Ученная степень», «Должность» с таблицы «Преподаватели». В форму будут добавлены соответствующие поля. В списке полей отдельно выделятся поля связанной таблицы «Предметы». В списке доступных полей связанной таблицы необходимо щелкнуть кнопкой мыши на поле «Название предмета». Будет добавлена таблица с информацией о читаемых предметах выбранного преподавателя.

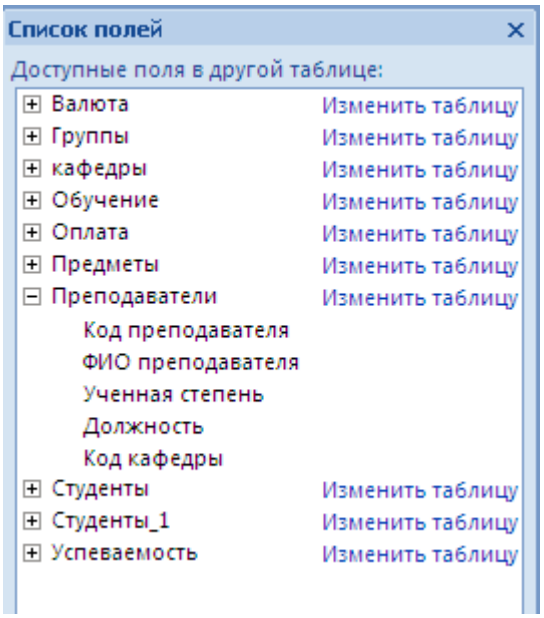



Рис. 2. Список полей в пустой форме.

Рассмотрим пример. Допустим, необходимо создать форму для ввода данных в таблицу успеваемость. Для этого создаем пустую форму, в списке доступных полей двойным щелчком мыши выбираем поля «Оценка», «Дата получения оценки» из таблицы «Успеваемость». Далее из доступных полей связанной таблицы выбираем «Название предмета» из таблицы «Предметы» и «ФИО студента» из таблицы «Студенты». В результате получаем требуемую форму (рис. 3).

Рис. 3. Форма для ввода данных в таблицу «Успеваемость».

Для изменения существующей записи необходимо сначала найти нужную запись. Между записями можно перемещаться последовательно, используя панель навигации, или же использовать поиск нужной записи.

Рассмотрим эту панель, расположенную в нижней части формы (рис. 4).

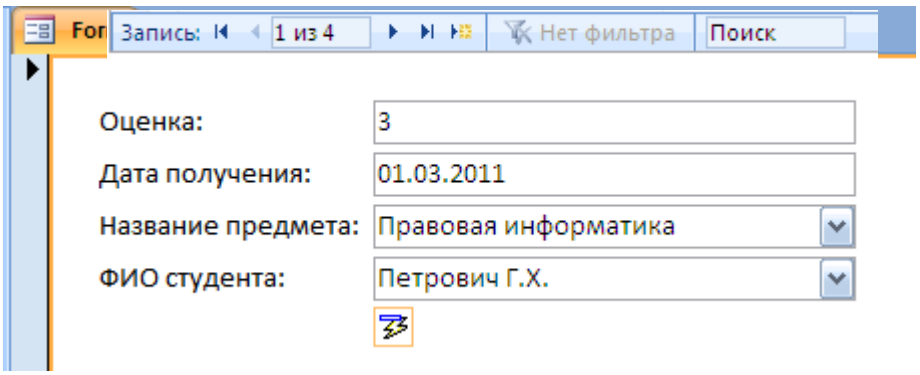



Рис. 4. Панель навигации.

Панель навигации содержит следующие элементы управления:

- кнопка перемещения к первой записи таблицы;
- кнопка перемещения к предыдущей записи;
- поле, указывающее номер текущей записи; в это поле можно ввести номер требуемой записи, если он известен;
- кнопка перемещения к следующей записи;
- кнопка перемещения к последней записи в таблице;
- кнопка создания новой записи;
- описание текущего фильтра;

- поле поиска, в которое можно ввести искомое значение.

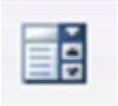
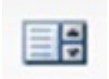
**Доработка формы в режиме конструктора.**

Для каждой формы существуют три режима, в которых она может быть представлена.

- Режим формы — в таком виде форма будет использоваться для работы с данными.
- Режим макета — упрощенный интерфейс создания и редактирования формы.
- Режим конструктора — наиболее гибкий и точный интерфейс для редактирования формы, но и самый сложный.

При открытии конструктора форм на вкладке «Конструктор» появляется панель элементов управления. С помощью кнопок на панели элементов управления в форму можно добавлять различные объекты, типы которых перечислены в таблице.

Таблица 1. Элементы управления.

| Название элемента | | Назначение |
|---|---|---|
| Поле |  | Используется для ввода и отображения информации полей таблиц и запросов, а также для вывода результатов вычислений |
| Подпись |  | Подписи создаются автоматически вместе с элементами типа текстовое поле, поле со списком и т. п. Они разъясняют смысл поля ввода. Дополнительные надписи могут использоваться для заголовков и пояснений |
| Кнопка |  | Щелчок на такой кнопке выполняет команду, с помощью которой можно перейти в другую форму, переместиться по записям и выполнить многие другие операции |
| Поле со списком |  | Разрешает как непосредственный ввод значения в поле, так и его выбор в раскрывающемся списке предопределенных значений |
| Список |  | Позволяет выбирать данные из набора пунктов и не разрешает непосредственный ввод значений |
| Подчиненная форма/отчет |  | Вставляет в форму (или отчет) область с другой формой (или отчетом), связанной с главной |
| Линия |  | Добавляет прямую линию |
| Прямоугольник |  | Рисует прозрачный или непрозрачный прямоугольник с цветной границей любой толщины |
| Присоединенная рамка объекта |  | Объект OLE, связанный с полем данных таблицы или запроса |
| Группа переключателей |  | Группа переключателей ссылается на некоторое поле таблицы. Каждый переключатель группы соответствует определенному целочисленному значению этого поля |
| Флажок |  | Предназначен для представления полей типа да/нет. Отмеченный квадратик соответствует величине «да», а пустой — «нет» |

| | | |
|---|---|---|
| Переключатель |  | Элемент группы переключателей, в которой может быть отмечен только один из них (его кружок выглядит зачерненным) |
| Выключатель |  | Позволяет вводить информацию типа да/нет. Величине «да» или «истина» соответствует утопленное положение выключателя |
| Вкладка |  | Позволяет разместить на одном и том же пространстве экрана несколько наборов элементов управления. Удобен в тех случаях, когда элементы легко разделяются на логические группы |
| Вставить диаграмму |  | |
| Свободная рамка объекта |  | Объект OLE, не связанный ни с каким источником данных |
| Рисунок |  | Рисунок любого графического формата с рамкой |
| Разрыв страниц |  | Линия, по которой формируется перевод страницы при выводе формы или отчета на принтер |
| Вставить гиперссылку |  | Создание ссылки на Web-страницу, рисунок, адрес электронной почты или программу |
| Вложение |  | Присоединенные файлы |

**Структура формы**

Перед тем как приступить к добавлению элементов управления на форму, необходимо понять, из каких структурных частей форма состоит, так как каждый раздел формы характеризуется не только местом расположения, но и допустимым набором элементов управления.

Любая форма может включать следующие разделы.

- Заголовок формы — находится в верхней части формы. При печати выводится только на первой странице. Может включать графику, текст. Добавляется одновременно с областью примечаний формы.
- Верхний колонтитул — используется для печати формы, поэтому отображается только в режиме предварительного просмотра. При печати выводится вверху каждой страницы. Добавляется одновременно с нижним колонтитулом.
- Область данных — основная часть формы, содержащая данные. Здесь располагаются присоединенные элементы управления, а также любые другие. Количество областей данных при печати соответствует количеству записей в источнике данных.
- Нижний колонтитул — используется аналогично верхнему колонтитулу, при печати выводится в нижней части каждой страницы. Добавляется одновременно с верхним колонтитулом.
- Примечания — раздел используется для пояснений и примечаний, а также итоговых полей. Печатается только на последней странице. Добавляется одновременно с областью заголовка.

**Свойства формы**

При работе с формой в режиме конструктора форм может понадобиться окно свойств формы. Если данное окно не открыто, то для ее вызова необходимо вызвать контекстное меню, правой кнопкой мыши, щелкнув в верхний левый угол на пересечении двух линеек, и выбрать команду свойства (рис. 5). В результате появиться окно свойств (рис. 6).

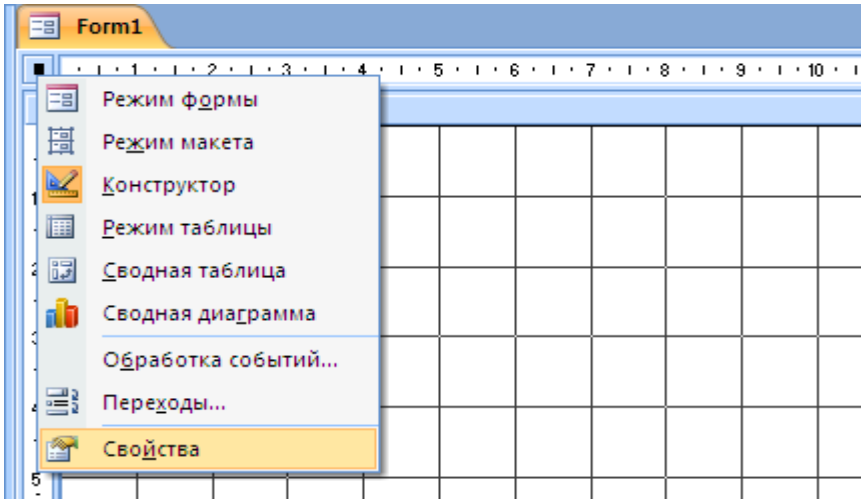



Рис. 5. Вызов окна свойств формы.

Окно свойств состоит из пяти вкладок.

- Макет — содержит свойства, определяющие вид макета формы: размер, линии прокрутки, наличие кнопок, формат печати. В большинстве случаев подходят значения по умолчанию.
- Данные — свойства этой вкладки определяют взаимодействие с источником данных.
- События — вкладка позволяет назначить макрос или процедуру для какого-либо события, происходящего с формой.
- Другие — настройки вида и поведения формы.
- Все — содержит все свойства вышеперечисленных вкладок.

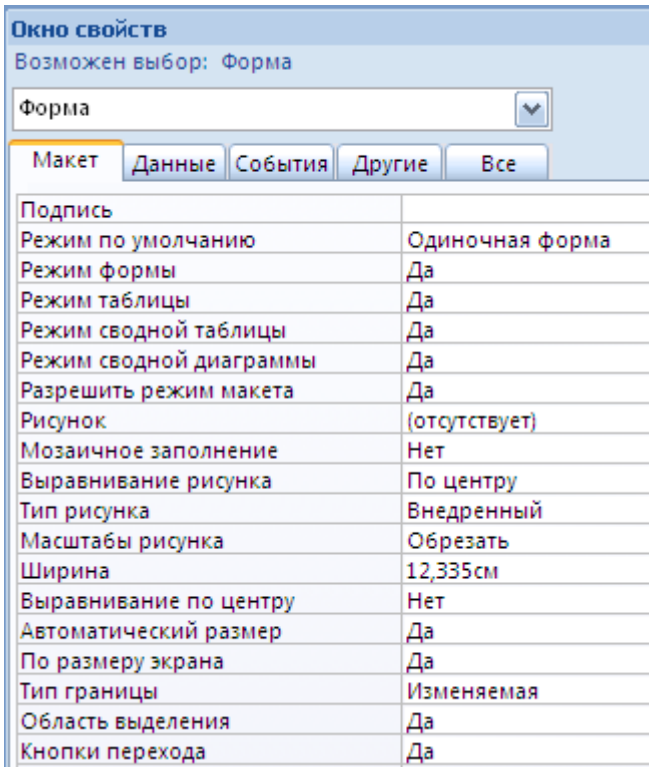



Рис. 6. Окно свойств формы.

**Создание формы в режиме конструктора.**

Рассмотрим пример создания формы в режиме конструктора для отображения зачетной книжки студента. Для правильного отображения данных в форме, предварительно необходимо создать запрос, который будет включать следующие поля: «Код студента», «Название предмета», «Количество часов», «ФИО преподавателя», «Дата получения», «Оценка», «Оценка прописью». Созданный запрос назовем «Запрос_зачетная_книжка» (рис. 7).

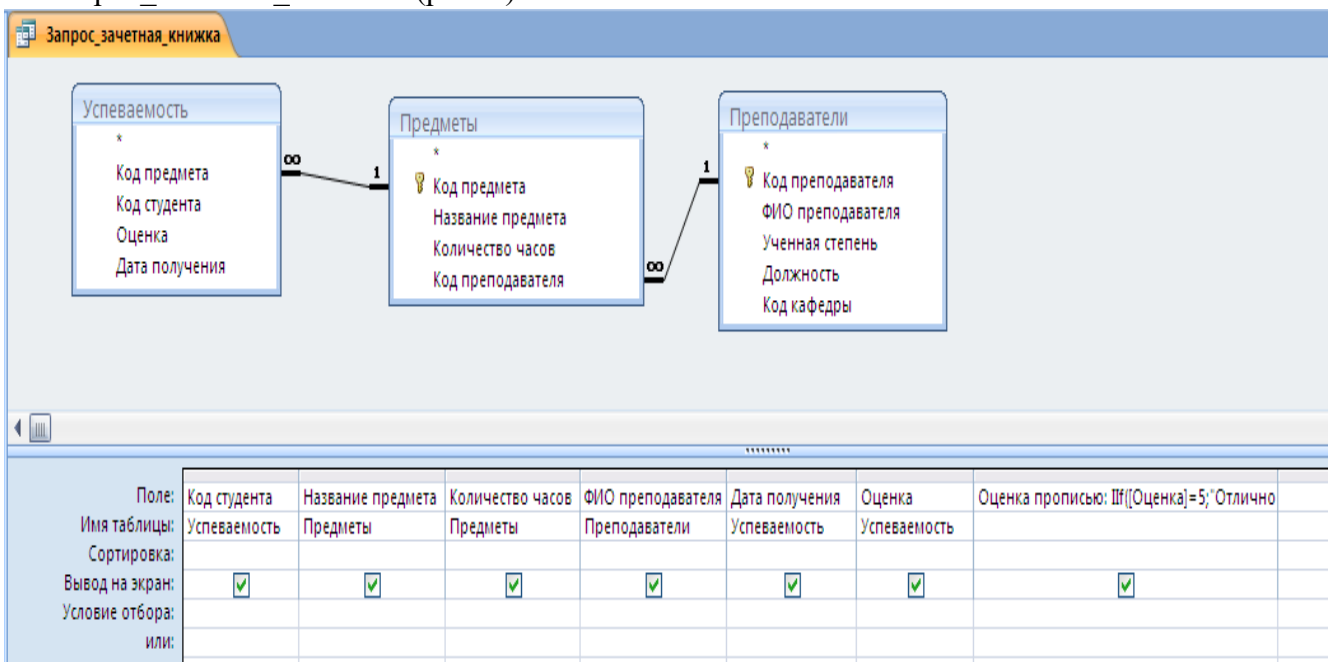
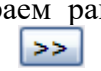



Рис. 7. Запрос для формы.

Создадим новую форму в режиме конструктора, для этого выполним команду «Конструктор форм» на вкладке «Создание». В режиме конструктора форм в окне «Окно

свойств» для формы, на вкладке «Данные» для свойства «Источник записей» выберем из раскрывающегося списка таблицу «Студенты».

Далее на вкладке «Конструктор» на панели элементов управления выбираем элемент «Поле со списком» и рисуем элемент в верхней части области данных. Появляется мастер создания полей со списком. На первом шаге работы мастера выбираем «Поиск записи в форме на основе значения, которое содержит поле со списком» и нажимаем кнопку «Далее». На втором шаге, двойным кликом левой кнопки мыши выбираем поле «ФИО студента» и нажимаем кнопку «Далее». Затем можно нажать кнопку «Готово».

Далее также на панели элементов управления выбираем элемент «Подчиненная форма/отчет» и рисуем его в области данных формы, появляется «Мастер подчиненных форм». На первом шаге работы мастера выбираем «Имеющиеся таблицы и запросы» и нажимаем кнопку «Далее». В поле «Таблицы и запросы» выбираем ранее созданный запрос «Запрос: Запрос_зачетная_книжка». С помощью кнопки 

выбираем все доступные поля и нажимаем кнопку «Далее». На следующем шаге необходимо самостоятельно определить поля связи между главной и подчиненной формой. Такими полями является поле «Код студента» (рис. 8). Выбрав поля связи, нажимаем кнопку «Далее».

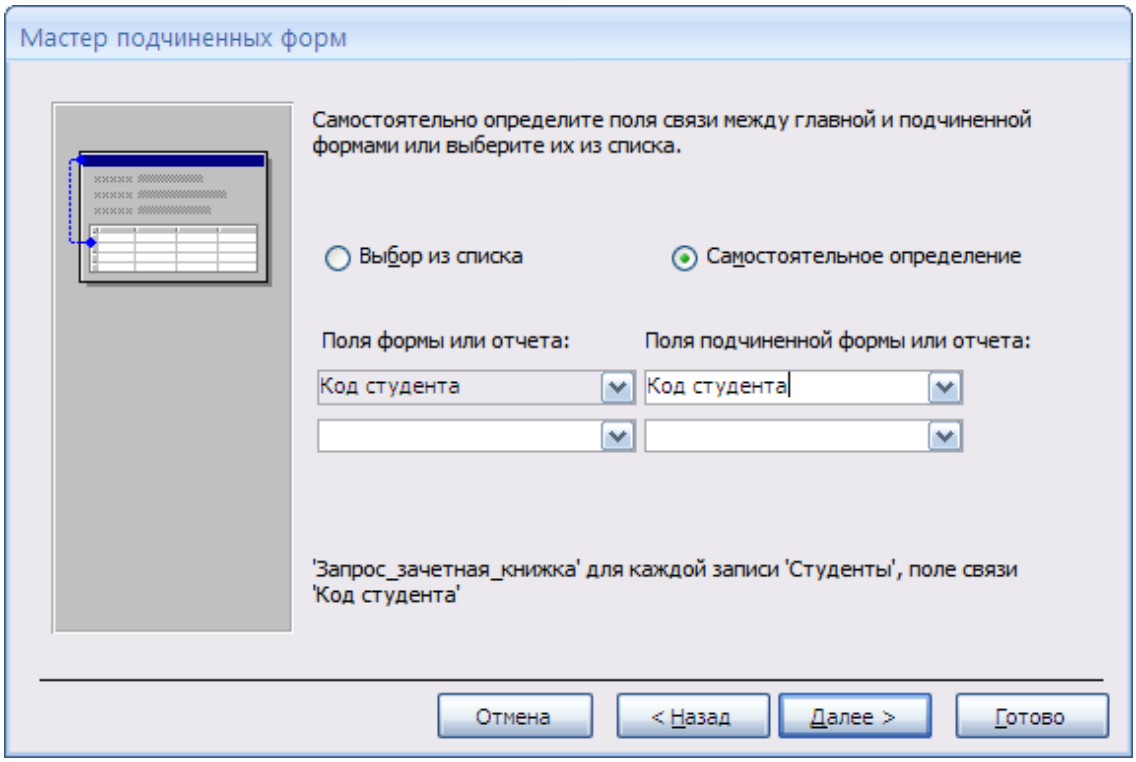



Рис. 8. Мастер подчиненных форм.

На следующем шаге задаем имя подчиненной формы и нажимаем кнопку «Готово». Далее настраиваем элементы управления, меняем их макет и сохраняем форму под названием «Зачетная книжка» (рис. 9).

Полученная форма называется составной формой, так как она состоит из главной формы, где выбирается студент, и подчиненной формы, в которой непосредственно и отображается успеваемость студента.

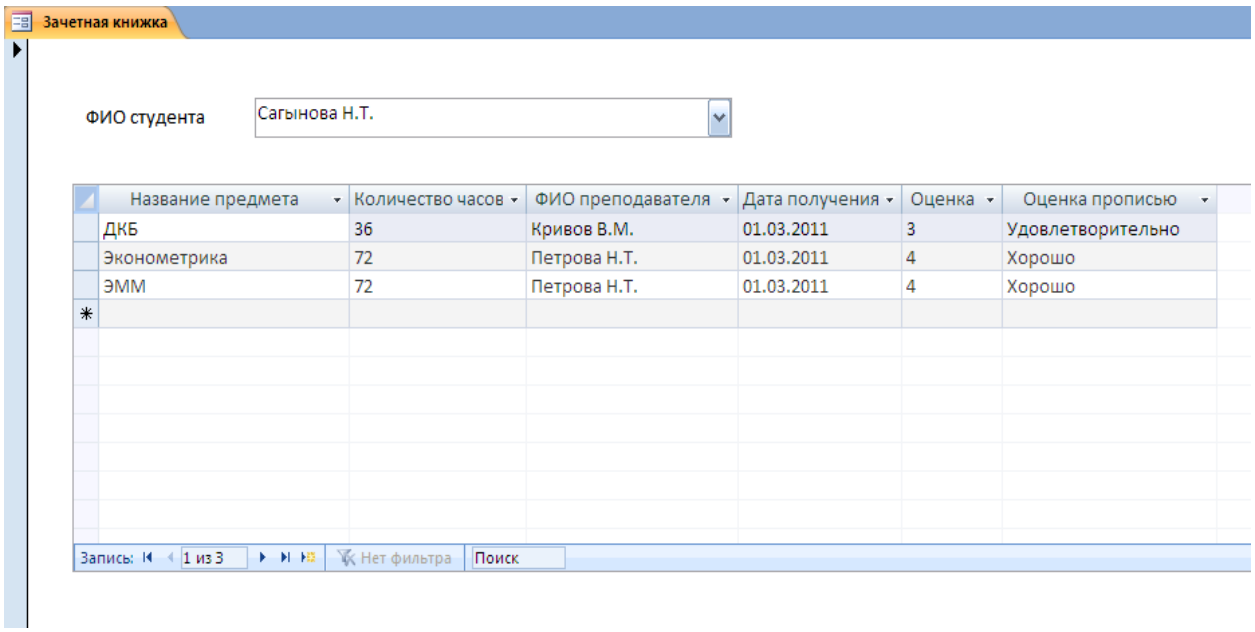



Рис. 9. Форма «Зачетная книжка».

Рассмотрим еще один пример. Необходимо создать форму для ввода информации в таблицу «студенты» (для поля «код группы» использовать элемент поле со списком, для поля «код вида обучения» - группу переключателей).

Для создания данной формы воспользуемся созданием формы с помощью пустой формы, а далее доработаем форму в режиме конструктора.

Выберем команду «Пустая форма» на вкладке «Создание». Откроется пустой макет формы, из списка доступных полей (рис. 2) двойным кликом левой кнопки мыши выберем следующие поля: «ФИО студента», «Дата рождения», «Адрес», «Дата поступления». Далее из доступных полей в связанной таблице двойным кликом левой кнопки мыши выбираем «Название группы». Автоматически формируется список существующих групп и производится их привязка к полю «Код группы» таблицы «Студенты». Далее из доступных полей в связанной таблице двойным кликом левой кнопки мыши выбираем «Вид обучения». Автоматически формируется список видов обучения и производится их привязка к полю «Код обучения» таблицы «Студенты». Далее сохраняем форму, называем ее «Студенты» и переходим в режим конструктора с помощью команды «Режим»/«Конструктор» на вкладке «Главная» (рис. 10).

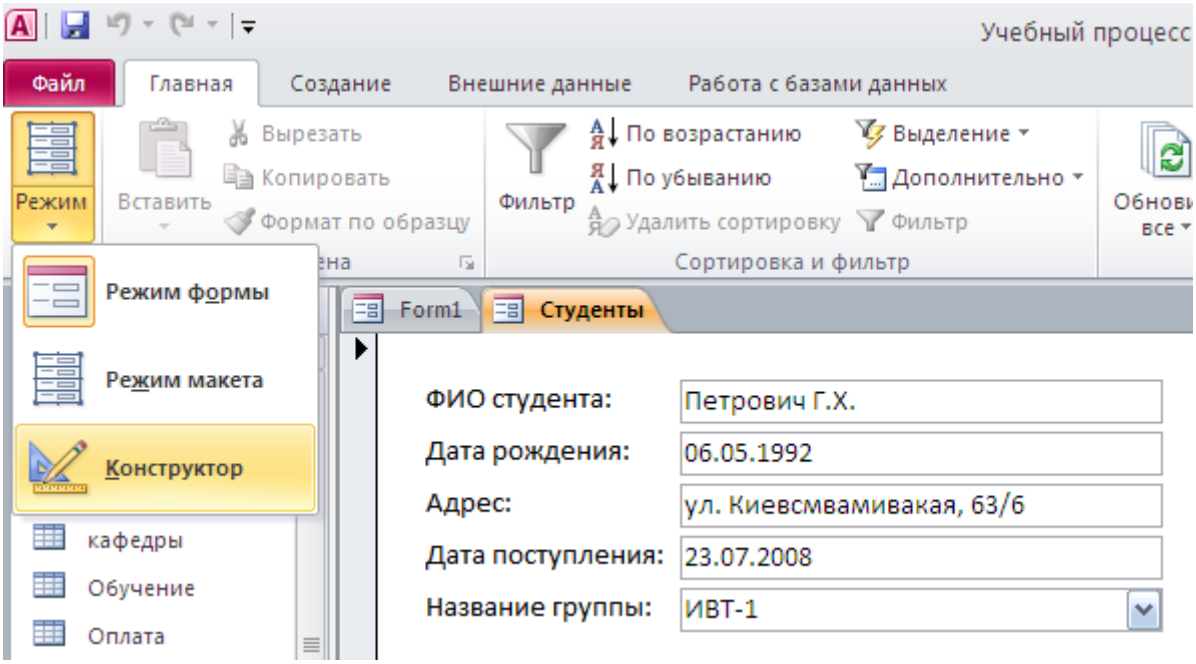



Рис. 10. Переход в режим конструктора.

В режиме конструктора на панели элементов управления выбираем элемент «Группа переключателей» и рисуем его в области данных под полем «Вид обучения». Появляется окно мастера «Создание группы переключателей».

На первом шаге задаются подписи для каждого переключателя, так как в нашей базе данных, в таблице «Обучение» два вида обучения «Бюджет», «Контракт», следовательно, группа переключателей будет состоять из двух переключателей (рис. 11).

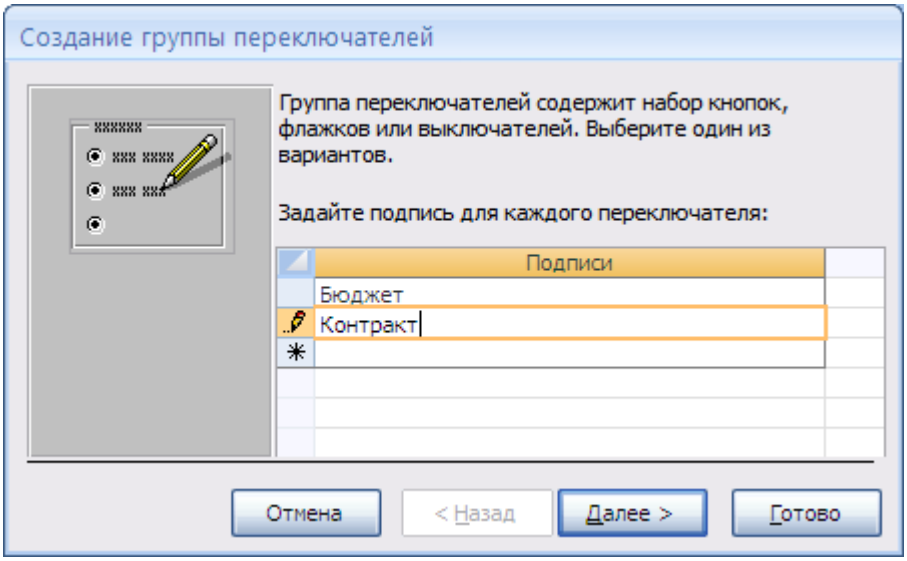



Рис. 11. Мастер создания группы переключателей.

На втором шаге можно выбрать требуется ли задать переключатель по умолчанию или нет.

На третьем шаге каждому переключателю присваивается значение, в нашем случае можно согласиться с предложением мастера.

На четвертом шаге обязательно необходимо сохранить значение в поле «Код обучения», для сохранения новых или измененных записей в таблице «Студенты».

На последующих шагах можно выбрать внешний вид переключателей и стиль, на последнем шаге необходимо указать имя группы переключателей «Вид обучения» и нажать кнопку «Готово». Группа переключателей создана.

Далее необходимо удалить созданное ранее поле со списком для вида обучения.

Форма готова, но можно расширить ее возможности, добавив такой элемент управления, как «Кнопка».

Для этого необходимо выбрать элемент «Кнопка» на панели элементов на вкладке «Конструктор» и нарисовать кнопку справа от созданной ранее группы переключателей. Появится мастер создания кнопок.

На первом шаге работы мастера можно выбрать категорию и непосредственно само действие. Выберем категорию «Переходы по записям» и действие «Найти запись» и нажмем кнопку «Далее».

На следующем шаге можно выбрать внешний вид кнопки и нажать кнопку «Готово». Создадим еще одну кнопку, выберем категорию «Работа с формой» и действие «Закрыть форму».

Кроме этого можно добавить рисунок. Для этого на панели элементов управления, на вкладке «Конструктор» выбираем элемент «Рисунок». И рисуем его в правой части области данных, появляется окно для выбора источника рисунка, необходимо выбрать требуемый рисунок и нажать кнопку «ОК».

Форма «Студенты» создана. Сохраняем форму и открываем ее в режиме формы (рис. 12). Для того чтобы переместится по записям или добавить новую запись можно воспользоваться панелью навигации (рис. 4).

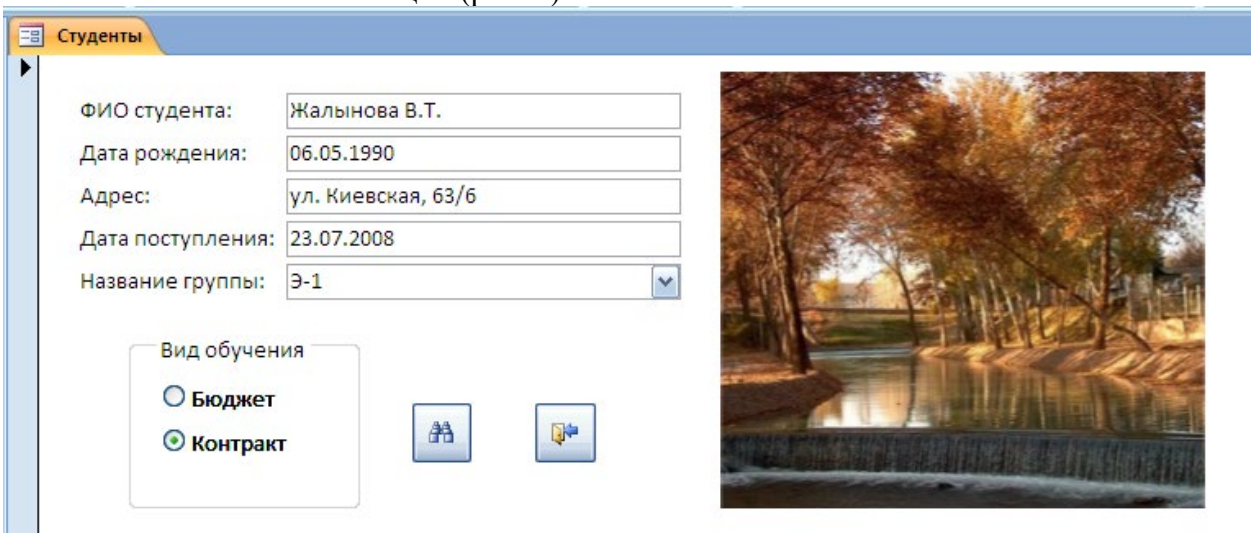



Рис. 12. Форма для таблицы «Студенты».

**Контрольные вопросы:**

1. Формы. Назначение форм.
2. Режимы форм.
3. Способы создания форм.
4. Мастер форм.
5. Конструктор форм.
6. Свойства формы.
7. Элементы управления.
8. Простые и сложные формы.
9. Главная и подчиненная форма.
10. Многотабличные формы.

**Задания к лабораторной работе:**

База данных «Учебный процесс».

1. Создать форму для просмотра и ввода информации в таблицу «кафедры».
2. Создать форму для просмотра и ввода информации в таблицу «группы».

3. Создать форму для ввода информации в таблицу «преподаватели» (для поля код кафедры (название кафедры) использовать элемент поле со списком).
4. Создать форму для ввода информации в таблицу «студенты» (для поля код группы (название группы) использовать элемент поле со списком, для поля код вида обучения - группу переключателей). (Рис 12)
5. Создать многотабличную форму для просмотра студентов по группам. (Рис 1)
6. Создать многотабличную форму для просмотра преподавателей по кафедрам.
7. Создать форму для ввода информации в таблицу «предметы» (для поля код преподавателя (ФИО преподавателя) использовать элемент поле со списком).
8. Создать форму для заполнения таблицы "Успеваемость", в которой должно быть: поле со списком для выбора студента; группа переключателей или поле со списком для выбора предмета; поле для ввода оценки; поле для вывода оценки прописью. поле для ввода даты.
9. Создать форму для заполнения таблицы "Оплата", в которой должно быть: поле со списком для выбора студента; поле даты (дата должна сниматься с таймера); поле для ввода суммы платежа; группа переключателей для выбора валюты.
10. Создать форму для отображения зачетной книжки. Форма содержит: ФИО студента, предметы, кол-во часов, ФИО преподавателя, оценки, оценки прописью, кнопки навигации. (Рис 9)
11. Создать форму позволяющую вывести студентов по группам, их успеваемость и статус, а также средний бал успеваемости по группам.
12. Создать форму, в которой отображается список студентов, оплативших за обучение по группам; а также сумма по каждому студенту и сумма по группам.
13. Создать форму о студентах, которая содержит 3 вкладки. Первая вкладка содержит личные данные. Вторая – оплата за обучение и сумма оплаты. Третья – успеваемость студента по предметам и среднюю успеваемость.
14. Создать форму об оплате с несколькими вкладками, каждая вкладка (наименование валюты) содержит список студентов и их сумму оплаты в соответствующей валюте.
15. Создать форму о студентах, которая содержит две вкладки. В зависимости от вида обучения на каждой вкладке размещается информация о студенте (ФИО, наименование группы, год рождения, адрес).

## Лабораторная работа №4. Создание отчетов.

Отчеты предназначаются для вывода информации из базы данных на принтер. Поэтому в отчетах данные форматируют так, чтобы их было удобно размещать на отдельных страницах. Отчеты поддерживают самые разнообразные способы оформления и позволяют группировать данные, разбивая их на логически цельные блоки.

Процесс создания отчета необходимо начинать с выбора полей, которые должны входить в отчет, а также определения таблиц или запросов, в которых эти поля находятся. Возможно, что при создании отчета вам могут потребоваться данные какой-нибудь одной таблицы, но чаще всего в нем должна содержаться информация из нескольких таблиц. В таком случае сначала должен быть создан запрос, который будет объединять в себе все нужные поля, взятые из разных таблиц.

В Access работа с отчетами может осуществляться в нескольких режимах: предварительный просмотр, представление отчета, режим конструктора и режим макета.

Чтобы правильно выбрать режим работы, необходимо четко определить, какие действия будут производиться над отчетом.

- Режим предварительного просмотра отображает отчет в том виде, в каком он будет напечатан, поэтому его целесообразно выбирать в том случае, когда отчет необходимо просмотреть перед отправкой на печать.
- Режим отчета является режимом просмотра отчетов, установленным по умолчанию. Этот режим удобен тем, что позволяет скопировать данные из отчета в буфер обмена и изменять состав выводимых для просмотра данных.
- Если необходимо внести изменения в отчет, редактировать данные либо изменить его структуру, то лучше использовать режим работы макета или режим конструктора.

**Структура отчета.**

В Access каждый отчет включает в себя несколько разделов. Каждый раздел имеет свое предназначение и используется только для определенных целей. В отчете выделяют семь основных разделов: заголовок отчета, верхний колонтитул или заголовок страницы, заголовок группы, область данных, примечание группы, нижний колонтитул или примечание страницы, примечание отчета. От правильности выбора раздела для расположения каждого из элементов управления зависит то, насколько грамотно будет составлен отчет, поэтому необходимо точно знать назначение каждого из разделов отчета.

- Заголовок отчета обычно содержит сведения, которые располагают на титульном листе или обложках, такие как эмблемы компании, название отчета или дата его создания. Заголовок отчета располагается над верхним колонтитулом и печатается всего один раз в самом начале отчета. Заголовок можно использовать для расчета каких-либо значений. Так, если в заголовке отчета расположен вычисляемый элемент управления, содержащий функцию Sum, подсчет значения будет осуществляться для всего отчета.
- Заголовок страницы располагается в верхней части каждой страницы отчета и используется в тех случаях, когда необходимо повторное отображение названия отчета на каждой его странице.
- Заголовок группы предназначен для вывода названия группы записей и располагается непосредственно перед ней. Каждой новой группе записей соответствует свой заголовок. В данном разделе также можно использовать вычисляемые элементы управления, в таком случае расчет будет производиться только для текущей группы.
- Область данных содержит элементы управления, из которых складывается основное содержание всего отчета. Этот раздел выводится только один раз для каждой строки данных.
- Примечание группы печатается в конце каждой группы записей и содержит сводную информацию по текущей группе.
- Примечание страницы выводится в нижней части каждой страницы и используется в том случае, если необходимо напечатать какие-либо постраничные сведения или номера страниц.
- Примечание отчета располагается в конце отчета и, как и его заголовок, печатается только один раз. Основным назначением данного раздела является вывод итогов по всему отчету или других сводных данных.

Такая последовательность расположения разделов соблюдается при создании отчета в режиме конструктора, однако при выводе отчета на печать, а также в режиме предварительного просмотра она немного изменяется. Примечание отчета, располагаемое в режиме конструктора после примечания страницы, при печати отчета будет отображаться перед ним, сразу после последнего примечания группы или после области данных последней страницы отчета.

В режиме конструктора можно добавить или удалить заголовок, а также примечание для отчета.

- Для добавления заголовка или примечания нужно вызвать контекстное меню щелчком правой кнопки мыши в окне отчета и выбрать команду «Заголовок/примечание отчета». Если в отчете уже созданы заголовок и примечание отчета, то при выборе команды «Заголовок/примечание отчета» эти разделы будут удалены.
- Добавить и удалить верхний и нижний колонтитулы можно, вызвав контекстное меню и выбрав команду «Колонтитулы страницы».
- Добавить заголовок для группы данных можно с помощью окна «Группировка, сортировка и итоги». Чтобы вызвать это окно, нужно перейти на вкладке «Конструктор» в группу «Группировка и итоги» и выбрать команду «Группировка».

Рассмотрим пример создания отчета в режиме конструктора.

Создадим отчет с одним уровнем группировки. Выведем список студентов по группам и подведем итоги, по каждой группировке, посчитав количество студентов в группе.

Создадим новую форму в режиме конструктора, для этого выполним команду «Конструктор отчетов» на вкладке «Создание». В режиме конструктора отчетов, с помощью контекстного меню, вызовем «Окно свойств» для отчета, на вкладке «Данные» для свойства «Источник записей» с помощью кнопки  вызовем построитель запросов, для формирования источника записей для отчета.

В построителе запросов добавим таблицу «Группы» и «Студенты». Из добавленных таблиц выберем поля: «Название группы», «ФИО студента», «Дата поступления», «Адрес». После выбора полей закрываем построитель запросов, при закрытии подтверждаем сохранение изменений и обновление свойства.

Далее необходимо добавить уровень группировки, для этого в режиме конструктора отчетов на вкладке «Конструктор» выбираем команду «Группировка». Группа представляет собой набор записей наряду с их вводными и итоговыми сведениями, например, с заголовком. Группа состоит из заголовка группы, вложенных групп (если таковые имеются), подробных записей и примечаний. Группировка позволяет наглядно разделить записи и отобразить вводные и итоговые данные для каждой из групп. После выбора команды в нижней части окна появиться панель «Группировка, сортировка и итоги». На этой панели необходимо выбрать команду «Добавить группировку». И в появившемся списке полей выбрать поле, в нашем случае, «Название группы» (рис. 1).

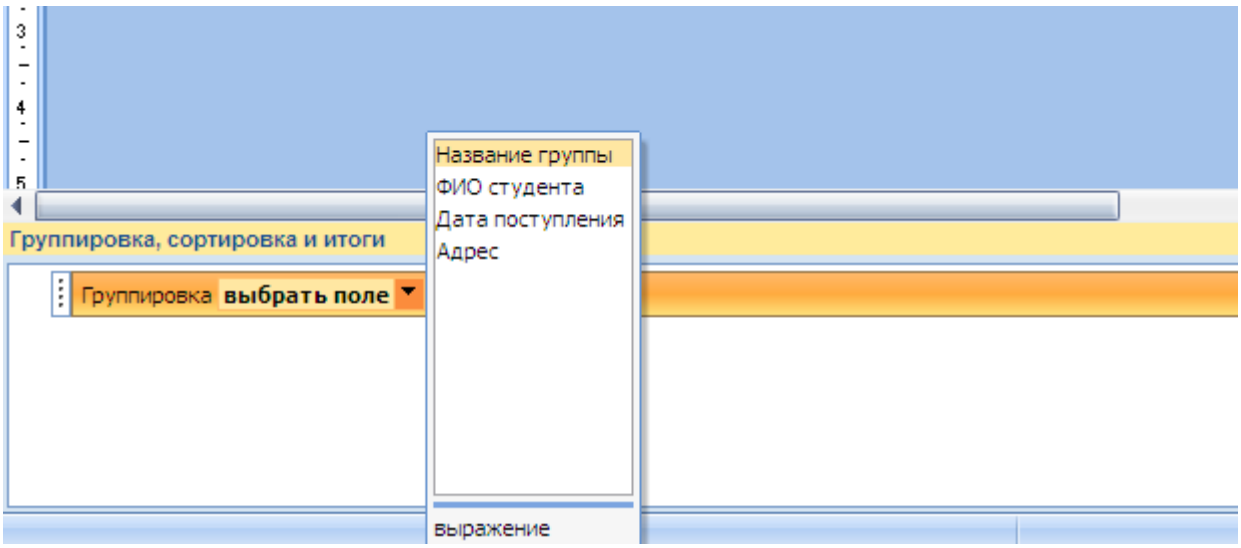



Рис. 1. Добавление группировки.

После добавления уровня группировки появляется «Заголовок группы». Далее на вкладке «Конструктор» в группе «Сервис» выбираем команду «Добавить поля». В правой части окна появиться список доступных полей. Выбираем поле «Название группы» и с помощью мыши переносим в область «Заголовок группы». А поля: «ФИО студента», «Дата поступления» и «Адрес» переносим в «Область данных». Далее размешаем элементы, таким образом, как показано на рис. 2.

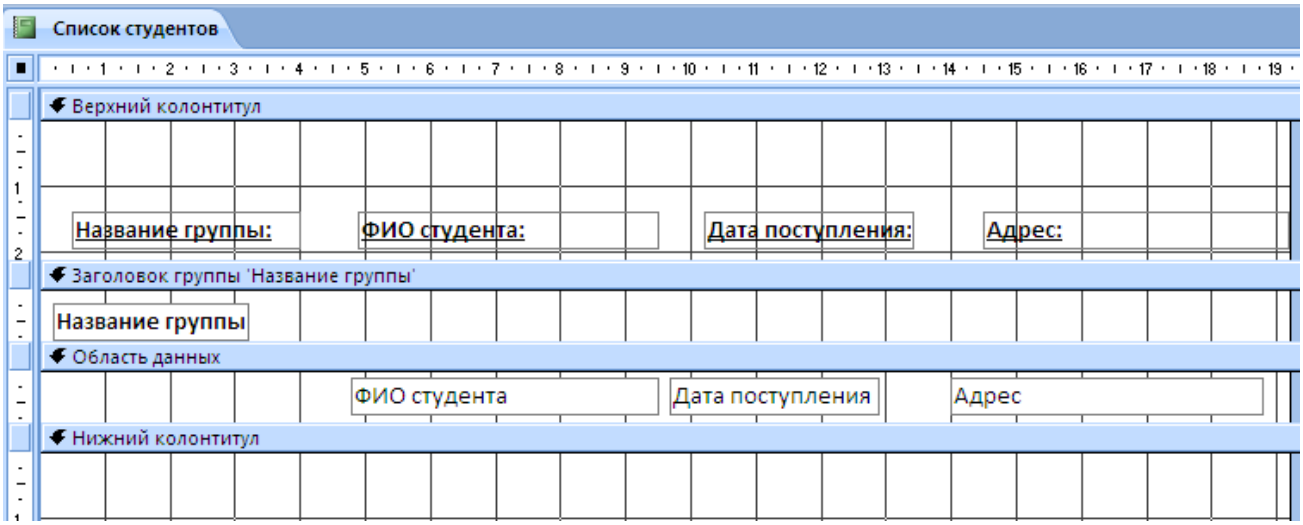



Рис. 2. Размещение элементов в отчете.

То есть сами поля оставляем в заголовке группы и в области данных, а надписи перемещаем в верхний колонтитул.

Для подведения итогов, в частности для нахождения количества студентов в каждой группе необходимо на панели «Группировка, сортировка и итоги» выбрать «Больше» (рис. 3). В результате появятся дополнительные параметры.

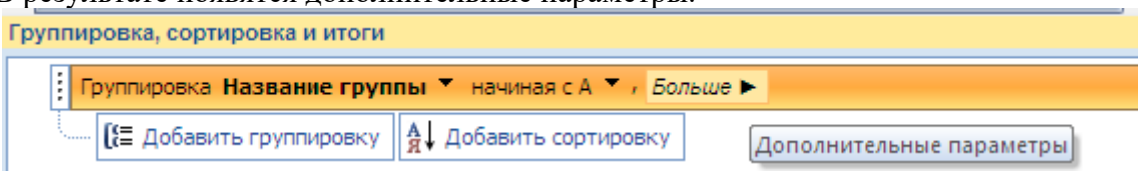



Рис. 3. Панель «Группировка, сортировка и итоги».

На появившейся панели дополнительных параметров открыть список «без итогов» и в раскрывшемся окне установить флажок «Показать в примечании группы» (рис. 4).

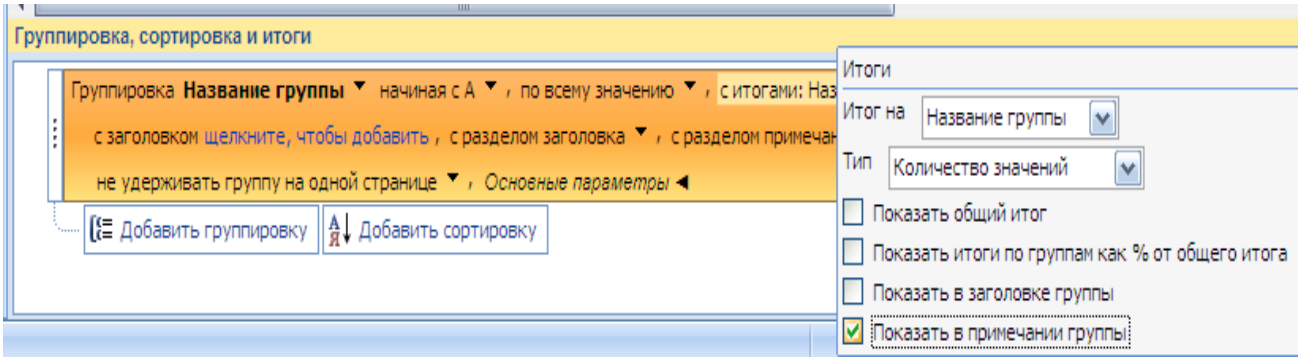



Рис. 4. Добавление итоговых полей.

В бланке отчета автоматически появиться «Примечание группы» с вычисляемым полем. Для пояснения добавляем надпись, используя элемент «Надпись».

Также кроме итогов в примечании группы, итоги можно автоматически добавлять в заголовок группы, в примечание всего отчета («Показать общий итог») и находить проценты от общего итога.

После добавления итоговых полей необходимо добавить заголовок для отчета. Для добавления заголовка или примечания нужно вызвать контекстное меню щелчком правой кнопки мыши в окне отчета и выбрать команду «Заголовок/примечание отчета». С помощью элемента «Надпись», добавим заголовок «Список студентов по группам».

Отчет готов, но для окончательного завершения добавим номера страниц и текущую дату. Для этого необходимо в нижний колонтитул добавить два элемента «Поле». Для первого поля в окне свойств поля, на вкладке «Данные», в свойстве «Данные» с помощью кнопки ... вызываем построитель выражений. В категории «Общие выражения» выбираем «Текущая дата и время» (рис. 5).

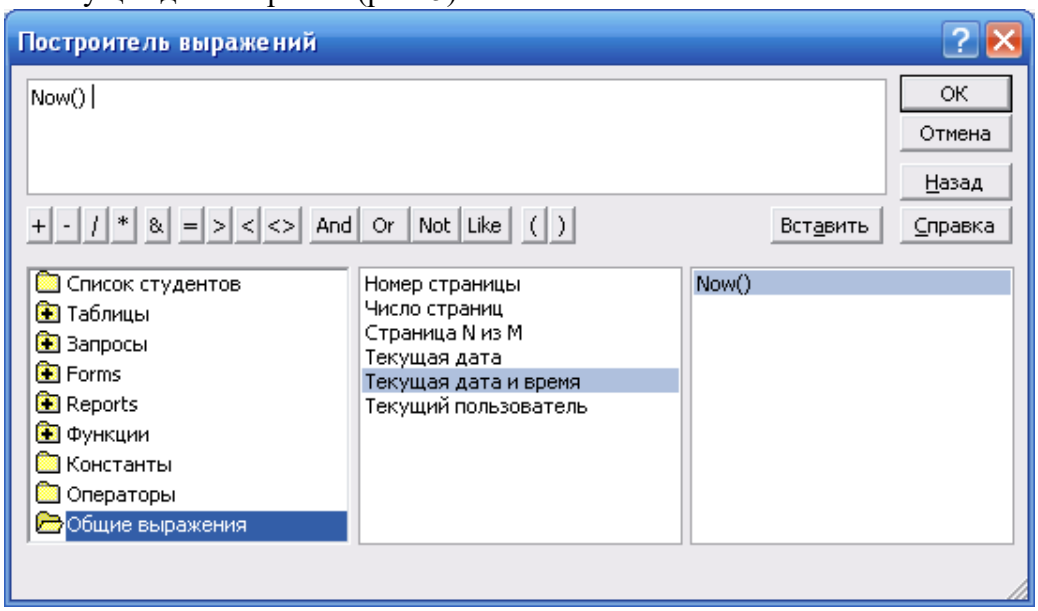



Рис. 5. Добавление текущей даты.

А для второго добавленного поля, также в окне свойств поля, на вкладке «Данные», в свойстве «Данные» с помощью кнопки ... вызываем построитель выражений. В категории «Общие выражения» выбираем «Страница N из M». Это выражение выглядит следующим образом: "Страница " & Page & " из " & Pages.

В результате получим готовый отчет, в режиме конструктора он будет выглядеть следующим образом (рис. 6):

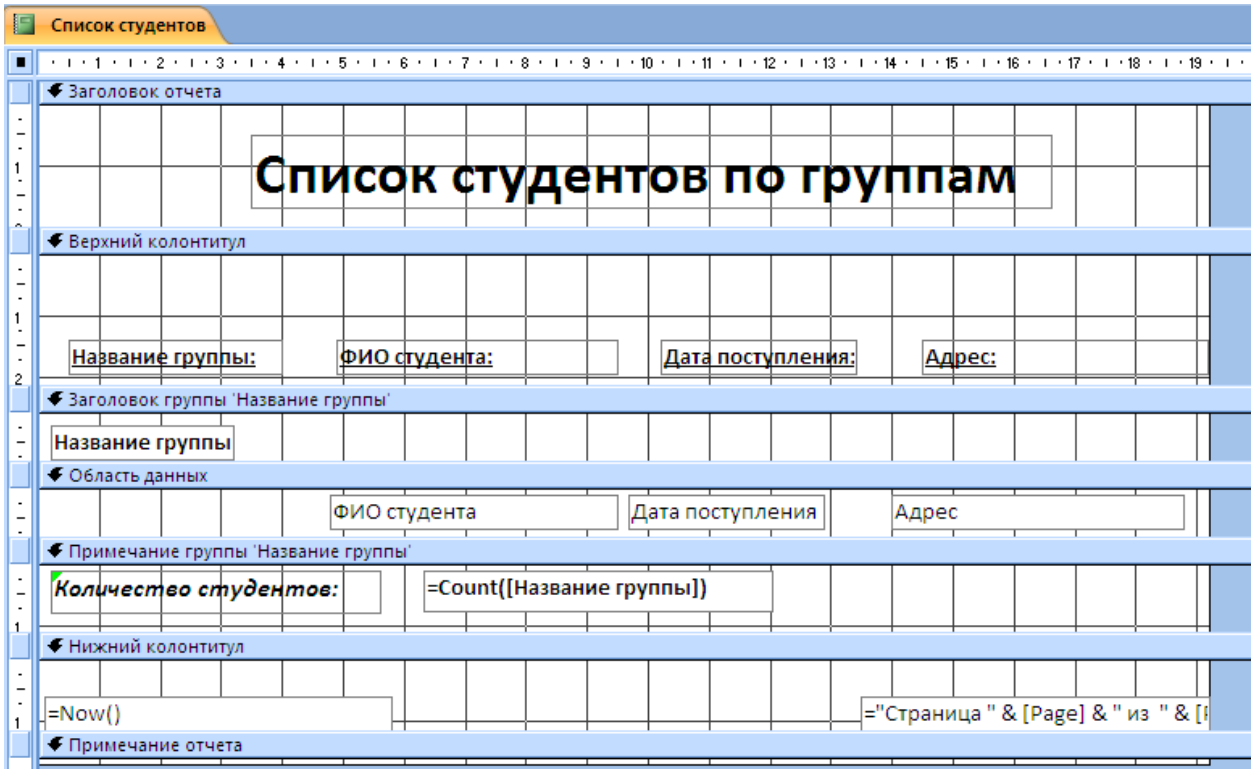



Рис. 6. Отчет в режиме конструктора.

**Контрольные вопросы:**

1. Отчеты. Назначение отчетов.
2. Разделы отчета.
3. Оформление отчета и свойства элементов управления.
4. Создание простых отчетов.
5. Группировка и сортировка данных.
6. Составные отчеты.
7. Внедрение объектов.
8. Вычисления в отчетах.

**Задания к лабораторной работе:**

База данных «Учебный процесс».

1. Создать отчет о преподавателях с группировкой по кафедрам.
2. Создать отчет о студентах по группам на базе параметрического запроса. Т.е. пользователь вводит название группы в окне параметров. Отчет содержит: Название группы (в заголовке отчета), ФИО студента и его данные.
3. Создать отчет об успеваемости студентов. Вывести ФИО студента, название предмета, оценку. Произвести группировку по ФИО студента и итоговое поле по оценке (посчитать среднюю успеваемость).
4. Создать отчет о внесении оплаты за обучение.
5. Создать отчет с группировкой по дате поступления, вывести ФИО студента. Добавить итоговое поле по количеству студентов.
6. Создать отчет с группировкой по названию группы, виду обучения, вывести студентов и их среднюю успеваемость. Подвести итоги и вывести среднюю успеваемость по виду обучения и по названию группы.
7. Создать отчет с двумя уровнями группировки (Название группы, ФИО студента). Подвести итоговое поле по общему количеству часов.

**Лабораторная работа №5. Разработка кнопочных форм.**

Главная кнопочная форма создается с целью навигации по базе данных, т.е. она может использоваться в качестве главного меню БД. Элементами главной кнопочной формы являются объекты форм и отчётов.

Чтобы быстро и легко создать кнопочные формы можно использовать диспетчер кнопочных форм

Создание кнопочной формы:

1. На вкладке «Работа с базами данных» в группе «Работа с базами данных» нажмите кнопку «Диспетчер кнопочных форм». (Если выводится приглашение «создать кнопочную форму», нажмите кнопку «Да».) Диспетчер кнопочных форм выводит на экран список страниц. Каждая страница — отдельная часть меню кнопочной формы. У самых простых кнопочных форм только одна страница, что означает поддержку формой одного уровня глубины и выполнение каждой кнопкой полезного действия (например, открытие формы или отчета).
2. Нажмите кнопку «Создать». На экране появится диалоговое окно «Создание» (рис. 1).

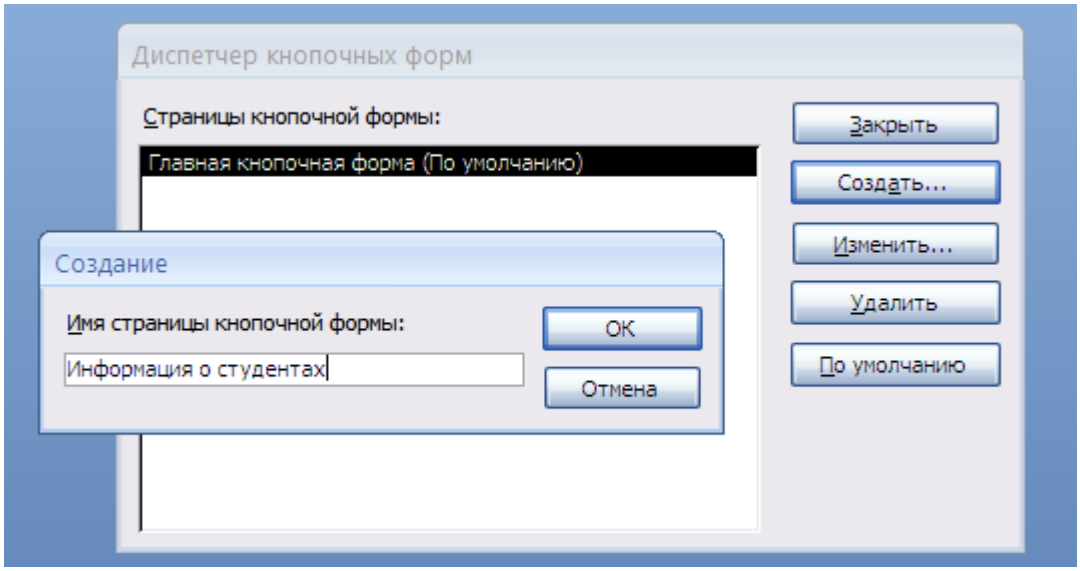



Рис. 1. Диспетчер кнопочных форм.

3. Введите имя новой кнопочной формы и нажмите кнопку «ОК». Имя новой кнопочной формы добавляется в список поля «Страницы кнопочной формы».
4. Выберите имя новой кнопочной формы и нажмите кнопку «Изменить». Будет открыто диалоговое окно «Изменение страницы кнопочной формы» (рис. 2). Окно «Изменение страницы кнопочной формы» позволяет создавать команды меню, удалять те, которые больше не нужны, изменять порядок их следования (этот порядок определяет порядок команд на кнопочной форме).

   Возможны следующие варианты:

   - Перейти к кнопочной форме — переходит на другую страницу кнопочной формы. Страницы кнопочной формы можно использовать для разбиения действительно длинных меню на несколько более коротких:
   - Открыть форму для добавления — открывает форму в режиме ввода данных, поэтому можно вставлять новые записи;
   - Открыть форму для изменения — открывает форму в обычном режиме для просмотра и редактирования записей;
   - Открыть отчет — открывает отчет в режиме предварительного просмотра;

- Конструктор приложения — открывает окно «Диспетчера кнопочных форм», таким образом, позволяя редактировать меню кнопочной формы. Этот вариант редко требуется включать в меню;
- Выйти из приложения — завершает программу Access;
- Выполнить макрос и Выполнить программу — запускают созданный вами макрос или написанный вами код на языке Visual Basic.

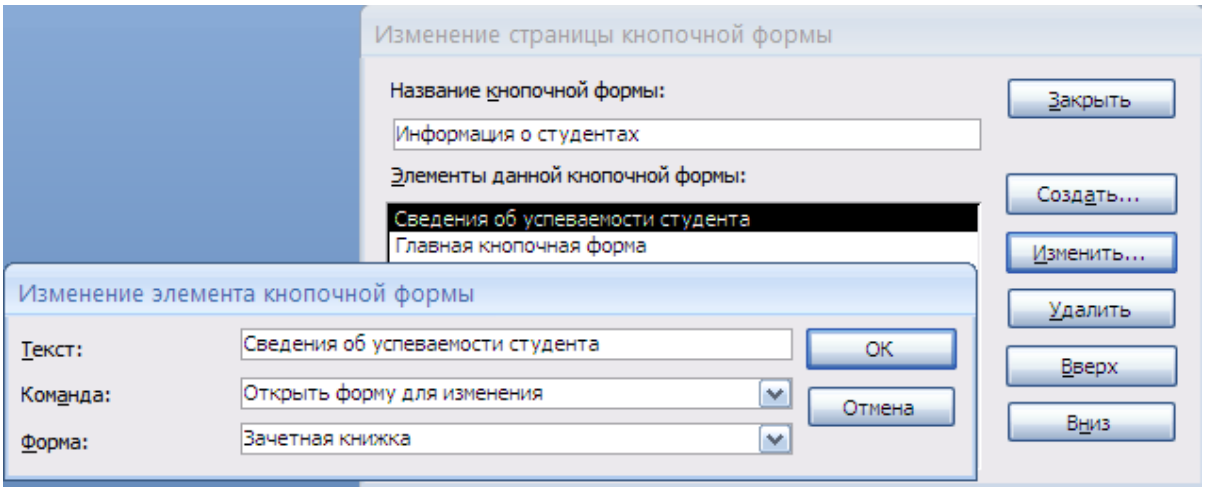



Рис. 2. Изменение страницы кнопочной формы.

5. Нажмите кнопку «Создать». Откройте диалоговое окно «Изменение элемента кнопочной формы» (рис. 2).
6. В поле «Текст» введите имя для первой кнопки кнопочной формы, а затем выберите для нее команду из списка «Команда». Например, введите текст «Сведения об успеваемости студента» в поле «Текст», а затем выберите «Открыть форму для добавления» из списка «Команда». Чтобы создать кнопочную форму, открывающую другие кнопочные формы (рис. 3), выберите в поле «Команда» команду «Перейти к кнопочной форме», а затем выберите кнопочную форму из списка или введите имя другой кнопочной формы.

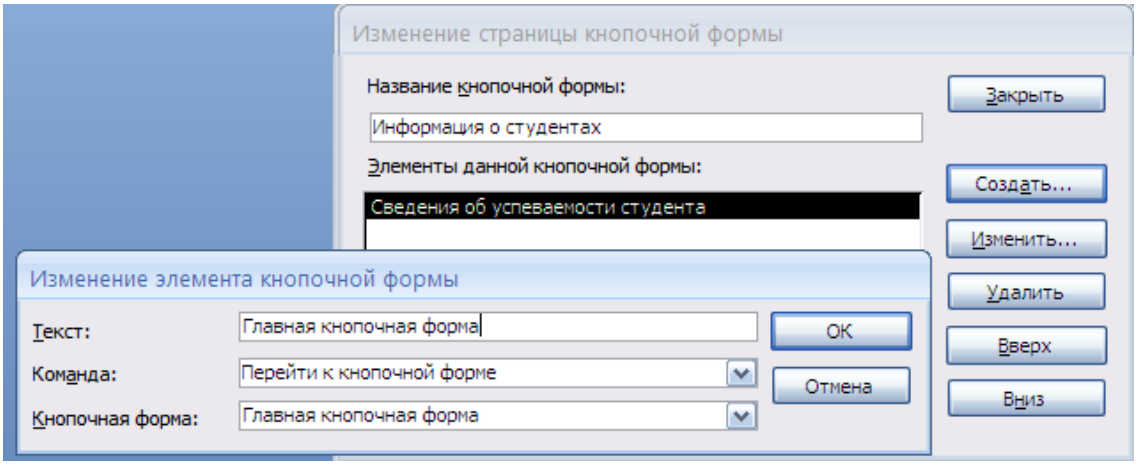



Рис. 3. Изменение элемента кнопочной формы.

7. В зависимости от выбранной команды в приложении Access может быть выведен еще один список под списком «Команда». Например, если выбрана команда «Открыть отчет», в открытой базе данных появляется список отчетов. Если требуется, выберите элемент из списка.

Повторяйте шаги 5 и 7 до тех пор, пока не будут добавлены все элементы кнопочной формы.

8. Для того чтобы осуществлялся переход ко всем новым созданным страницам кнопочной формы, необходимо на первой странице «Диспетчера кнопочных форм»

выбрать «Главная кнопочная форма» и нажать кнопку «Изменить». Далее в диалоговом окне «Изменение страницы кнопочной формы» нажимаем кнопку «Создать» и осуществляем переход к существующей подчиненной кнопочной форме (рис. 4).

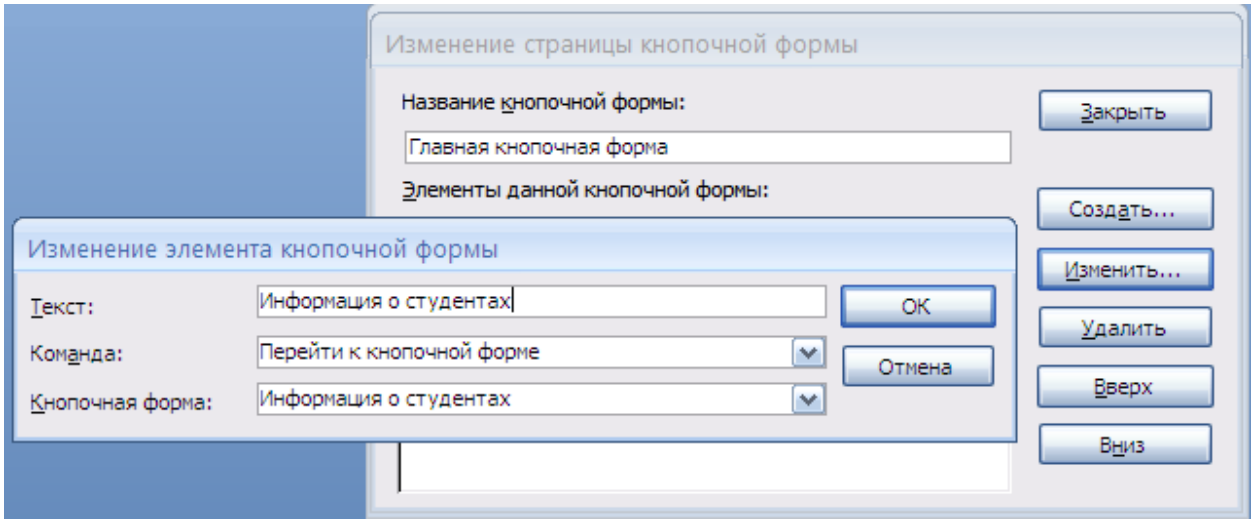



Рис. 4. Создание подчиненных кнопочных форм.

Чтобы завершить создание кнопочной формы, нажмите кнопку «Закрыть».

Таким образом, кнопочная форма — это форма, единственная цель которой перенаправить пользователя к другим формам (обычно при щелчке мышью кнопки формы). Она — своего рода главное меню базы данных. Такая форма — одновременно и отправная точка, и центр действий. Типичная кнопочная форма содержит набор кнопок, направляющих в разные места.

**Контрольные вопросы:**

1. Кнопочная форма.
2. Разделы кнопочной формы.

**Задания к лабораторной работе:**

1. Создать кнопочную форму. (База данных «Учебный процесс»)

## Лабораторная работа №6. Программирование в Access.

**Макросы.**

Макрос — это средство для автоматизации задач и добавления функциональных возможностей в формы, отчеты и элементы управления. Например, при добавлении командной кнопки в форму событие кнопки OnClick связывается с макросом, который содержит команды, выполняемые при каждом нажатии кнопки.

В приложении Access макросы можно рассматривать как упрощенный язык программирования, на котором программа записывается в виде списка макрокоманд. Основной компонент макроса; замкнутая инструкция, самостоятельно или в комбинации с другими макрокомандами определяющая выполняемые в макросе действия. В других макроязыках макрокоманды иногда называют просто командами.) для выполнения. При создании макроса каждая макрокоманда выбирается из раскрывающегося списка, после чего к ней добавляется необходимая информация. Макросы позволяют добавлять функциональные возможности в формы, отчеты и элементы управления без необходимости написания кода в модуле Макросы обеспечивают выполнение части

команд, доступных в VBA, и для большинства пользователей создание макроса оказывается проще, нежели написание кода VBA.

Предположим, требуется запустить отчет непосредственно из формы ввода данных. Можно добавить в форму кнопку и затем создать макрос, который будет запускать отчет. Макрос может быть изолированным (отдельный объект в базе данных), который затем связывается с событием OnClick для кнопки, или же он может быть внедрен непосредственно в событие кнопки OnClick . В любом случае, при нажатии кнопки выполняется макрос, который запускает отчет.

Рассмотрим пример создания простейшего макроса для открытия таблицы и перехода на последнюю запись. Выберите на ленте «Создание» в группе «Макросы и код» команду «Макрос». На экране появится новое окно для создания нового макроса (Конструктор макросов). Любой макрос состоит либо из последовательности одного или нескольких шагов, либо из действий (макрокоманд). Для создания макроса вы формируете список макрокоманд программы. У программы Access есть предварительно подготовленный список макрокоманд, которые можно использовать для приготовления макросов. Когда добавляется команда, ее просто выбирают из этого списка. Выберите макрокоманду «ОткрытьТаблицу». Далее устанавливаются параметры для вашего макроса.

Большинству макрокоманд для выполнения их работы нужна от вас некоторая информация. В макрокоманде «ОткрытьТаблицу» мало смысла до тех пор, пока вы не сообщите программе Access, какую именно таблицу хотите открыть. Эти дополнительные сведения и называют аргументами.

У макрокоманды «ОткрытьТаблицу» три аргумента.

- Имя таблицы ссылается на таблицу, которую вы хотите открыть. Его можно выбрать из раскрывающегося списка таблиц. В данном примере можно использовать любую непустую таблицу.
- Режим позволяет выбрать применяемый режим представления. Можно выбрать обычный режим «Таблица» для ввода информации, «Конструктор» для изменения структуры таблицы, «Просмотр» для подготовки данных к печати или «Сводную таблицу» и «Сводную диаграмму» для работы с итогами сводной таблицы. В данном примере выберите режим «Таблица».
- Режим данных определяет, какой тип изменений разрешен. Можно использовать стандартный вариант «Изменение» для разрешения любых изменений, «Только чтение» для запрета каких бы то ни было изменений или «Добавление» для разрешения только добавления записей. В данном примере выберите «Только чтение».

Вы можете вставить в макрос практически неограниченное число макрокоманд. Каждая команда занимает отдельную строку, и программа Access выполняет их по порядку, перемещаясь сверху вниз.

Для завершения данного примера добавьте макрокоманду «НаЗапись». Это действие обеспечивает получение нужной записи в таблице, которую вы только что открыли. При использовании аргументов следует указывать корректный объект. Затем можно использовать аргументы «Запись» и «Смещение» для точного задания той позиции, в которую вы хотите перейти. С помощью аргумента «Запись» можно выбрать переход к предыдущей строке, к следующей строке, к заполнителю новой строки в конце таблицы к заданной строке, к первой строке или, как в данном примере, к последней строке. Если выбрать «Конкретная», можно использовать аргумент «Смещение» для указания точной позиции, например, если задать «Смещение 5», обеспечивается переход к 5-й записи.

Сохраните созданный макрос (рис. 1).

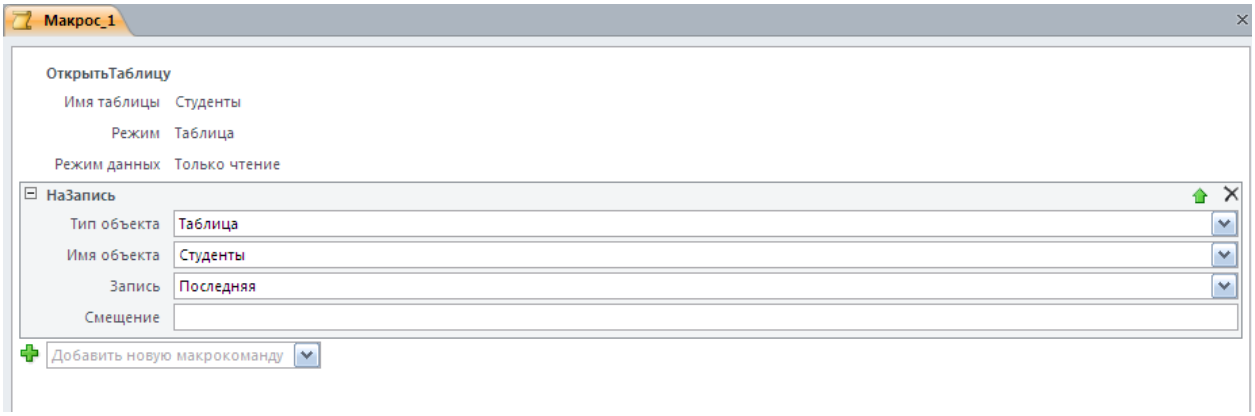



Рис. 1. Создание макроса.

Рассмотрим следующий пример. Создать форму со списком студентов, при нажатии на кнопку, возле фамилии студента, открывается форма с зачетной книжкой данного студента. То есть необходимо реализовать макрос, открывающий зачетную книжку студента.

Открываем форму «Студенты» в режиме конструктора. На панели элементов управления выбираем элемент «Кнопка» и размещаем элемент в области данных формы. Отменяем «Мастер создания кнопок». Далее необходимо вызвать окно свойств для созданной кнопки. В окне свойств, на вкладке «События» в строке «Нажатие кнопки» вызываем построитель с помощью кнопки . В появившемся окне необходимо выбрать «Макросы» и нажать кнопку «ОК». Появится конструктор макросов.

В конструкторе макросов выберите макрокоманду «ОткрытьФорму». Установите следующие аргументы макрокоманды:

- «Имя формы» – «Зачетная книжка» (Форма «Зачетная книжка» была создана в 3 лабораторной работе).
- «Режим» - «Форма».
- «Условие отбора» - [ФИО студента] =Forms![Студенты]![ФИО студента] (То есть необходимо указать по каким полям происходит связь между формами «Студенты» и «Зачетная книжка»). Если в форме «Зачетная книжка» отсутствует поле «ФИО студента», то такое поле обязательно следует добавить. Для этого необходимо открыть форму «Зачетная книжка» в режиме конструктора и в области данных под полем со списком добавить элемент «Поле». В окне свойств, для элемента «Поле», на вкладке «Данные» в свойстве «Данные» необходимо выбрать «ФИО студента».

Макрос готов (рис. 2). Сохраняем и закрываем макрос. Далее проверяем его работоспособность через форму «Студенты» нажав на кнопку.

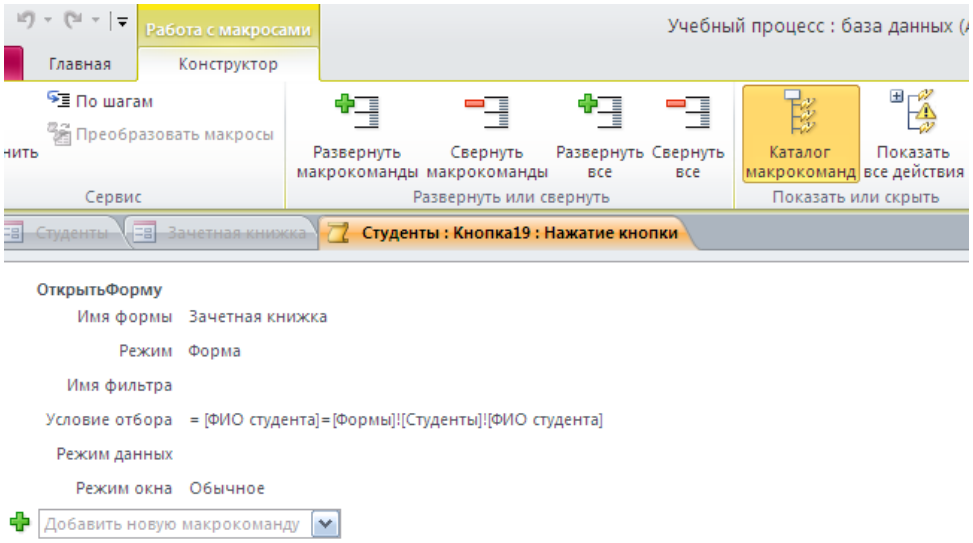



Рис. 2. Макрос для открытия формы.

**Макросы с условиями.**

Все рассмотренные макросы, до настоящего момента, были линейные. Они выполняли свои макрокоманды по порядку от начала к концу.

Для создания макроса с условием необходимо применить команду «Если». Для этого необходимо в «Каталоге макрокоманд», который обычно отображается в правой части окна, двойным кликом мыши выбрать команду «Если». В появившейся строке «Если» задаются условия. Условие похоже на выражение, но оно в результате всегда выдает одно из двух значений: «Истина» или «Ложь». Программа Access проверяет условие и решает с его помощью выполнить или нет соответствующую макрокоманду.

Рассмотрим пример, Допустим, вы хотите создать простое условие, которое проверяет добавление или обновление определенной записи. Например, в форме «Студенты» есть поле «Адрес», необходимо при событии «Потеря фокуса» для этого поля, чтобы появлялось сообщение «Вы не ввели адрес!», если поле осталось пустым.

Для этого в свойстве поля «Адрес», на вкладке «События» необходимо вызвать построитель для свойства «Потеря фокуса». Выбрать «Макросы».

В конструкторе макросов добавить строку «Если» и прописать условие IsNull([Формы]![Студенты]![Адрес]). Далее выбрать макрокоманду «ОкноСообщения» и указать соответствующие аргументы. Далее после окончания блока «Если» для установки фокуса можно выбрать макрокоманду «КЭлементуУправления» и в качестве аргумента указать «Адрес» (рис.3).

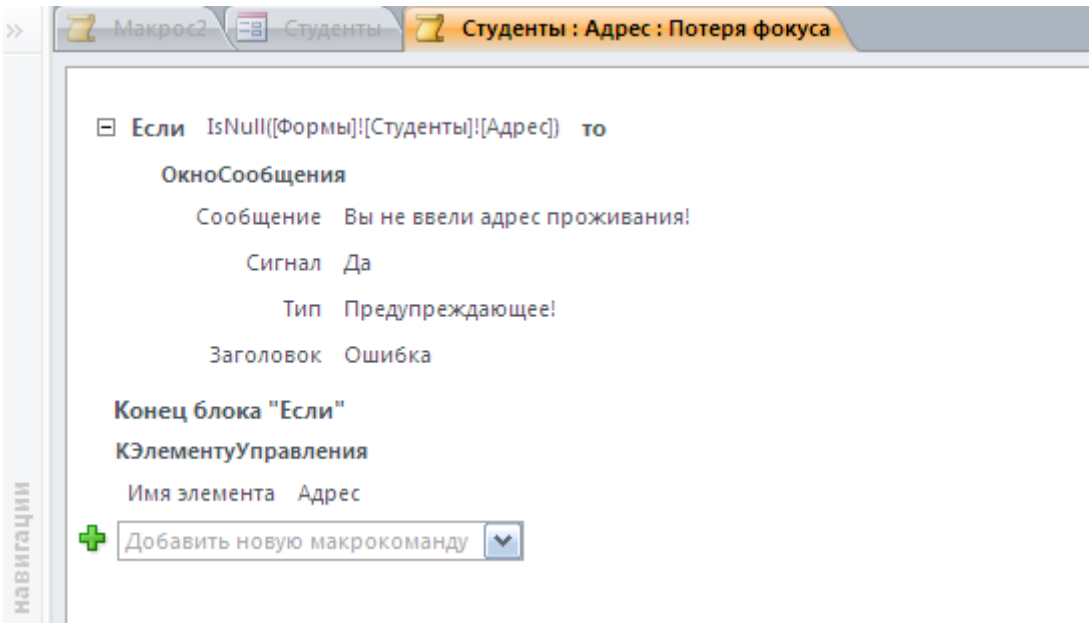



Рис. 3. Макрос с условием.

Макрос готов. Необходимо сохранить и закрыть его.

**Модули.**

В Access пользователь может управлять выполнением приложения, например, изменить информацию в таблице или щелкнуть на командной кнопке. Используя в приложении макросы и процедуры, можно реализовать реакцию пользователя на эти действия.

Для создания процедур обработки событий используется язык VBA (Visual Basic for Application). Язык VBA является современным языком структурного программирования. С помощью VBA можно настроить формы и отчеты, запустить макросы, а также отобразить объект Access в других приложениях или извлечь данные. Несмотря на то, что программный код на Visual Basic хранится в вашей БД, вы должны применять специальное средство для его - просмотра и редактирования. Оно называется редактором Visual Basic. Редактор Visual Basic работает во взаимодействии с программой Access, но открывается в отдельном окне. Для перехода в редактор Visual Basic выберите на ленте «Работа с базами данных» в группе «Макрос» команду «Visual Basic».

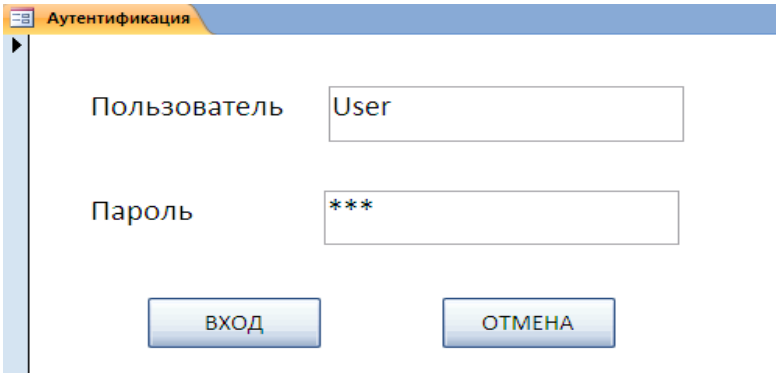
Для создания процедур VBA необходимы базовые знания по алгоритмизации и знания языка Visual Basic.

Рассмотрим пример. Необходимо реализовать окно аутентификации пользователя с использованием языка VBA. Пароли и имена хранятся в таблице базы данных.

Для начала создадим таблицу для хранения имен и паролей пользователей. И назовем ее «Пользователи». Таблица будет содержать два поля «Login» и «Password». Оба поля будут иметь тип данных «Текстовый». Введем в таблицу данные.

Далее создадим пустую форму. В режиме конструктора форм добавим два элемента «Поле» и два элемента «Кнопка». В свойствах элемента «Поле» на вкладке «Другие» в свойстве «Имя» назовем поля соответственно Login и Password. Кнопки переименуем на ВХОД и ОТМЕНА. В результате получим форму для аутентификации (рис. 4.).

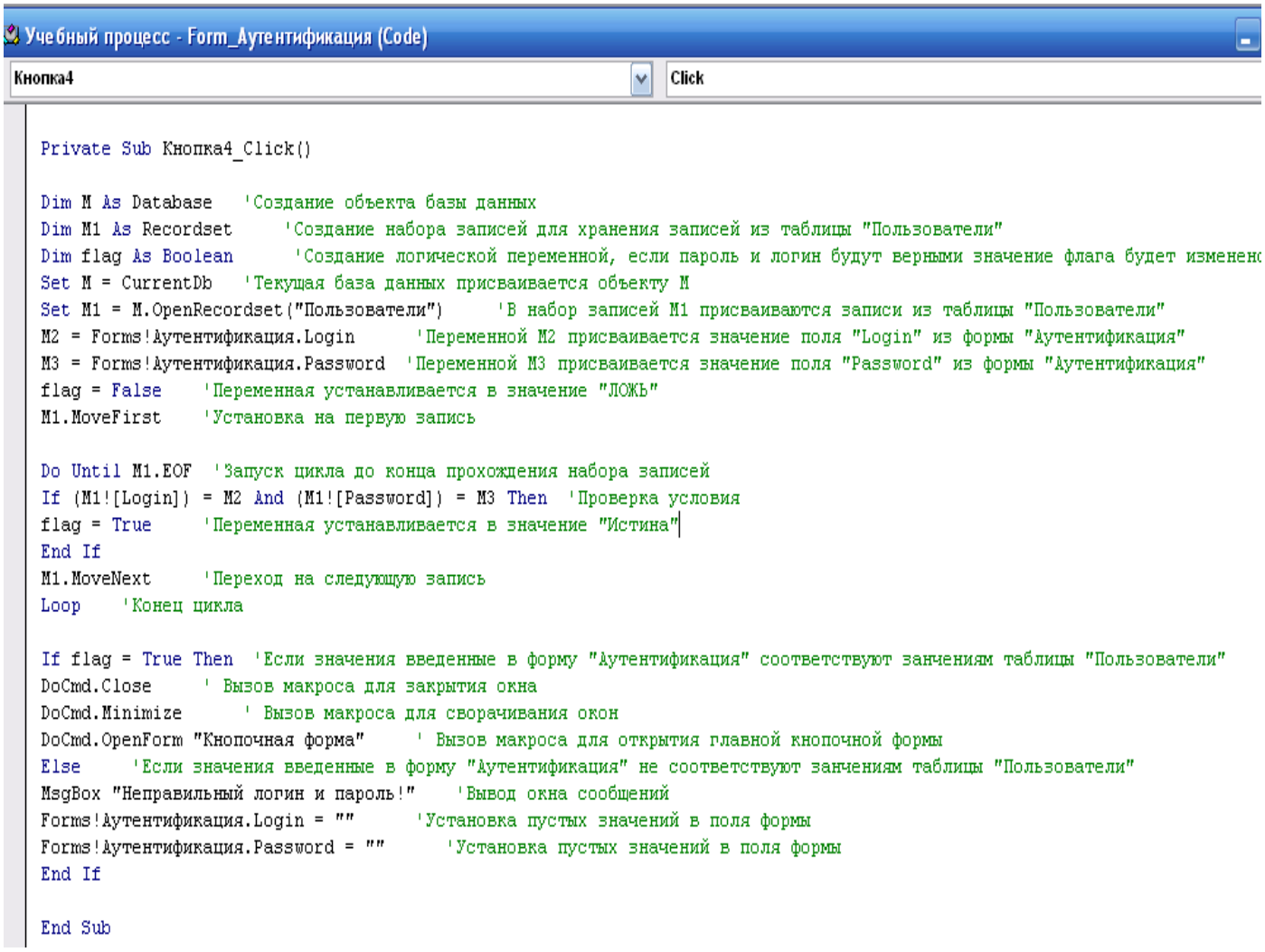
Рис. 4. Форма аутентификации пользователя.



В свойствах кнопки «ВХОД» на вкладке «События» для события «Нажатие кнопки» вызываем построитель и из предложенного списка действий выбираем «Программы». Запускается редактор VBA, с созданной процедурой обработки события для кнопки. В теле процедуры записываем программный код, следующего вида:

```
Dim M As Database   'Создание объекта базы данных
Dim M1 As Recordset      'Создание набора записей для хранения записей из таблицы "Пользователи"
Dim flag As Boolean      'Создание логической переменной, если пароль и логин будут верными, значение флага будет изменено
Set M = CurrentDb   'Текущая база данных присваивается объекту M
Set M1 = M.OpenRecordset("Пользователи")      'В набор записей M1 присваиваются записи из таблицы "Пользователи"
M2 = Forms!Аутентификация.Login      'Переменной M2 присваивается значение поля "Login" из формы "Аутентификация"
M3 = Forms!Аутентификация.Password  'Переменной M3 присваивается значение поля "Password" из формы "Аутентификация"
flag = False    'Переменная устанавливается в значение "ЛОЖЬ"
M1.MoveFirst    'Установка на первую запись
Do Until M1.EOF  'Запуск цикла до конца прохождения набора записей
If (M1![Login]) = M2 And (M1![Password]) = M3 Then  'Проверка условия
flag = True     'Переменная устанавливается в значение "Истина"
End If
M1.MoveNext     'Переход на следующую запись
Loop    'Конец цикла
If flag = True Then     'Если значения введенные в форму "Аутентификация" соответствуют занчениям таблицы "Пользователи"
DoCmd.Close     ' Вызов макроса для закрытия окна
DoCmd.Minimize      ' Вызов макроса для сворачивания окон
DoCmd.OpenForm "Кнопочная форма"        ' Вызов макроса для открытия главной кнопочной формы
Else        'Если значения введенные в форму "Аутентификация" не соответствуют занчениям таблицы "Пользователи"
MsgBox "Неправильный логин и пароль!"    'Вывод окна сообщений
Forms!Аутентификация.Login = ""      'Установка пустых значений в поля формы
Forms!Аутентификация.Password = ""      'Установка пустых значений в поля формы
End If
```

Рис. 5. Листинг программы.




Обработчик для кнопки «ВХОД» готов, закрываем редактор VBA.

В конструкторе формы «Аутентификация» для кнопки «ОТМЕНА», в свойстве «Нажатие кнопки» на вкладке «События» вызываем построитель и выбираем «Макросы». В конструкторе макросов выбираем макрокоманду «Выход». Сохраняем и закрываем конструктор. Таким образом, при нажатии на кнопку «ОТМЕНА» будет полностью закрываться приложение.

Форма «Аутентификации» готова, но для того чтобы данная форма запускалась сразу при открытии базы данных необходимо создать специальный макрос автозапуска.

Для этого создадим макрос. Добавим макрокоманды «Свернуть» и «ОткрытьФорму». Для макрокоманды «ОткрытьФорму» в аргументах макрокоманды в аргументе «Имя формы» необходимо указать форму «Аутентификация» (рис. 6). Для того чтобы макрос запускался автоматически необходимо назвать макрос «AutoExec». При открытии

приложения программа автоматически запускает макрос с названием «AutoExec».

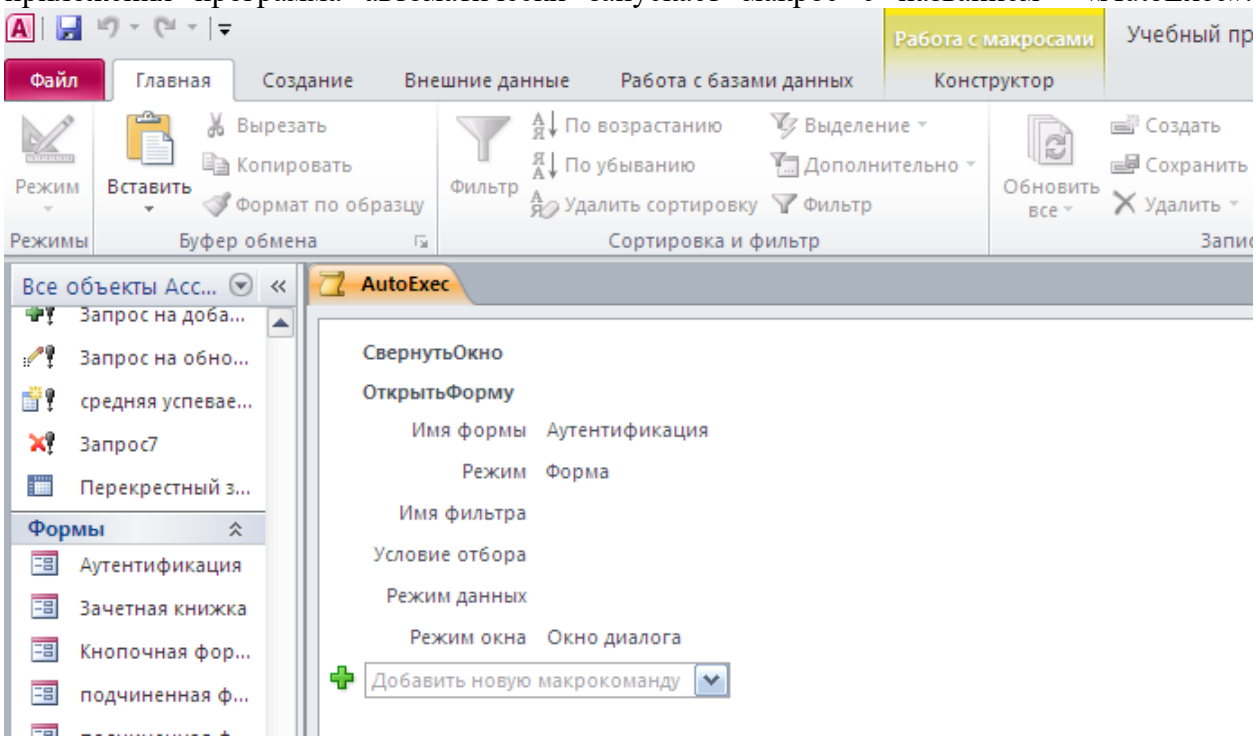



Рис. 6. Макрос «AutoExec».

**Контрольные вопросы:**

1. Макрос. Назначения макроса.
2. Основные макрокоманды.
3. Конструктор макросов.
4. Условия выполнения макросов.
5. Группы макросов.
6. Запуск макроса.
7. Связывание макроса с событием.
8. Модуль. Назначение модулей.
9. Модули класса.
10. Стандартные модули.
11. Преобразование макросов в модули.
12. Элементы программы.
13. Переменные, константы, массивы.
14. Процедуры и функции.
15. Управление выполнением программы.
16. Семейства, объекты, методы, свойства.
17. Иерархическая структура объектов.
18. Взаимосвязь макросов и программ VBA.
19. События. Виды событий.

**Задания к лабораторной работе:**

База данных «Учебный процесс».

1. Создать пять макросов.
2. Реализовать все способы запуска макроса.
3. Реализовать окно аутентификации пользователя, с использованием макросов (для одного имени и пароля).
4. Создать форму со списком студентов, при нажатии на кнопку, возле фамилии студента, открывается форма с зачетной книжкой данного студента. (Рис 2)

5. Создать форму со списком преподавателей, при нажатии на кнопку возле фамилии преподавателя открывается форма с информацией о предметах, которые данный преподаватель преподает.
6. Создать форму с наименованием групп, при нажатии на кнопку возле названия группы открывается форма с информацией о студентах, обучающихся в выбранной группе.
7. Реализовать окно аутентификации пользователя с использованием языка VBA. Пароли и имена хранятся в таблице базы данных.

1. **Расширяемость: Возможность создания функций и типов данных.**
2. **Поддержка SQL: PostgreSQL полностью совместим с языком SQL и имеет богатый набор операторов и функций.**
3. **Транзакционность : Поддержка транзакций с использованием ACID-свойств (Атомарность, Согласованность, Изолированность, Устойчивость).**
4. **Многопользовательская поддержка: Возможность одновременного доступа нескольких пользователей к базе данных.**
5. **Гибкость и расширяемость  PostgreSQL поддерживает хранимые процедуры, триггеры, функции и множество расширений.**
6. **Открытый исходный код: PostgreSQL распространяется под лицензией PostgreSQL, что позволяет свободно использовать, изменять и распространять его.**

**Приложение 3**

## Исследовательский проект по курсу Базы данных

Студентам необходимо самостоятельно выполнить один из проектов, описанных ниже.

***Требования к выполнению проекта и представлению результатов***

Основными частями проектной работы являются:

1. База данных. Это основная часть проекта. Исходя из описания предметной области и требований к базе данных, должны быть последовательно разработаны концептуальная, логическая и физическая модель базы данных. Затем проект базы данных реализуется в виде приложения *MS Access*.
2. Электронные таблицы. Для решения ряда задач необходимо провести дополнительный анализ данных с помощью сводных таблиц, расчетов трендов и других возможностей *MS Excel.*
3. Компьютерная презентация. Необходимо представить обзор проекта в виде презентации *MS Power Point*.
4. Отчет по проекту. Проект должен быть описан в отчете в виде документа *MS Word*.

***Требования к базе данных***

База данных должна быть создана в *MS Access* в соответствии с проектом структуры базы данных. Должны быть заданы первичные ключи, установлена ссылочная целостность базы данных.

Все данные должны вводиться через экранные формы, объединенные главной кнопочной формой.

Должны быть построены следующие виды запросов:

- запросы на выборку, включая параметрические, перекрестные;
- запросы на выборку с группировкой и с вычисляемыми полями;
- созданы итоговые и текущие отчеты с группировкой данных по таблицам и запросам.

***Требования к анализу данных в Excel***

Некоторые таблицы и результаты запросов следует экспортировать в файл *Excel* для дополнительного анализа. В качестве примера можно привести таблицу-результат запроса обо всех заказах за прошлый год вместе с информацией по поставщикам, покупателям и сотрудникам.

По экспортированным в *Excel* таблицам можно построить сводные таблицы, таблицы с промежуточными итогами и различные диаграммы.

Могут быть проведены дополнительные статистические расчеты, например, частота заказов в различные города, прогнозы продаж, линейные тренды продаж и т. п.

***Требования к презентации***

В структуре презентации должны содержаться следующие слайды:

1. Титульный слайд презентации.
2. Обзор базы данных.
3. Проект базы данных.
4. Концептуальная модель.
5. Логическая модель.
6. Структура таблиц в *MS Access*.
7. Первичные ключи.
8. Схема данных.
9. Получение информации.
10. Запросы на выборку.
11. Запросы с группировкой.
12. Параметрические запросы.
13. Перекрестные запросы.

14. Запросы с вычисляемыми полями.
15. Экранные формы.
16. Стандартные отчеты.
17. Анализ данных.

***Требования к отчету***

1. Отчет следует представить как в виде *Word* файла (15 – 25 стр.).
2. Основная часть отчета:

   ***Титульный лист***
   ***Содержание***
   ***Введение***
   ***Проект базы данных***
   - Концептуальная модель;
   - Логическая модель;
   - Физическая модель.

   ***Реализация в MS Access***
   - Структура таблиц, ключи;
   - Связи между таблицами;
   - Основные запросы;
   - Стандартные отчеты;
   - Описание пользовательского интерфейса (формы).

   ***Анализ данных***
   - Примеры анализа в Excel;
   - Диаграммы.

   ***Выводы***
   ***Руководство пользователя***
   - Как запустить приложение;
   - Назначение основных форм;
3. Содержание титульного листа:

   Учебное заведение, факультет
   Курс " "
   Название проекта («Разработка базы данных …»)
   Автор, группа:
   Руководитель:
   Бишкек 201..

Все заголовки разделов и подразделов должны быть выполнены стилями заголовков. На каждой странице, кроме первой, должны присутствовать колонтитулы с названием работы и фамилией автора.

Поля: верхнее, нижнее – 2 см, правое – 3 см, левое – 1,5 см, шрифт Times New Roman, размер шрифта – 14, интервал – 1,5, абзац – 1,25, выравнивание по ширине. Объем 15-25листов. Графики, рисунки, таблицы обязательно подписываются (графики и рисунки снизу, таблицы сверху).

***Варианты заданий:***

1. Разработка базы данных «Салон красоты».
2. Разработка базы данных «Автошкола».
3. Разработка базы данных «Агентство недвижимости».
4. Разработка базы данных «Империя кадров».
5. Разработка базы данных «Аптека».
6. Разработка базы данных «Абитуриент».
7. Разработка базы данных «Спортивный клуб».
8. Разработка базы данных «Кулинарные рецепты».
9. Разработка базы данных «Стоматологическая поликлиника».
10. Разработка базы данных «Музыкальная школа».
11. Разработка базы данных «Склад бытовой техники».
12. Разработка базы данных «Звезды кино».
13. Разработка базы данных «Адвокатская контора».
14. Разработка базы данных «Туристическая фирма».
15. Разработка базы данных «Пансионат».
16. Разработка базы данных «Отдел кадров».
17. Разработка базы данных «Магазин ювелирных изделий».
18. Разработка базы данных «Энергосбыт».
19. Разработка базы данных «Электронная библиотека».
20. Разработка базы данных «Горнолыжная база».

*Вариант задания соответствует порядковому номеру студента в списке группового журнала.*

*vv*

**Приложение 4**

## Тесты
## по курсу Базы данных

1. Каким образом осуществляется переход к столбцу в реляционной базе данных?
    - по имени
    - по номеру
    - по индексу
2. Как называются столбцы, значения в которых изначально должны быть различными для разных строк этой таблицы?
    - первичный ключ
    - внешний ключ
    - индексный ключ
3. Как называются столбцы, которые не являются первичным ключом таблицы, но представляют первичный ключ другой таблицы?
    - внешний ключ
    - вторичный ключ
    - индексный ключ
4. Какая модель базы данных определяется, как набор таблиц с уникальными именами?
    - реляционная модель
    - объектная модель
    - иерархическая модель
5. Укажите тип данных в Access, который должен содержать уникальное значение.
    - счетчик
    - поле MEMO
    - поле OLE
6. Укажите тип данных в Access, который может содержать данные нескольких типов.
    - поле OLE
    - поле MEMO
    - денежный
7. Укажите тип данных в Access, который может содержать текст более чем 255 символов.
    - поле MEMO
    - поле OLE
    - текстовый
8. Какое свойство поля, позволяет автоматизировать проверку ввода символов в поле?
    - маска ввода
    - индексированное поле
    - условие на значение
9. Какое значение устанавливается в свойстве «индексированное поле», если столбец является первичным ключом?
    - да (совпадения не допускаются)
    - да (совпадения допускаются)
    - нет
10. Какое поле в таблице содержит однозначный идентификатор для каждой записи?
    - первичный ключ

- индексированный ключ
- идентификационный ключ

11. Какой символ шаблона, при использовании маски ввода, означает любое число произвольных символов?
    - *
    - ?
    - #
12. Как необходимо указать диапазон символов, при использовании маски ввода?
    - в квадратных скобках [A-P]
    - в фигурных скобках {A-P}
    - в круглых скобках (A-P)
13. Сколько частей в маске ввода?
    - три
    - две
    - четыре
14. Что такое «связь между таблицами»?
    - определение типа отношения между таблицами
    - определение индексов в нескольких таблицах
    - передача данных из одной таблицы в другую
15. Что означает «каскадное обновление связанных полей»?
    - это означает, что при изменении ключевого поля главной таблицы автоматически изменяются соответствующие значения поля в связанных записях
    - это означает, что при изменении ключевого поля в подчиненной таблице автоматически изменяются соответствующие значения в главной таблице
    - это означает, что при изменении данных в главной таблице не происходит никаких изменений в подчиненных таблицах
16. При каком типе отношения между таблицами, одной записи главной таблицы соответствует несколько записей связанной таблицы?
    - один ко многим
    - один к одному
    - многие ко многим
17. Как называется тип запроса, когда исходные записи, имеющие одинаковое значение во всех указанных полях, будут представлены как одна запись результирующего набора?
    - запрос с группировкой
    - параметрический запрос
    - перекрестный запрос
18. Как называется запрос, который представляет собой варианты базового запроса и отличается только разными значениями условия отбора?
    - параметрический запрос
    - перекрестный запрос
    - запрос на выборку
19. С помощью какого типа запроса можно компактно отобразить выбранные данные и объединить однородную информацию?
    - перекрестный запрос
    - параметрический запрос
    - запрос на выборку

20. К какому типу ошибок, при выполнении запросов действий, относятся ошибки появляющиеся при попытке сделать изменения в таблице открытой другими пользователями?
    - заблокированные записи
    - ошибки преобразования данных
    - нарушение условий на значение
21. Как называется форма, которая встраивается в другую форму?
    - подчиненная форма
    - вложенная форма
    - встроенная форма
22. Можно ли в форме запретить изменения значений в каком-либо поле?
    - да, если в режиме конструктора формы для нужного элемента вызвать окно Свойств и установить для свойства Блокировка (Locked) значение Да (Yes)
    - да, если в режиме конструктора формы выделить нужный элемент и выполнить команду Формат | На задний план
    - нет, этого сделать нельзя
23. Вы хотите, чтобы при открытии базы данных на экране автоматически отображалась заданная форма. Можно ли это сделать и как, если можно?
    - да, если дать команду Сервис | Параметры запуска (Tools | Startup)
    - да, если дать команду Файл | Свойства базы данных (File | Properties of Database)
    - да, если дать команду Сервис | Параметры (Tools | Options)
24. Между двумя таблицами в базе данных установлена связь «один-ко-многим», при этом включен режим "Обеспечение целостности данных", но выключен режим "Каскадное удаление". Что случиться, если Вы удаляете из главной таблицы запись, которая имеет связанные записи в подчиненной таблице?
    - MS Access не позволит выполнить удаление
    - MS Access попросит пользователя подтвердить удаление
    - MS Access удалит связь между таблицами
25. Могут ли дублироваться значения в ключевом поле (primary key) таблицы?
    - нет, никогда
    - да, если это поле будет использовано для установления связи с другими таблицами
    - да, если ключевое поле состоит более чем одного поля таблицы
26. Вы поместили в форму командную кнопку для запуска макроса. Как нужно модифицировать свойства кнопки, чтобы она запускала макрос?
    - для свойства Нажатие кнопки (OnClick) указать имя макроса
    - для свойства До обновления (BeforeUpdate) указать имя макроса
    - имя кнопки должно быть таким же, как имя макроса
27. Можно ли создать запрос, который при выполнении отображает диалоговое окно для ввода параметров условия отбора?
    - да, если создать запрос с параметрами
    - нет, этого сделать нельзя
    - да, если создать запрос с изменениями
28. Вы удалили в таблице запись. Можно ли восстановить удаленную запись в таблице и как?
    - нет, этого сделать нельзя
    - да, если сразу после удаления записи нажать клавишу Esc
    - да, если сразу после удаления записи нажать кнопку Восстановить запись (Undo)

29. Какое свойство поля таблицы должно использоваться, чтобы при добавлении новой записи в таблицу в ячейке автоматически устанавливалось заданное значение?
    - значение по умолчанию (Default Value)
    - обязательное поле (Required)
    - условие на значение (Validation Rule)
30. Что можно использовать в качестве источника данных для отчета?
    - комбинацию таблиц и запросов
    - запросы
    - таблицы
31. Что должно входить в структуру СУБД?
    - процессор языка запросов и командный интерфейс
    - визуальная оболочка
    - система помощи
32. Что такое однопользовательская система (single-user system)?
    - система, с которой одновременно может работать только один человек
    - система с высоким уровнем защиты, с которой может работать только один авторизованный человек
    - сетевая система, которая не различает своих пользователей по именам
33. Какое из определений более точно отражает смысл термина "независимость данных"?
    - пользователь не зависит от физической структуры хранения данных
    - пользователь не может редактировать данные - они не зависят от пользователя
    - данные между собой независимы, то есть не пересекаются
34. Что понимают под термином "общие данные"?
    - возможность использования отдельных областей данных, несколькими разными пользователями
    - данные, содержащиеся в нескольких таблицах одновременно
    - данные, не содержащие конкретных значений параметров
35. Возможно ли, при помощи реляционной модели представления данных описать иерархическую структуру?
    - да, возможно
    - нет, невозможно
    - зависит от решаемой задачи
36. Что решается на этапе логического проектирования БД?
    - что хранить
    - как хранить
    - в какой последовательности запоминать
37. Что такое физическое проектирование БД?
    - определение внутренней схемы
    - определение отображения «внешний - концептуальный»
    - разметка диска для файла базы данных
38. Какова основная роль СУБД?
    - обеспечить выполнение операций над хранимой БД
    - обеспечить оптимальную обработку запросов пользователя
    - обеспечить безопасность и целостность БД
39. Какое из перечисленных свойств не является свойством реляционной базы?
    - несколько узлов уровня связаны с узлом одного уровня
    - порядок следования строк в таблице произвольный

- каждый столбец имеет уникальное имя

40. Какая база данных строится на основе таблиц и только таблиц?
    - реляционная
    - иерархическая
    - сетевая
41. В какой модели баз данных существуют горизонтальные и вертикальные связи между элементами?
    - сетевой
    - иерархической
    - реляционной
42. Какой из ниже перечисленных запросов нельзя построить?
    - параллельный
    - перекрестный
    - простой
43. В чем заключается функция ключевого поля?
    - однозначно определять запись
    - однозначно определять таблицу
    - определять заголовок столбца таблицы
44. Из чего состоит макрос?
    - из набора макрокоманд
    - из набора гиперссылок
    - из совокупности операторов Visual Basic
45. Какого раздела не существует в конструкторе форм?
    - итоговый раздел
    - верхний колонтитул
    - раздел заголовка
46. Что можно разместить в поле OLE СУБД Access?
    - файл
    - число
    - калькулятор
47. Какие типы полей записей в СУБД Access допустимы?
    - логический , дата, числовой, денежный, OLE
    - числовой, текстовый, гипертекстовый, логический
    - числовой, символьный, графический музыкальный
48. Что такое макрос в СУБД Access?
    - объект, представляющий собой структурированное описание одного или нескольких действий
    - часть командного процессора
    - язык программирования
49. Как называется структура данных, для которой характерна подчиненность объектов нижнего уровня объектам верхнего уровня?
    - иерархической
    - реляционной
    - табличной
50. Что не относится  к основным функциям СУБД?
    - хранение данных
    - определение данных
    - управление данными

51. Какой командой создается база данных в PostgreSQL?
    - CREATE DATABASE mydb;
    - NEW DATABASE mydb;
    - ADD DATABASE mydb;
    - MAKE DATABASE mydb;

52. Какая команда используется для удаления базы данных?
    - DROP DATABASE mydb;
    - DELETE DATABASE mydb;
    - REMOVE DATABASE mydb;
    - ERASE DATABASE mydb;

53. Какой тип данных используется для хранения больших текстов?
    - TEXT
    - VARCHAR(255)
    - CHAR(255)
    - STRING

**Приложение 5**

**Виды работ и шкалы оценок**
**по курсу Базы данных**

**Лабораторная/Практическая работа**

Лабораторная работа — один из видов практических работ, реализуемых кафедрой ЭММ.

Целью лабораторной работы является углубление и закрепление теоретических знаний через развитие навыков обработки данных для решения поставленной задачи в присутствии и под руководством преподавателя.

Лабораторная работа служит для оценки освоения общепрофессиональных и профессиональных компетенций уровня «уметь» и «владеть».

Лабораторные работы включают задания по обработке количественных и качественных данных и решения исследовательских задач на их основе.

Поскольку задания являются обширными, непосредственно в аудитории преподавателем разбирается постановка задачи, обосновываются и демонстрируются инструменты необходимые для ее решения, уточняются требования к оформлению результатов.

Окончательное выполнение лабораторной работы происходит в форме самостоятельной домашней работы.

Выполненная домашняя работа сдается по расписанию следующей лабораторной работы в виде файла.

Работа проверяется преподавателем. Ошибки обсуждаются со студентом. Выставляется оценка.

*Шкала оценивания уровня умений с помощью лабораторной работы*

| | **Низкий, 0-30 баллов** | **Фрагментарный, 31-59 баллов** | **Поверхностный, 60-69 баллов** | **Достаточный, 70-84 балла** | **Высокий, 85-100 баллов** | **оценка** | **вес** |
|---|---|---|---|---|---|---|---|
| Решение поставленной задачи | Задача решена неверно, ход решения ошибочен, есть грубые ошибки | Задача решена неверно, ход решения верен, есть грубые ошибки | Задача решена неверно, ход решения верен, есть не более 5 мелких ошибок, оказавших воздействие на ответ | Задача решена верно, есть не более 4 мелких ошибок. | Задача решена верно, есть не более 2 мелких ошибок | X1 | 0,6 |
| Оформление результатов | Не выдержаны требования к оформлению | Большая часть требований не выполнена | Есть не более 5 мелких ошибок в оформлении | Есть не более 4 мелких ошибок в оформлении | Есть не более 2 мелких ошибок в оформлении | X2 | 0,3 |

| | | | | |
|---|---|---|---|---|
| Своевременность сдачи | **Не своевременно, 0 баллов** | **Своевременно, 100 баллов** | X3 | 0,1 |
| **Итоговая оценка** | **0,6*X1+0.3*X2+0.1*X3** | | | |

**Тесты**

Тест – инструмент обязательного объективного контроля знаний студентов, обучающихся по дисциплинам, обеспечиваемых кафедрой ЭММ.

Целью тестирования является экспресс-оценка уровня знаний на основе использования стандартизованных вопросов или задач с ответами закрытого типа.

Тест служит для оценки освоения общепрофессиональных и профессиональных компетенций уровня «знать» и «уметь».

Преподаватель определяет количество вопросов для тестирования и время прохождения теста.

Тестирование проводится в системах ЭММ-тест, MyTest, Iren test.

*Алгоритм оценивания теста*

1. Определяется количество вопросов в тесте – N;
2. Рассчитывается вес вопроса – 100/N баллов;
3. Определяется общее количество баллов, полученных за тест **100/N*K**, где **K** – количество верных ответов.

*Шкала оценивания уровня знаний с помощью теста*

| | | | | |
|---|---|---|---|---|
| **Низкий, 0-30 баллов** | **Фрагментарный, 31-59 баллов** | **Поверхностный, 60-69 баллов** | **Достаточный, 70-84 балла** | **Высокий, 85-100 баллов** |

**Исследовательский проект**

Исследовательский проект – один из видов самостоятельной работы студентов, реализуемых кафедрой ЭММ.

Цель исследовательского проекта – на основании технического задания, разработанного преподавателем, провести в заданные сроки законченное исследование по заданной/выбранной теме, включающее, но не ограниченное следующими этапами: обзор литературы, разработка инструментов для сбора данных, сбор данных, формулирование гипотез, выбор методов обработки данных, обработка данных, описание полученных результатов, формулирование выводов, оформление исследовательского проекта в печатном виде.

Темы исследовательских проектов, как правило, предоставляются преподавателем, но возможна инициатива со стороны студентов.

Исследовательский проект служит для оценки освоения профессиональных компетенций уровня «уметь», «владеть».

| | **Низкий 0-30 баллов** | **Фрагментарный 31-59 баллов** | **Поверхностный 60-69 баллов** | **Достаточный 70-84 балла** | **Высокий 85-100 баллов** | **оценка** | **вес** |
|---|---|---|---|---|---|---|---|
| Формальное выполнение технического задания (ТЗ) | Цели ТЗ не достигнуты | Цели ТЗ не достигнуты, но направление исследования в целом верное. | Формально цели ТЗ достигнуты не полностью. Выводы не сделаны или не обоснованы | Формально цели ТЗ достигнуты, но не все выводы сделаны или обоснованы. | Формально цели ТЗ достигнуты Выводы в целом обоснованы. | X1 | 0,4 |
| Сбор и обработка данных | Данных недостаточно для проведения исследования | Данных достаточно, данные из одного источника, многие из них некорректны и неполны | Данных достаточно, из нескольких источников, корректны, но неполны. Есть риски неверных выводов. | Данных достаточно, из нескольких источников, корректны, неполны. Есть риски неверных выводов. | Данных достаточно, источники данных дополняют друг друга, данные корректны и полны. Риски неверных выводов минимальны или отсутствуют. | X2 | 0,3 |
| Количество используемых источников информации | Не более 2 | Не более 3 | Не более 4 | Не более 5 | Более 5 | X3 | 0,1 |
| Оформление печатной работы | Не соответствует ГОСТ | Работа изобилует ошибками в оформлении | Не более 5 ошибок в оформлении работы | Не более 4 ошибок в оформлении работы | Не более 2 ошибок в оформлении работы | X4 | 0,1 |
| Своевременность сдачи | **Не своевременно, 0 баллов** | | **Своевременно, 100 баллов** | | | X5 | 0,1 |
| **Итоговая оценка** | **0,4*X1+0.3*X2+0.1*X3+0.1*X4+0.1*X5** | | | | | | |

**Приложение 6**

**Технологическая карта**

**по курсу Базы данных**

Дисциплина: Базы данных
Группа: ЭММ-1-23
Курс/семестр: 3/6
Количество кредитов (ЗЕ): 5
Отчетность: **Зачет с оценкой**
Преподаватель: Мусакулова Жылдыз Абдыманаповна

| **Название модулей дисциплины согласно РПД** | **Контроль** | **Форма контроля** | **зачетный минимум** | **зачетный максимум** | **график контроля** |
|---|---|---|---|---|---|
| Модуль 1 | | | | | |
| СУБД Microsoft Access. Базовые понятия. | Текущий контроль | Лабораторные и самостоятельные работы. | 5 | 8 | 26 |
| | Рубежный контроль | Контрольная работа / Тест | 3 | 6 | |
| Модуль 2 | | | | | |
| СУБД Microsoft Access. Объекты. | Текущий контроль | Лабораторные и самостоятельные работы. | 5 | 8 | 31 |
| | Рубежный контроль | Контрольная работа / Тест | 3 | 6 | |
| Модуль 3 | | | | | |
| СУБД PostgreeSQL. | Текущий контроль | Лабораторные и самостоятельные работы. | 5 | 8 | 33 |

| | | | | | |
|---|---|---|---|---|---|
| | Рубежный контроль | Контрольная работа / Тест | 3 | 6 | |
| Модуль 4 | | | | | |
| Запросы Pl/SQL | Текущий контроль | Лабораторные и самостоятельные работы. | 5 | 8 | 38 |
| | Рубежный контроль | Контрольная работа / Тест | 3 | 6 | |
| Модуль 5 | | | | | |
| Интеграция БД и пользовательского интерфейса. | Текущий контроль | Лабораторные и самостоятельные работы. | 5 | 8 | 42 |
| | Рубежный контроль | Контрольная работа / Тест | 3 | 6 | |
| ВСЕГО за семестр | | | 40 | 70 | |
| Промежуточный контроль (Зачет с оценкой) | | Итоговый тест | 20 | 30 | |
| Семестровый рейтинг по дисциплине | | | 60 | 100 | |

**Вес работ по курсу**

**Базы данных**

| **Содержание дисциплины** | **Тип контроля** | **Форма контроля** | **Уровень освоения компетенции** | **Количество единиц** | **Максимальный балл за контрольную единицу/за весь контроль** | **Вес** | **зачетный максимум** |
|---|---|---|---|---|---|---|---|
| Модуль 1 | | | | | | | |
| Объекты базы данных. | Текущий контроль | Лабораторные работы | Уметь, владеть | 4 | 100/400 | 0.05 | 20 |
| | Рубежный контроль | Тест | Знать, уметь | 20 | 5/100 | 0.15 | 15 |
| Модуль 2 | | | | | | | |
| Создание пользовательского интерфейса**.** | Текущий контроль | Лабораторные работы | Уметь, владеть | 2 | 100/200 | 0.05 | 10 |
| | Рубежный контроль | Исследовательский проект | Уметь, владеть | 1 | 100/100 | 0.25 | 25 |
| Промежуточный контроль (Зачет) | | | | | | | 30 |
| Семестровый рейтинг по дисциплине | | | | | | | 100 |