**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Поволжский государственный университет телекоммуникаций и информатики»

**КОЛЛЕДЖ СВЯЗИ**

УТВЕРЖДАЮ

Директор КС ПГУТИ

\_\_\_\_\_\_\_\_\_\_\_\_\_Андреев Р.В.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

**СБОРНИК**

**практических занятий**

по дисциплине

**Основы проектирования баз данных**

**Базы данных**

для специальностей:

09.02.03 – Программирование в компьютерных системах

09.02.04 – Информационные системы

09.02.06 – Сетевое и системное администрирование

09.02.07 – Информационные системы и программирование

Номера занятий: **№ 1 – 14**

Сборник рассчитан на 30 часов

Составлен преподавателем Терехиным Д.В.

Рассмотрен на заседании П(Ц)К

«Информационные системы и технологии»

Протокол № \_\_ от \_\_\_\_\_\_\_\_\_\_2020г.

Председатель П(Ц)К\_\_\_\_\_\_\_\_Шомас Е.А.

Самара

2020 г

**Перечень компетенций**

|  |  |
| --- | --- |
| **Общие компетенции** | |
| **ОК 1** | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам; |
| **ОК 2** | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; |
| **ОК 4** | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; |
| **ОК 5** | Осуществлять устную и письменную коммуникацию на государственном языке с учётом особенностей социального и культурного контекста; |
| **ОК 9** | Использовать информационные технологии в профессиональной деятельности; |
| **ОК 10** | Пользоваться профессиональной документацией на государственном и иностранном языке |
| **Профессиональные компетенции (09.02.03)** | |
| **ПК 2.1** | Разрабатывать объекты баз данных |
| **ПК 2.2** | Реализовывать базу данных в конкретной системе управления базами данных (СУБД) |
| **ПК 2.3** | Решать вопросы администрирования базы данных |
| **ПК 2.4** | Реализовывать методы и технологии защиты информации в базах данных |
| **Профессиональные компетенции (09.02.04)** | |
| **ПК 1.1** | Собирать данные для анализа использования и функционирования информационной системы, участвовать в составлении отчетной документации, принимать участие в разработке проектной документации на модификацию информационной системы |
| **ПК 1.2** | Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности |
| **ПК 1.3** | Производить модификацию отдельных модулей информационной системы в соответствии с рабочим заданием, документировать произведенные изменения |
| **ПК 1.7** | Производить инсталляцию и настройку информационной системы в рамках своей компетенции, документировать результаты работ |
| **ПК 1.9** | Выполнять регламенты по обновлению, техническому сопровождению и восстановлению данных информационной системы, работать с технической документацией |
| **Профессиональные компетенции (09.02.06)** | |
| **ПК 1.2** | Осуществлять выбор технологии, инструментальных средств и средств вычислительной техники при организации процесса разработки и исследования объектов профессиональной деятельности |
| **ПК 1.5** | Выполнять требования нормативно – технической документации, иметь опыт оформления проектной документации |
| **Профессиональные компетенции (09.02.07)** | |
| **ПК 11.1** | Осуществлять сбор, обработку и анализ информации для проектирования баз данных |
| **ПК 11.2** | Проектировать базу данных на основе анализа предметной области |
| **ПК 11.3** | Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области |
| **ПК 11.4** | Реализовывать базу данных в конкретной системе управления базами данных |
| **ПК 11.5** | Администрировать базы данных |
| **ПК 11.6** | Защищать информацию в базе данных с использованием технологии защиты информации |

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 1**

**НАИМЕНОВАНИЕ:** Проектирование структуры базы данных. Нормализация таблиц

### ЦЕЛЬ РАБОТЫ: Изучить этапы проектирования модели баз данных. Спроектировать базу данных по исходным данным. Формировать компетенции ОК 1, 2, 4, 5, 9, 10; овладеть знаниями и умениями для освоения ПК 2.1 (09.02.03), ПК 1.1 (09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.1, 11.2 (09.02.07)

1. **ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:
   1. Что такое база данных? Для каких целей они создаются?
   2. Что значит реляционная БД?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

**5. ЗАДАНИЕ:**

5.1. Изучите теоретические вопросы по проектированию базы данных, типы моделей баз данных, виды взаимосвязей в базах данных, принципы нормализации таблиц баз данных

5.2. По исходным данным построить физическую модель базы данных, реализующую принцип работы предметной области вашего варианта

5.3. Выделите объекты и характеристики проектируемой информационной базы данных

5.4. Определите основные классы сущностей

5.5. Определите связи между сущностями и их типы

5.6. Изучить принципы нормализации таблиц баз данных

5.7. Постройте концептуальную модель, приведенную к 3НФ

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

6.1. Изучить по приложению материал работы

6.2. Нарисовать рисунки, отображающие разные виды связей.

6.3. Нарисовать информационно-логическую модель реляционной базы данных

6.4. Нарисовать физическую модель базы данных, которая должна содержать объекты предметной области

6.5. Ответить на контрольные вопросы

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2. Типы связей между объектами выбранной предметной области.

7.3. Физическая модель базы данных

7.4. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Назовите виды моделей баз данных. Отличия, достоинства, недостатки

8.2. Назовите виды взаимосвязей между объектами баз данных. Отличия, достоинства, недостатки

8.3. Что такое нормализация таблиц, для какой цели проводят процедуру нормализации?

8.4. Перечислите этапы проектирования

8.5. Дать определения понятиям сущность, экземпляр сущности, атрибут сущности, ключ сущности, связь. Привести примеры

**ПРИЛОЖЕНИЕ**

**Организация данных**

Слово "реляционная" происходит от английского relation - отношение. Отношение - тематическое понятие, но в терминологии моделей данных отношения удобно изображать в виде таблицы. При этом строки таблицы соответствуют кортежам отношения, а столбцы - атрибутам. Ключом называют любую функцию от атрибутов кортежа, которая может быть использована для идентификации кортежа. Такая функция может быть значением одного, из атрибутов (простой ключ), задаваться алгебраическим выражением, включающим значения нескольких атрибутов (составной ключ). Это означает, что данные в строках каждого из столбцов составного ключа могут повторяться, но комбинация данных каждой строки этих столбцов является уникальной. Например, в таблице Студенты есть столбцы Фамилии и Год рождения. В каждом из столбцов есть некоторые повторяющиеся данные, т.е. одинаковые фамилии и одинаковые года рождения. Но если студенты, имеющие одинаковые фамилии, имеют разные года рождения, то эти столбцы можно использовать в качестве составного ключа. Как правило, ключ является уникальным, т.е. каждый кортеж определяется значением ключа однозначно, но иногда используют и неуникальные ключи (ключи с повторениями). В локализованной (русифицированной) версии Access вводится термин ключевое поле, которое можно трактовать как первичный ключ.

В Access можно выделить три типа ключевых полей: простой ключ, составной ключ и внешний ключ.

Одно из важнейших достоинств реляционных баз данных состоит в том, что вы можете хранить логически сгруппированные данные в разных таблицах и задавать связи между ними, объединяя их в единую базу. Для задания связи таблицы должны иметь поля с одинаковыми именами или хотя бы с одинаковыми форматами данных. Связь между таблицами устанавливает отношения между совпадающими значениями в этих полях. Такая организация данных позволяет уменьшить избыточность хранимых данных, упрощает их ввод и организацию запросов и отчетов. Поясним это на примере. Допустим, вам в базе надо хранить, данные о студентах (фамилия, изучаемая дисциплина) и преподавателях (фамилия, номер кафедры, ученая степень, преподаваемая дисциплина). Если хранить данные в одной таблице, то в строке с фамилией студента, изучающего конкретную дисциплину, будут храниться все атрибуты преподавателя, читающего эту дисциплину. Это же огромная избыточность данных. А если хранить данные о студенте в одной таблице, о преподавателе - в другой и установить связь между полями "Читаемая дисциплина" - "Изучаемая дисциплина" (фактически это одинаковые поля), то избыточность хранимых данных многократно уменьшится без ущерба для логической организации информации.

В Access можно задать три вида связей между таблицами; Один-ко-многим, Многие-ко-многим и Один-к-одному.

Связь Один-ко-многим - наиболее часто используемый тип связи между таблицами. В такой связи каждой записи в таблице А может соответствовать несколько записей в таблице В (поля с этими записями называют внешними ключами), а запись в таблице В не может иметь более одной соответствующей ей записи в таблице А.

При связи Многие-ко-многим одной записи в таблице А может соответствовать несколько записей в таблице В, а одной записи в таблице В - несколько записей в таблице А. Такая схема реализуется только с помощью третьей (связующей) таблицы, ключ которой состоит по крайней мере из двух полей, одно из которых является общим с таблицей А, а другое - общим с таблицей В.

При связи Один-к-одному запись в таблице А может иметь не более одной связанной записи в таблице В и наоборот. Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением Один-к-одному применяют для разделения очень широких таблиц, для отделения части таблицы в целях ее защиты, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице.

Тип создаваемой связи зависит от полей, для которых определяется связь:

* связь Один-ко-многим создается в том случае, когда только одно из полей является ключевым или имеет уникальный индекс, т.е. значения в нем не повторяются;
* связь Один-к-одному создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы;
* связь Многие-ко-многим фактически представляет две связи типа один-ко-многим через третью таблицу, ключ которой состоит, по крайней мере, из двух полей, общих для двух других таблиц.

**Целостность данных**

Целостность данных означает систему правил, используемых в СУБД Access для поддержания связей между записями в связанных таблицах (таблиц, объединенных с помощью связи), а также обеспечивает защиту от случайного удаления или изменения связанных данных. Контролировать целостность данных можно, если выполнены следующие условия:

* связанное поле (поле, посредством которого осуществляется связь) одной таблицы является ключевым полем или имеет уникальный индекс;
* связанные поля имеют один тип данных. Здесь существует исключение. Поле счетчика может быть связано с числовым полем, если оно имеет тип Длинное целое,
* обе таблицы принадлежат одной базе данных Access. Если таблицы являются связанными, то они должны быть таблицами Access. Для установки целостности данных база данных, в которой находятся таблицы, должна быть открыта. Для связанных таблиц из баз данных других форматов установить целостность данных невозможно.

1. Перед разработкой информационно-логической модели реляционной базы данных рассмотрим, из каких информационных объектов должна состоять эта база данных. Можно выделить три объекта, которые не будут обладать избыточностью, - Студенты, Дисциплины и Преподаватели. Представим состав реквизитов этих объектов в виде "название объекта (перечень реквизитов)": Студенты (код студента, фамилия, имя, отчество, номер группы, дата рождения, стипендия, оценки). Дисциплины (код дисциплины, название дисциплины), Преподаватели (код преподавателя, фамилия, имя, отчество, дата рождения, телефон, заработная плата).

Рассмотрим связь между объектами Студенты и Дисциплины. Студент изучает несколько дисциплин, что соответствует многозначной связи и отражено на рис.1 двойной стрелкой. Понятно, что каждая дисциплина изучается множеством студентов. Это тоже многозначная связь, обозначаемая двойной стрелкой (связь "один" обозначена одинарной стрелкой). Таким образом, связь между объектами Студенты и Дисциплины - Многие-ко-многим (М : N).



Рис.1. Типы связей между объектами Студенты, Дисциплины и Преподаватели

Множественные связи усложняют управление базой данных, например, в СУБД Access при множественных связях нельзя использовать механизм каскадного обновления. Поэтому использовать такие связи нежелательно и нужно строить реляционную модель, не содержащую связей типа Многие-ко-многим. В Access дня контроля целостности данных с возможностью каскадного обновления и удаления данных необходимо создать вспомогательный объект связи, который состоит из ключевых реквизитов связываемых объектов и который может быть дополнен описательными реквизитами. В нашем случае таким новым объектом для связи служит объект Оценки, реквизитами которого являются код студента, код дисциплины и оценки. Каждый студент имеет оценки по нескольким дисциплинам, поэтому связь между объектами Студенты и Оценки будет Один-ко-многим (1:М). Каждую дисциплину сдает множество студентов, поэтому связь между объектами Дисциплины и Оценки также будет Один-ко-многим (1:М). В результате получаем информационно-логическую модель базы данных, приведенную на рис. 2

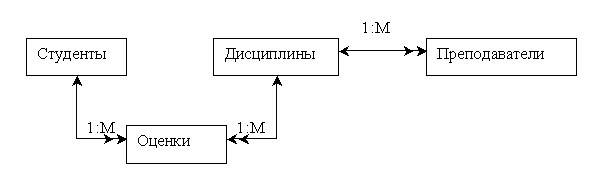
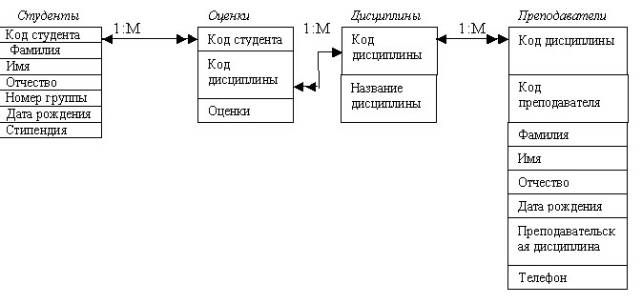


Рис. 2. Информационно-логическая модель реляционной базы данных

2. В реляционной базе данных в качестве объектов рассматриваются отношения, которые можно представить в виде таблиц. Таблицы между собой связываются посредством общих полей, т.е. одинаковых по форматам и, как правило, по названию, имеющихся в обеих таблицах. Рассмотрим, какие общие поля надо ввести в таблицы для обеспечения связности данных. В таблицах Студенты и Оценки таким полем будет "Код студента", в таблицах Дисциплины и Оценки - "Код дисциплины", в таблицах Преподаватели и Дисциплины - "Код дисциплины". Выбор цифровых кодов вместо фамилий или названий дисциплин обусловлен меньшим объемом информации в таких полях: например, число "2". по количеству символов значительно меньше слова "математика". В соответствии с этим логическая модель базы данных представлена на рис. 3, где жирными буквами выделены ключевые поля.

Рис. 3. Логическая модель базы данных

**ВАРИАНТЫ ЗАДАНИЙ**

***Вариант 1***

Описание предметной области (Ресторан)

Посетители ресторана обслуживаются за столиками. За одним столом может располагаться не более 4 посетителей, каждый из которых может сделать заказ тех или иных блюд. Столики обслуживают официанты. У одного официанта в обслуживании несколько столов.

Задачи для БД:

Есть ли свободные столы? Сколько посетителей обслужил официант за смену? Сколько каких блюд было реализовано?

***Вариант 2***

Описание предметной области (Колледж)

Студенты колледжа объединены в группы. Набор дисциплин, изучаемых студентом, зависит от номера группы в которой он учится. Преподаватели читают дисциплины и выставляют зачеты студентам. Один преподаватель может читать несколько дисциплин, но каждую дисциплину ведет один преподаватель. Задачи для БД: Какие дисциплины изучает студент? Какая оценка у студента по данной дисциплине? Кто выставил эту оценку?

***Вариант 3***

Описание предметной области (Театральная касса)

В театральной кассе продаются билеты на спектакли. Стоимость билета зависит от ряда, театра и спектакля. Каждый день в театре может идти не более одного спектакля. Спектакль характеризуется названием и автором. Каждый покупатель может купить сколько угодно билетов на любые спектакли.

Задачи для БД:

Какие спектакли идут в определенный день? Есть ли билеты на конкретный спектакль? Сколько стоит конкретный билет?

***Вариант 4***

Описание предметной области (Грузоперевозки)

АТП имеет грузовые автомобили с гос. номерами и организует перевозки для своих заказчиков. Стоимость перевозки зависит от расстояния и грузоподъемности автомобиля, который ее выполняет. Каждый заказчик может сделать заказ нескольких перевозок. Одну перевозку выполняет один грузовик.

Задачи для БД:

Какие грузовики свободны? Какой заказчик сделал самый дорогой заказ? Какой грузовик выполнил наибольшее количество заказов?

***Вариант 5***

Информационная система туристического клуба

Туристы, приходящие в туристический клуб, могут не только ходить в плановые походы, но и заниматься в различных секциях в течение всего года. Для этого они записываются в группы, относящиеся к определенным секциям.

Туpистов можно условно pазделить на любителей, споpтсменов и тpенеpов. Каждая из перечисленных категорий может иметь свой набор характеристик-атрибутов. Секции клуба возглавляются руководителями, в функции которых входит контроль за работой секции. Руководитель секции назначает каждой группе тренера. Тренер может тренировать несколько групп, причем необязательно принадлежащих его секции.

В течение года клуб организует различные походы. Каждый поход имеет свой маршрут, на который отводится определенное количество дней. По маршруту и количеству дней определяется категория сложности данного похода. Поход возглавляет инстpуктоp, котоpым может быть какой-либо тpенеp или споpтсмен. Он набиpает гpуппу в количестве 5-15 человек для своего похода, исходя из типа похода (пеший, конный, водный, гоpный) и физических данных туpистов (по их занятиям в секциях: водники, спелеологи, альпинисты и дpугие, с учетом специфики занятий - не умеющего плавать никогда не возьмут на сплав, а в пеший поход небольшой категоpии сложности могут взять любого туpиста).

1. Для организации работы клуба необходимо: получить список и общее число туpистов, занимающихся в клубе, в указанной секции, гpуппе, по половому пpизнаку, году pождения, возpасту; получить список и общее число тpенеpов указанной секции; получить перечень и общее число маpшpутов, по котоpым ходили туpисты из указанной секции, в обозначенный период вpемени.

***Вариант 6***

Информационная система зоопарка

Служащих зоопаpка можно подразделить на несколько категорий: ветеpинаpы, уборщики, дpессиpовщики, стpоители-pемонтники, pаботники администpации. Каждая из перечисленных категорий работников имеет уникальные атрибуты-характеристики, определяемые профессиональной направленностью. За каждым животным ухаживает опpеделенный кpуг служащих, пpичем только ветеpинаpам, убоpщикам и дpессиpовщикам pазpешен доступ в клетки к животным.

В зоопарке обитают животные различных климатических зон, поэтому часть животных на зиму необходимо пеpеводить в отапливаемые помещения. Животных можно подpазделить на хищников и травоядных. Пpи pасселении животных по клеткам необходимо учитывать

не только потpебности данного вида, но и их совместимость с животными в соседних клетках (нельзя pядом селить, напpимеp, волков и их добычу - pазличных копытных).

Для коpмления животных необходимы pазличные типы коpмов: pастительный, живой, мясо и pазличные комбикоpма. Растительный коpм это фpукты и овощи, зеpно и сено. Живой коpм - мыши, птицы, коpм для pыб. Для каждого вида животных pассчитывается свой pацион. Таким обpазом у каждого животного в зоопаpке имеется меню на каждый день, в котоpом указывается количество и вpемя коpмлений в день, количество и вид пищи (обезьянам необходимы фpукты и овощи, мелким хищникам - хоpькам, ласкам, совам, некотоpым кошачьим, змеям - надо давать мышей).

Ветеpинаpы должны пpоводить медосмотpы, следить за весом, pостом, pазвитием животного, ставить своевpеменно пpививки и заносить все эти данные в каpточку.

Необходимо получить список и общее число служащих зоопаpка; получить перечень и общее число всех животных в зоопаpке либо животных указанного вида; получить перечень и общее число нуждающихся в теплом помещении на зиму; получить перечень животных, которым не сделаны прививки вовремя; составить рацион для каждого вида животных.

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 2**

**НАИМЕНОВАНИЕ:** Знакомство с СУБД Access. Создание структуры таблиц базы данных. Ввод исходных данных

### 1. ЦЕЛЬ РАБОТЫ: Изучить интерфейс СУБД Access. Изучить режимы создания таблиц. Формировать компетенции ОК 1, 2, 4, 5, 9, 10; овладеть знаниями и умениями для освоения ПК 2.1, 2.2 (09.02.03), ПК 1.1, 1.3 (09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.4 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Что такое СУБД?

2.2 Назначение таблиц в СУБД?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 СУБД Access

**5. ЗАДАНИЕ:**

### 5.1. Изучите теоретические вопросы по созданию базы данных

### 5.2. Изучите состав окна Access

### 5.3. Изучите режимы создания таблиц

### 5.4. Создайте новую базу данных

### 5.5. Сохраните созданные таблицы

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

* 1. **Для создания новой базы данных:**
     1. Открыть Access
     2. Зайти в меню «Файл»
     3. Открыть вкладку «Создать»
     4. Нажать «Пустая база данных рабочего стола»
     5. Задать имя и путь хранения

6.1.6 Нажать «Создать»

### Для определения полей таблицы:

6.2.1 Включить режим «Конструктора» в правом верхнем углу

6.2.2 Введите в строку столбца имя 1 поля

6.2.3 В строке «Тип данных» указать тип данных

* 1. **Для сохранения** 
     1. Зайти в меню «Файл»
     2. Нажать «Сохранить»

### Заполнение базы данных.

* + 1. Заполните таблицы в соответствии с заданием в приложении

**Внимание!** При заполнении структуры таблиц будьте очень внимательны, для того чтобы избежать дальнейших проблем при работе с базой данных

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Структуры таблиц в СУБД Access

7.3 Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Назначение СУБД Access?

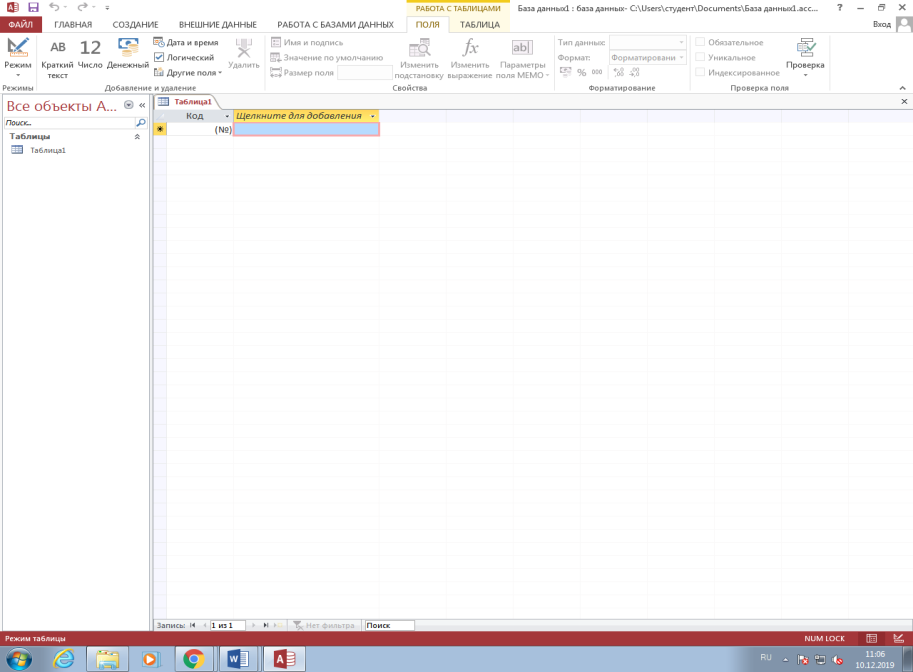
8.2. Что входит в состав окна Access?

8.3. Какие режимы создания таблиц Вы знаете?

8.4. Используемые типы данных и их свойства?

**ПРИЛОЖЕНИЕ**

Окно создания таблицы



**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 3**

**НАИМЕНОВАНИЕ:** Поиск и фильтрация данных в таблице. Основы конструирования запросов

### 1. ЦЕЛЬ РАБОТЫ: Научиться работать с данными таблиц базы данных. Изучить порядок конструирования запросов на выборку. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3 (09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.4 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Как настраивается фильтр в СУБД Access?

2.2 Назначение конструктора в СУБД?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 СУБД Access

**5. ЗАДАНИЕ:**

### 5.1. Запустить базу данных

### 5.2. Научиться выполнять поиск данных таблиц

### 5.3. Научиться сортировать данные таблиц

### 5.4. Научиться производить отбор записей, используя фильтр

### 5.5. Научиться производить корректировку данных в таблицах

### 5.6. Войти в режим Запрос и с помощью Конструктора создать запросы по заданию преподавателя

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

### Для поиска записей по заданному значению поля нужно нажать кнопку панели инструментов Найти. Курсор предварительно установить в это поле.

### Сортировка записей в порядке возрастания или убывания может быть выполнена как в режиме Таблицы, так и в режиме Формы. Для этого используются специальные кнопки на панели инструментов.

### Для просмотра и корректировки записей, удовлетворяющих указанным условиям отбора предусмотрены фильтры – это набор условий, применяемый для отбора подмножества записей из таблицы, формы или запроса.

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.4. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Что такое Фильтр?

8.2. Какие фильтры вы знаете?

8.3. Как можно отсортировать данные?

8.4. В каких режимах можно производить корректировку данных?

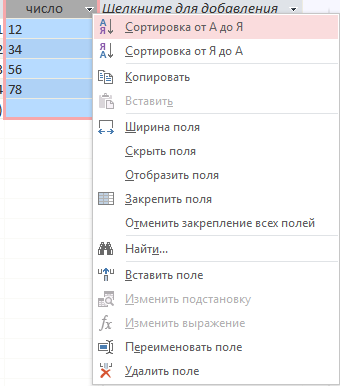
8.5. Что такое запрос?

8.6. Назначение запросов

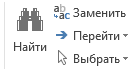
8.7. Основные виды запросов

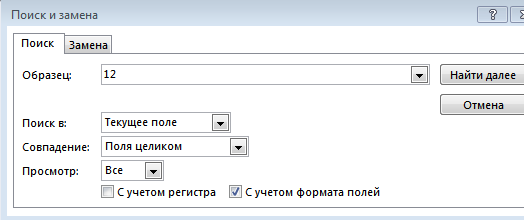
**ПРИЛОЖЕНИЕ**

***Сортировка:***



***Поиск:***





**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 4**

**НАИМЕНОВАНИЕ:** Изменение данных средствами запроса (запросы на обновление, добавление и удаление)

### 1. ЦЕЛЬ РАБОТЫ: Научится использовать запросы при работе с данными. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3 (09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.4 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Что такое запрос?

2.2 Как можно создать запрос?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 СУБД Access

**5. ЗАДАНИЕ:**

### 5.1. Запустить базу данных

### 5.2. Войти в режим Запрос и с помощью Конструктора создать запросы на обновление, добавление и удаление

### Прежде чем выполнить запрос необходимо предварительно просмотреть его и представить преподавателю

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

### 6.1. Запрос на обновление

### Для обновления данных можно использовать запрос на обновление. Значения для изменений в полях определяются в бланке запроса в строке Обновление. Первоначально этот запрос создается как запрос на выборку, а затем в окне конструктора запросов перемещается в запрос на обновление с помощью кнопки Обновление. После этого в бланке появляется строка Обновление

### 6.2. Запрос на добавление

### С помощью запроса на добавление производится добавление записей из таблицы запроса в другую таблицу.

### 6.3. Запрос на удаление

### Данный запрос позволяет удалить записи из одной таблицы или нескольких взаимосвязанных таблиц.

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Запросы

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

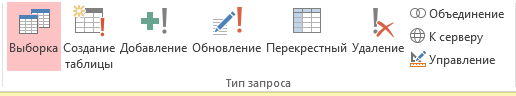
8.1. Какими запросами можно воспользоваться для изменения данных

8.2. Как можно предварительно просмотреть запрос, не выполняя его

8.3. На основе какого запроса строятся запросы на обновление, добавление и удаление

**ПРИЛОЖЕНИЕ**

***Выбор вида запроса:***



**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 5**

**НАИМЕНОВАНИЕ:** Создание сложных групповых и перекрестных запросов

### 1. ЦЕЛЬ РАБОТЫ: Научиться создавать сложные запросы к базе данных Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3 (09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.4 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Какие виды запросов существуют?

2.2 Для каких целей создаются запросы?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 СУБД Access

**5. ЗАДАНИЕ:**

### 5.1. Запустить базу данных

### 5.2. Создать следующие запросы:

### Создать перекрестную таблицу на примере таблиц заданной предметной области.

### Создать запрос по заданию преподавателя

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

### Для создания запроса 1. в окне базы данных нажать кнопку Создать. В открывшемся окне Новый запрос выбрать Перекрестный запрос. Далее по инструкции Мастера (окно Мастера представлено в Приложении).

### Отчитаться преподавателю и выполнить индивидуальное задание

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Запросы

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1 Какие запросы называются групповыми?

8.2. Назначение перекрестных запросов

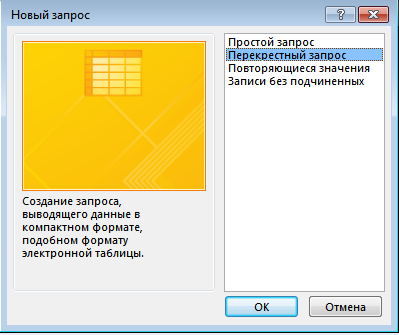
8.3. Основные принципы построения групповых запросов

8.4. Основные принципы построения перекрестных запросов

8.5. Принцип построения запроса на основе другого запроса

**ПРИЛОЖЕНИЕ**

***Окно Мастер запросов***



**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 6**

**НАИМЕНОВАНИЕ:** Создание БД и управление ей с помощью SQL - запросов

### ЦЕЛЬ РАБОТЫ: Научиться создавать простые SQL - запросы. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3, ПК 1.7(09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.5 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Назначение языка SQL?

2.2 Синтаксис простого запроса на выборку средствами языка SQL

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 MySQL

**5. ЗАДАНИЕ:**

### 5.1. Запустить программу MySQL

### 5.2. Создать базу данных при помощи SQL-запросов

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

6.1 Запустите сервер MySQL.

6.2 Следуйте инструкциям, представленным в пункте Приложение

### 6.3 Далее поработайте с таблицами (просмотр, удаление)

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Запросы

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Какие две составляющие части, необходимые для работы с базами данных, присутствуют в языке SQL?

8.2. Объясните порядок следования фраз в операторе SELECT. Какие фразы являются обязательными, а какие – нет?

8.3. Какой ряд типов данных определен в языке SQL?

8.4. Выполнить индивидуальное задание

**ПРИЛОЖЕНИЕ**

* Вызовите программу клиента **mysql**, вводя в строке приглашения **mysql**. Приглашение изменится на **mysql>.**
* Введите команду: **create database employees;**

(**Примечание**: Команда заканчивается символом точки с запятой).

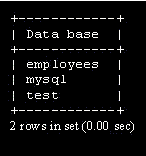
Сервер MySQL должен ответить примерно следующее

Результат работы команды создания таблицы

[Запрос обработан, изменилась 1 строка (0.00 сек)]

Это означает, что была успешно создана база данных. Теперь давайте посмотрим, сколько баз данных имеется в системе. Выполните следующую команду: **show databases;**

Сервер ответит списком баз данных



Здесь показаны три базы данных, две были созданы MySQL во время установки и вновь созданная база данных **employees**.

Чтобы вернуться снова к приглашению DOS, введите команду **quit** в приглашении **mysql**.

Рассмотрим команды MySQL для создания таблиц базы данных и выбора базы данных. Базы данных хранят данные в таблицах. Чем же являются эти таблицы? Проще всего таблицы можно представлять себе, как состоящие из строк и столбцов. Каждый столбец определяет данные определенного типа. Строки содержат отдельные записи. Рассмотрим таблицу [1](http://www.intuit.ru/department/database/mysql/3/2.html#table.3.1), в которой приведены персональные данные некоторых людей:

Таблица 1 – Персональные данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Возраст | Страна | | e-mail |
| Михаил Петров | 28 | Россия | misha@yandex.ru | |
| Джон Доусон | 32 | Австралия | j.dow@australia.com | |
| Морис Дрюон | 48 | Франция | md@france.fr | |
| Снежана | 19 | Болгария | sneg@bulgaria.com | |

Приведенная выше таблица содержит четыре столбца, в которых хранятся имя, возраст, страна, и адрес e-mail. Каждая строка содержит данные одного человека. Эта строка называется записью. Чтобы найти страну и адрес e-mail Снежаны, сначала надо выбрать имя в первом столбце, а затем посмотреть содержимое третьего и четвертого столбцов этой же строки.

База данных может содержать множество таблиц, именно таблицы содержат реальные данные. Следовательно, можно выделить связанные (или несвязанные) данные в различные таблицы. Для базы данных **employees** определена одна таблица, которая содержит данные компании о сотрудниках, а другая таблица будет содержать персональные данные. Давайте создадим первую таблицу. Команда SQL для создания такой таблицы выглядит следующим образом:

CREATE TABLE employee\_data

(

emp\_id int unsigned not null auto\_increment primary key,

f\_name varchar(20),

l\_name varchar(20),

title varchar(30),

age int,

yos int,

salary int,

perks int,

email varchar(60)

);

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 7**

**НАИМЕНОВАНИЕ:** Ввод данных в таблицы БД с помощью SQL - запросов

### ЦЕЛЬ РАБОТЫ: Научиться заполнять таблицы данными с помощью SQL – запросов. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3, ПК 1.7(09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.5 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Как создать новую БД средствами языка SQL?

2.2 Какой командой создается таблица в БД на языке SQL?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 MySQL

**5. ЗАДАНИЕ:**

### 5.1. Запустить программу MySQL

### 5.2. Выполнить работу, следуя порядку выполнения

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

6.1 Заполните таблицу данными через оператора INSERT (синтаксис оператора представлен в пункте Приложение)

6.2 Осуществите запрос на выборку данных из таблицы, используя оператор SELECT (синтаксис оператора представлен в пункте Приложение)

6.3 Выполните индивидуальные задания из пункта Приложение

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Запросы

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Какие операции могут программироваться с помощью языка SQL?

8.2. Какие ключевые слова используются при организации запросов?

8.3. Какие языки программирования поддерживает SQL?

8.4. Какой символ дает сообщение о том, что команда сформирована и готова к выполнению?

**ПРИЛОЖЕНИЕ**

Оператор **INSERT** заполняет таблицу данными. Вот общая форма INSERT:

**INSERT into table\_name (column1, column2, ...)**

**values (value1, value2...);**

где **table\_name** является именем таблицы, в которую надо внести данные; **column1**, **column2** и т.д. являются именами столбцов, а **value1, value2** и т.д. являются значениями для соответствующих столбцов.

Следующий оператор вносит первую запись в таблицу **employee\_data**,

INSERT INTO employee\_data

(f\_name, l\_name, title, age, yos, salary, perks, email)

values

("Михаил", "Петров", "директор", 28, 4, 200000,

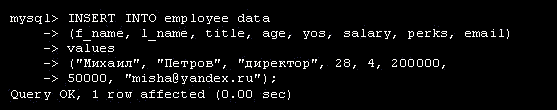
50000, "misha@yandex.ru");

Как и другие операторы MySQL, эту команду можно вводить на одной строке или разместить ее на нескольких строках.

Несколько важных моментов:

* Значениями для столбцов f\_name, l\_name, title и email являются текстовые строки, и они записываются в кавычках.
* Значениями для age, yos, salary и perks являются числа (целые), и они не имеют кавычек.
* Можно видеть, что данные заданы для всех столбцов кроме except emp\_id. Значение для этого столбца задает система MySQL, которая находит в столбце наибольшее значение, увеличивает его на единицу, и вставляет новое значение.

Если приведенная выше команда правильно введена в приглашении клиента mysql, то программа выведет сообщение об успешном выполнении



Создание дополнительных записей требует использования отдельных операторов INSERT. Чтобы облегчить эту работу можно поместить все операторы INSERT в файл.

### Запрос данных из таблицы MySQL

Таблица **employee\_data** содержит теперь достаточно данных, чтобы можно было начать с ней работать. Запрос данных выполняется с помощью команды MySQL **SELECT**. Оператор SELECT имеет следующий формат:

**SELECT имена\_столбцов from имя\_таблицы [WHERE ...условия];**

Часть оператора с условиями является необязательной (мы рассмотрим ее позже). По сути, требуется знать имена столбцов и имя таблицы, из которой извлекаются данные.

Например, чтобы извлечь имена и фамилии всех сотрудников, выполните следующую команду.

SELECT f\_name, l\_name from employee\_data;

Оператор приказывает MySQL вывести все данные из столбцов f\_name и l\_name. Результат работы оператора представлен на рисунке 1.

При ближайшем рассмотрении можно заметить, что данные представлены в том порядке, в котором они были введены. Более того, последняя строка указываете число строк в таблице - 21.

Чтобы вывести всю таблицу, можно либо ввести имена всех столбцов, либо воспользоваться упрощенной формой оператора SELECT.

**SELECT \* from employee\_data;**

Символ **\*** в этом выражении означает 'ВСЕ столбцы'. Поэтому этот оператор выводит все строки всех столбцов.

Рассмотрим ещё один пример.

**SELECT f\_name, l\_name, age from employee\_data;**

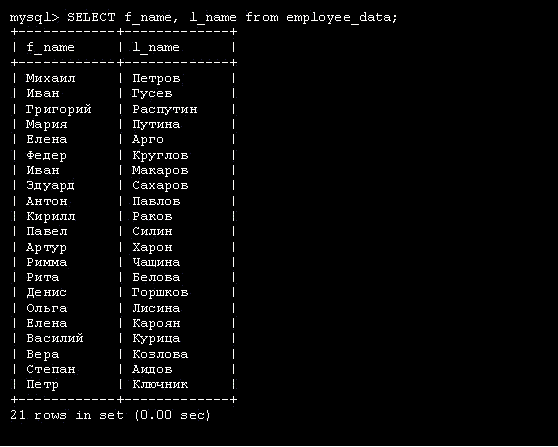


Рисунок 1 – Результат работы оператора SELECT

**Индивидуальные задания**

1. Напишите оператор SQL для создания новой базы данных с именем **addressbook**

2. Какой оператор используется для получения информации о таблице? Как используется этот оператор?

3. Как получить список всех баз данных, доступных в системе?

4. Напишите оператор для записи следующих данных в таблицу **employee\_data**

Имя: Рудольф

Фамилия: Курочкин

Должность: Программист

Возраст: 34

Стаж работы в компании: 2

Зарплата: 95000

Надбавки: 17000

email: rudolf@yandex.ru

5. Приведите две формы оператора SELECT, которые будут выводить все данные из таблицы **employee\_data**.

6. Как извлечь данные столбцов **f\_name, email** из таблицы **employee\_data**?

7. Напишите оператор для вывода данных из столбцов **salary, perks** и **yos** таблицы **employee\_data**.

8. Как узнать число строк в таблице с помощью оператора SELECT?

9. Как извлечь данные столбцов **salary, l\_name** из таблицы **employee\_data**?

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 8**

**НАИМЕНОВАНИЕ:** Выборка данных с помощью условий

### ЦЕЛЬ РАБОТЫ: Научиться осуществлять выбор данных с помощью SQL – запросов с условием Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3, ПК 1.7(09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.5 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Назначение оператора INSERT?

2.2 Назначение оператора SELECT?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 MySQL

**5. ЗАДАНИЕ:**

### 5.1. Запустить программу MySQL

### 5.2. Выполнить работу, следуя порядку выполнения

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

6.1 Создайте запрос на вывод имени и фамилии сотрудников с именем Иван (синтаксис запроса представлен в пункте Приложение);

6.2 Выбирает имена и фамилии всех сотрудников, которые являются программистами.

6.3 Выберите имена и фамилии всех сотрудников, которые старше 32 лет.

6.4 Найдите сотрудников, которые получают зарплату больше 120000.

6.5 Выведите всех сотрудников, которые имеют стаж работы в компании менее 3 лет.

6.6 Создайте запросы с оператором LIKE.

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Запросы

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Какие операции могут программироваться с помощью языка SQL?

8.2. Какие ключевые слова используются при организации запросов?

**ПРИЛОЖЕНИЕ**

Более подробно рассмотрим формат оператора SELECT. Его полный формат имеет вид:

**SELECT имена\_столбцов from имя\_таблицы [WHERE ...условия];**

В операторе SELECT условия являются необязательными. Оператор SELECT без условий выводит все данные из указанных столбцов. Одним из достоинств RDBMS является возможность извлекать данные на основе определенных условий. Теперь перейдём к рассмотрению операторов сравнения.

#### ***Операторы сравнения = и !=***

SELECT f\_name, l\_name from employee\_data where f\_name = 'Иван';

Этот оператор выводит имена и фамилии всех сотрудников, которые имеют имя Иван. Отметим, что слово Иван в условии заключено в одиночные кавычки. Можно использовать также двойные кавычки. Кавычки являются обязательными, так как MySQL будет порождать ошибку при их отсутствии. Кроме того сравнения MySQL не различают регистр символов, что означает, что с равным успехом можно использовать "Иван", "иван" и даже "ИвАн".

**SELECT f\_name,l\_name from employee\_data where title="программист";**

Выбирает имена и фамилии всех сотрудников, которые являются программистами.

**SELECT f\_name, l\_name from employee\_data where age = 32;**

Это список имен и фамилий всех сотрудников с возрастом 32 года. Вспомните, что тип столбца age был задан как int, поэтому кавычки вокруг 32 не требуются. Это - незначительное различие между текстовым и целочисленным типами столбцов.

Оператор != означает 'не равно' и является противоположным оператору равенства.

#### ***Операторы больше и меньше***

Давайте получим имена и фамилии всех сотрудников, которые старше 32 лет.

SELECT f\_name, l\_name from employee\_data where age > 32;

Попробуем найти сотрудников, которые получают зарплату больше 120000.

SELECT f\_name, l\_name from employee\_data where salary > 120000;

Теперь перечислим всех сотрудников, которые имеют стаж работы в компании менее 3 лет.

SELECT f\_name, l\_name from employee\_data where yos < 3;

**Операторы <= и >=**

Используемые в основном с целочисленными данными операторы меньше или равно (<=) и больше или равно (>=) обеспечивают дополнительные возможности.

select f\_name, l\_name, age, salary

from employee\_data where age >= 32;

Результат запроса запишите в отчет.

Выборка содержит имена, возраст и зарплаты сотрудников, которым больше 32 лет.

select f\_name, l\_name from employee\_data where yos <= 2;

Запрос выводит имена сотрудников, которые работают в компании меньше 3 лет.

### Поиск текстовых данных по шаблону

В данной части мы рассмотрим поиск текстовых данных по шаблону с помощью предложения **where** и оператора **LIKE**.

Оператор сравнения на равенство (=) помогает выбрать одинаковые строки. Таким образом, чтобы перечислить имена сотрудников, которых зовут Иван, можно воспользоваться следующим оператором SELECT.

select f\_name, l\_name from employee\_data where f\_name = "Иван";

Как быть, если надо вывести данные о сотрудниках, имя которых начинается с буквы В? Язык SQL позволяет выполнить поиск строковых данных по шаблону. Для этого в предложении where используется оператор LIKE следующим образом.

select f\_name, l\_name from employee\_data where f\_name LIKE "В%";

Можно видеть, что здесь в условии вместо знака равенства используется LIKE и знак процента в шаблоне. Знак % действует как символ-заместитель (аналогично использованию \* в системах DOS и Linux). Он заменяет собой любую последовательность символов. Таким образом "В%" обозначает все строки, которые начинаются с буквы В. Аналогично "%В" выбирает строки, которые заканчиваются символом В, а "%В%" строки, которые содержат букву В.

Давайте выведем, например, всех сотрудников, которые имеют в названии должности строку "про".

select f\_name, l\_name, title from employee\_data

where title like '%про%';

Перечислим всех сотрудников, имена которых заканчиваются буквой 'а'. Это очень просто сделать.

mysql> select f\_name, l\_name from employee\_data

where f\_name like '%a';

**Внимание! Результаты выполнения всех запросов необходимо занести в отчет**

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 9**

**НАИМЕНОВАНИЕ:** Работа с операторами IN и BETWEEN

### ЦЕЛЬ РАБОТЫ: Научиться использовать операторы IN и BETWEEN для выборки данных. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3, ПК 1.7(09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.5 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Какие операторы сравнения вам известны?

2.2 Назначение оператора LIKE?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 MySQL

**5. ЗАДАНИЕ:**

### 5.1. Запустить программу MySQL

### 5.2. Выполнить работу, следуя порядку выполнения

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

6.1 Осуществите запрос для нахождения сотрудников, которые являются разработчиками Web или системными администраторами

6.2 Выведите список сотрудников, которые не занимают должность программиста или системного администратора

6.3 Создайте запрос, который выводит сотрудников, зарплата которых меньше 90000 или больше 150000.

6.4 Выполните индивидуальные задания, предложенные в пункте Приложение

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Запросы

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Какие операции могут программироваться с помощью языка SQL?

8.2. Какие ключевые слова используются при организации запросов?

8.3. Какие языки программирования поддерживает SQL?

8.4. Какой символ дает сообщение о том, что команда сформирована и готова к выполнению?

**ПРИЛОЖЕНИЕ**

Чтобы найти сотрудников, которые являются разработчиками Web или системными администраторами, можно использовать оператор SELECT следующего вида:

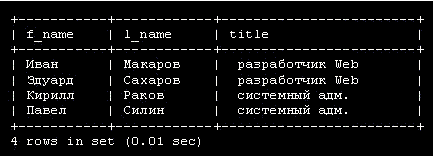
SELECT f\_name, l\_name, title from

-> employee\_data where

-> title = 'разработчик Web' OR

-> title = 'системный адм.';

Ниже приведен результат запроса.



В SQL имеется более простой способ сделать это с помощью оператора IN (в множестве). Его использование не представляет никаких трудностей.

SELECT f\_name, l\_name, title from

-> employee\_data where title

-> IN ('разработчик Web', 'системный адм.');

Использование NOT перед IN позволяет вывести данные, которые не входят в множество, определяемое условием IN. Следующий оператор выводит список сотрудников, которые не занимают должность программиста или системного администратора.

SELECT f\_name, l\_name, title from

-> employee\_data where title NOT IN

-> ('программист', 'системный адм.');

Оператор BETWEEN используется для определения целочисленных границ. Поэтому вместо age >= 32 AND age <= 40 можно использовать age BETWEEN 32 AND 40.

select f\_name, l\_name, age from

-> employee\_data where age BETWEEN

-> 32 AND 40;

NOT также можно использовать вместе с BETWEEN, как в следующем операторе, который выводит сотрудников, зарплата которых меньше 90000 или больше 150000.

select f\_name, l\_name, salary

-> from employee\_data where salary

-> NOT BETWEEN

-> 90000 AND 150000;

#### ***Индивидуальное задание***

1. Найдите всех сотрудников, которые занимают должность "старший программист" и "программист мультимедиа".

2. Выведите список имен сотрудников, зарплата которых составляет от 70000 до 90000.

3. Что делает следующий оператор?

SELECT f\_name, l\_name, title from

employee\_data where title NOT IN

('программист', 'старший программист',

'программист мультимедиа');

4. Вот более сложный оператор, который объединяет BETWEEN и IN. Что он делает?

SELECT f\_name, l\_name, title, age

from employee\_data where

title NOT IN

('программист', 'старший программист',

'программист мультимедиа') AND age

NOT BETWEEN 28 and 32;

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 10**

**НАИМЕНОВАНИЕ:** Сортировка, упорядочивание данных, поиск минимального и максимального значения

### ЦЕЛЬ РАБОТЫ: Научиться сортировать, упорядочивать данные с помощью SQL – запросов. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3, ПК 1.7(09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.5 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Назначение операторов IN и BETWEEN?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 MySQL

**5. ЗАДАНИЕ:**

### 5.1. Запустить программу MySQL

### 5.2. Выполнить работу, следуя порядку выполнения

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

6.1 Выполнить запросы на сортировку данных (оператор ORDER BY);

6.2 Внести ограничение на вывод данных из таблицы (оператор LIMIT);

6.3 Осуществите запрос с использованием исключения повторений оператор (DISTINCT);

6.4 Осуществите запрос на изменение данных (оператор UPDATE);

6.5 Выполните запросы с применением агрегатных функций (описание представлено в пункте Приложение).

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Запросы

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Какие операции могут программироваться с помощью языка SQL?

8.2. Какие ключевые слова используются при организации запросов?

8.3. Какие языки программирования поддерживает SQL?

8.4. Какой символ дает сообщение о том, что команда сформирована и готова к выполнению?

**ПРИЛОЖЕНИЕ**

Рассмотрим вопрос о том, как можно изменить порядок вывода данных, извлеченных из таблиц MySQL, используя предложение **ORDER BY** оператора SELECT.

Извлекаемые до сих пор данные всегда выводились в том порядке, в котором они были сохранены в таблице. В действительности SQL позволяет сортировать извлеченные данные с помощью предложения ORDER BY. Это предложение требует имя столбца, на основе которого будут сортироваться данные. Давайте посмотрим, как можно вывести имена сотрудников с упорядоченными по алфавиту фамилиями сотрудников (в возрастающем порядке).

SELECT l\_name, f\_name from

employee\_data ORDER BY l\_name;

А вот так сотрудников можно отсортировать по возрасту.

SELECT f\_name, l\_name, age

from employee\_data

ORDER BY age;

Предложение ORDER BY может сортировать в возрастающем порядке (ASCENDING или ASC) или в убывающем порядке (DESCENDING или DESC) в зависимости от указанного аргумента.

Чтобы вывести список сотрудников в убывающем порядке, можно использовать следующий оператор.

SELECT f\_name from employee\_data

ORDER by f\_name DESC;

Примечание: Возрастающий порядок (ASC) используется по умолчанию.

#### ***Задания***

1. Вывести список сотрудников в порядке, определяемом зарплатой, которую они получают.

2. Выведите список сотрудников в убывающем порядке их стажа работы в компании.

3. Что делает следующий оператор?

SELECT emp\_id, l\_name, title, age

from employee\_data ORDER BY

title DESC, age ASC;

4. Вывести список сотрудников (фамилию и имя), которые занимают должность "программист" или "разработчик Web" и отсортировать их фамилии по алфавиту.

### Ограничение количества извлекаемых данных

Далее рассмотрим, как ограничить число записей, выводимых оператором SELECT. По мере увеличения таблиц возникает необходимость вывода только подмножества данных. Это можно добиться с помощью предложения **LIMIT**. Например, чтобы вывести из таблицы имена только первых пяти сотрудников, используется оператор LIMIT с аргументом равным 5.

SELECT f\_name, l\_name from

employee\_data LIMIT 5;

Можно соединить оператор LIMIT с оператором **ORDER BY**. Таким образом, следующий оператор выведет четверых самых старых сотрудников компании.

SELECT f\_name, l\_name, age from

employee\_data ORDER BY age DESC

LIMIT 4;

Аналогично можно вывести двух самых младших сотрудников.

SELECT f\_name, l\_name, age from

employee\_data ORDER BY age

LIMIT 2;

### Извлечение подмножеств

LIMIT можно использовать также для извлечения подмножества данных, используя дополнительные аргументы. Общая форма оператора LIMIT имеет следующий вид:

SELECT (что-нибудь) from таблица LIMIT начальная строка, извлекаемое число записей;

SELECT f\_name, l\_name from

employee\_data LIMIT 6,3;

Будут извлечены три строки, начиная с шестой.

### Ключевое слово DISTINCT

Рассмотрим теперь, как выбрать и вывести записи таблиц MySQL с помощью ключевого слова DISTINCT (РАЗЛИЧНЫЙ), использование которого исключает появление повторяющихся данных. Чтобы вывести все должности базы данных компании, можно выполнить следующий оператор:

select title from employee\_data;

Можно видеть, что список содержит повторяющиеся данные. Предложение SQL DISTINCT выводит только уникальные данные. Вот как оно используется.

select DISTINCT title from employee\_data;

Из этого можно видеть, что в компании имеется 11 уникальных должностей.

Уникальные записи можно также отсортировать с помощью ORDER BY.

select DISTINCT age from employee\_data

ORDER BY age;

### Изменение записей

Команда UPDATE выполняет изменение данных в таблицах. Она имеет очень простой формат.

UPDATE имя\_таблицы SET

имя\_столбца\_1 = значение\_1,

имя\_столбца\_2 = значение\_2,

имя\_столбца\_3 = значение\_3, ...

[WHERE условия];

Как и все другие команды SQL можно вводить ее на одной строке или на нескольких строках. Рассмотрим несколько примеров.

Предположим, директор увеличил свою зарплату на 20000 и надбавки на 5000. Его предыдущая зарплата была 200000, а надбавки были 50000.

UPDATE employee\_data SET

salary=220000, perks=55000

WHERE title='директор';

Ниже приведен результат запроса.

Данные обновлены

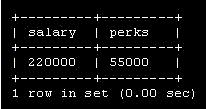
Можно проверить эту операцию, выводя данные из таблицы.

select salary, perks from

employee\_data WHERE

title = 'директор';

Ниже приведен результат запроса.



В действительности предыдущую зарплату знать не требуется. Можно воспользоваться арифметическими операторами. Следующий оператор сделает то же самое, при этом исходные данные знать заранее не требуется.

UPDATE employee\_data SET

salary = salary + 20000,

perks = perks + 5000

WHERE title='директор';

В качестве другого примера можно попробовать изменить название должности "разработчик Web" на "программист Web".

mysql> update employee\_data SET

-> title = 'программист Web'

-> WHERE title = 'разработчик Web';

Важно также перед выполнением изменений внимательно изучить часть оператора с условием, так как легко можно изменить не те данные. Оператор UPDATE без условий изменит все данные столбца во всех строках. Надо быть очень осторожным при внесении изменений.

### Поиск минимального и максимального значений

В MySQL имеются встроенные функции для вычисления минимального и максимального значений.

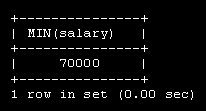
SQL имеет 5 агрегатных функций.

1. MIN(): минимальное значение
2. MAX(): максимальное значение
3. SUM(): сумма значений
4. AVG(): среднее значений
5. COUNT(): подсчитывает число записей

В этом параграфе мы рассмотрим поиск минимального и максимального значений столбца. **Минимальное значение**

select MIN(salary) from employee\_data;

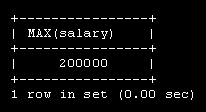
Ниже приведен результат запроса.



**Максимальное значение**

select MAX(salary) from employee\_data;

Ниже приведен результат запроса.



**Поиск среднего значения и суммы**

#### Суммирование значений столбца с помощью функции SUM.

Агрегатная функция SUM() вычисляет общую сумму значений в столбце. Для этого необходимо задать имя столбца, которое должно быть помещено внутри скобок. Давайте посмотрим, сколько компания BigFoot тратит на зарплату своих сотрудников.

select SUM(salary) from employee\_data;

Аналогично можно вывести общую сумму надбавок, выдаваемых сотрудникам.

select SUM(perks) from employee\_data;

Можно найти также общую сумму зарплаты и надбавок.

select sum(salary) + sum(perks) from employee\_data;

Здесь показаны также дополнительные возможности команды SELECT. Значения можно складывать, вычитать, умножать или делить. В действительности можно записывать полноценные арифметические выражения.

#### ***Вычисление среднего значения***

Агрегатная функция AVG() используется для вычисления среднего значения данных в столбце.

select avg(age) from employee\_data;

Пример выше вычисляет средний возраст сотрудников компании BigFoot, а следующий выводит среднюю зарплату.

select avg(salary) from employee\_data;

**Именование столбцов**

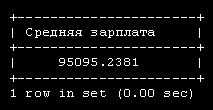
MySQL позволяет задавать имена для выводимых столбцов. Поэтому вместо f\_name или l\_name и т.д. можно использовать более понятные и наглядные термины. Это делается с помощью оператора AS.

select avg(salary) AS

'Средняя зарплата' from

employee\_data;

Ниже приведен результат запроса.



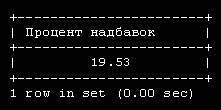
Такие псевдо-имена могут сделать вывод более понятным для пользователей. Важно только помнить, что при задании псевдо-имен с пробелами необходимо заключать такие имена в кавычки. Вот еще один пример:

select (SUM(perks)/SUM(salary) \* 100)

AS 'Процент надбавок' from

employee\_data;

Ниже приведен результат запроса.

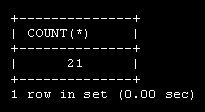


**Подсчет числа записей**

Агрегатная функция COUNT() подсчитывает и выводит общее число записей. Например, чтобы подсчитать общее число записей в таблице, выполните следующую команду.

select COUNT(\*) from employee\_data;

Ниже приведен результат запроса.



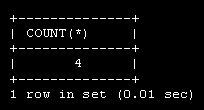
Как мы уже знаем, знак \* означает "все данные".

Теперь давайте подсчитаем общее число сотрудников, которые занимают должность "программист".

select COUNT(\*) from employee\_data

where title = 'программист';

Ниже приведен результат запроса.



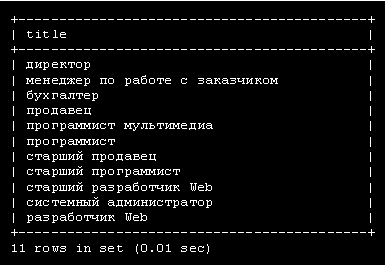
**Группировка данных**

Предложение GROUP BY позволяет группировать аналогичные данные. Поэтому, чтобы вывести все уникальные должности в таблице, можно выполнить команду

select title from employee\_data

GROUP BY title;

Ниже приведен результат запроса.

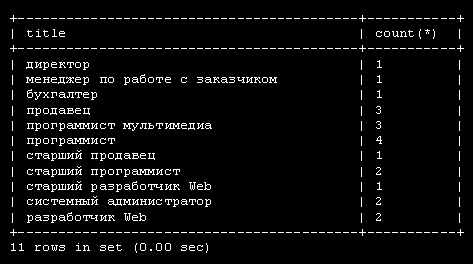


Вот как можно подсчитать число сотрудников имеющих определенную должность.

select title, count(\*)

from employee\_data GROUP BY title;

Ниже приведен результат запроса.



В предыдущей команде MySQL сначала создает группы различных должностей, а затем выполняет подсчет в каждой группе.

**Сортировка данных**

Теперь давайте найдем и выведем число сотрудников, имеющих различные должности, и отсортируем их с помощью ORDER BY.

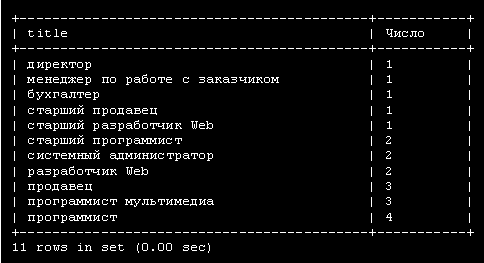
select title, count(\*) AS Number

from employee\_data

GROUP BY title

ORDER BY Number;

Ниже приведен результат запроса.



**Внимание! Результаты выполнения всех запросов необходимо занести в отчет**

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 11**

**НАИМЕНОВАНИЕ:** Создание многотабличных форм для загрузки подчиненных таблиц

### ЦЕЛЬ РАБОТЫ: Освоить порядок разработки форм с подчиненными таблицами. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3, ПК 1.7(09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.5, 11.6 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Назначение оператора UPDATE?

2.2 Какие агрегатные функции вам известны? Их назначение.

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 MySQL

**5. ЗАДАНИЕ:**

* 1. Создать форму для ввода информации в таблицы в удобном для пользователя формате.
  2. Создать сложную форму, объединив формы, созданные для разных таблиц.

### Разработать многотабличную форму без подчиненных форм

### Разработать форму в режиме конструктора

### 5.5 Отредактировать созданные формы

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

6.1 Загрузить базу данных. Пример создания формы представлен в пункте Приложение.

### 6.2 Разработать многотабличную форму без подчиненных форм.

### 6.3 Разработать форму в режиме конструктора

### 6.4 Создать поля со списком с помощью мастера

### 6.5 Создать поля со списком без использования мастера

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Формы

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Какая таблица является источником записей основной части формы?

8.2. Какая таблица является источником записей подчиненной формы?

8.3. Какая таблица является объектом загрузки данных?

8.4. Режимы создания форм

8.5. С помощью какой панели создается кнопка? Порядок создания

**ПРИЛОЖЕНИЕ**

### Рассмотрим пример создания формы

### 

Таблица – источник записей подчиненной формы

Таблица – источник записей основной части формы

Таблицы – источники полей для расшифровки идентификаторов подчиненной формы

**Таблица- объект загрузки записей**

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 12**

**НАИМЕНОВАНИЕ:** Создание макросов

### 1. ЦЕЛЬ РАБОТЫ: Изучить основы создания макросов и научиться конструировать макросы. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3, ПК 1.7(09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.5, 11.6 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Что такое макрос?

2.2 Каким способом можно создать макрос?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 MySQL

**5. ЗАДАНИЕ:**

### Запустить базу данных

* 1. Создать несколько макросов для открытия созданных на предыдущих занятиях таблиц, запросов и отчетов.
  2. Создать на панели инструментов кнопку для запуска одного из макросов.
  3. В режиме конструктора формы создать кнопки, запускающие все созданные макросы.
  4. Создать макрос с именем AUTOEXEC для открытия кнопочной формы и проверить его выполнение.

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

* 1. Для создания макроса, открывающего таблицы, рекомендуется выполнить следующие действия.
* Перейти на вкладку **Создание**, нажать кнопку **Макрос**, вследствие чего откроется окно конструктора макроса.
* Перетащить с помощью мыши таблицу Студенты в окно макроса. Таким образом, в макрос включается операция открытия таблицы Студенты.
* В списке макрокоманд выбрать **На Запись** и установить аргументы: **Тип Объекта-Таблица**
* Имя объекта-**Студенты**, Запись-**Последняя** Таким образом, после открытия таблицы, текущей станет последняя запись. В столбец
* Повторить те же действия для таблицы Экзамены.
* Сохранить макрос с именем «Открытие БД».
* Осуществить запуск макроса, воспользовавшись одним из способов, описанных в п.5.3.

Подобную последовательность действий нужно выполнить чтобы создать макросы «Просмотр отчета» и «Открытие запроса», для открытия любых ранее созданных отчета и запроса для таблицы Студенты.

**Примечание:** При создании макроса рекомендуется активизировать кнопку панели По шагам.

* 1. Создать на панели инструментов кнопку для запуска макроса «Просмотреть отчет». Выполнить макрос, щелкнув мышью по его кнопке на панели инструментов.

Если на кнопке нет текста, его можно установить, щелкнув правой кнопкой мыши и выбрав опцию Настройка. При открытом окне «Настройка» вновь щелкнуть правой клавишей на кнопке макроса и установить режим «Значок и текст».

* 1. Создать кнопочную форму, разместив на ней элементы (кнопки) для

− открытия базы данных;

− открытия запроса;

− просмотра отчета.

Кнопки будут использоваться для запуска ранее созданных макросов. Чтобы создать кнопку, нужно воспользоваться одним из способов, описанных в п.5.6.

Изменить размеры кнопок, шрифты для надписей, цвета. Сохранить форму с именем «Главная форма».

* 1. Создать автоматически выполняемый макрос с именем **AUTOEXEC** для открытия формы «Главная форма». Чтобы проверить работоспособность макроса, нужно закрыть, а затем вновь открыть базу данных.

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Макросы

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1. Что такое макрокоманда?

8.2 Что такое макрос?

8.3 Как создать макрос?

8.4 Какие существуют способы запуска макроса?

8.5 Какие ошибки могут появиться при выполнении макроса и как их исправить?

8.6 Как добавить на панель инструментов кнопку, запускающую макрос?

8.7 Как в режиме формы создать кнопку для запуска макроса?

8.8 Для чего предназначен макрос AutoExec?

8.9 Как создать макрос AutoExec?

**ПРИЛОЖЕНИЕ**

# Конструктор макросов

Открыть окно Конструктор макросов (Macro Design) для создания нового макроса Перейти на вкладку Создание, нажать кнопку Макрос, вследствие чего откроется окно конструктора макроса.

Окно Конструктора макросов построено аналогично окну Конструктора таблиц, т. е. разделено по горизонтали на две части: панель описаний и панель аргументов.

Верхняя часть окна Конструктора — **панель описаний** — состоит из нескольких столбцов. По умолчанию на этой панели находиться поле добавления команды,

После создания команды появятся параметры команды (рисунок 11.1)

Существует еще один очень простой способ ввода макрокоманды. Допустим, нужно создать макрос, который должен открывать форму "Клиенты» с левой части переносим форму «Клиенты» в макрос, автоматически создаться команда на открытие этой формы (рисунок 11.2)

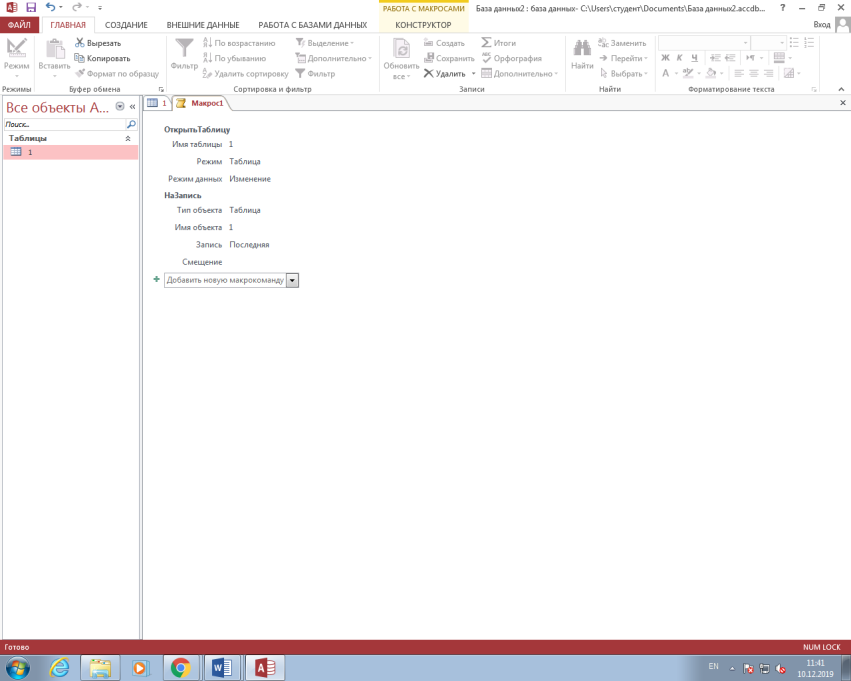


Рис. 11.1. Конструктор макросов и панель аргументов

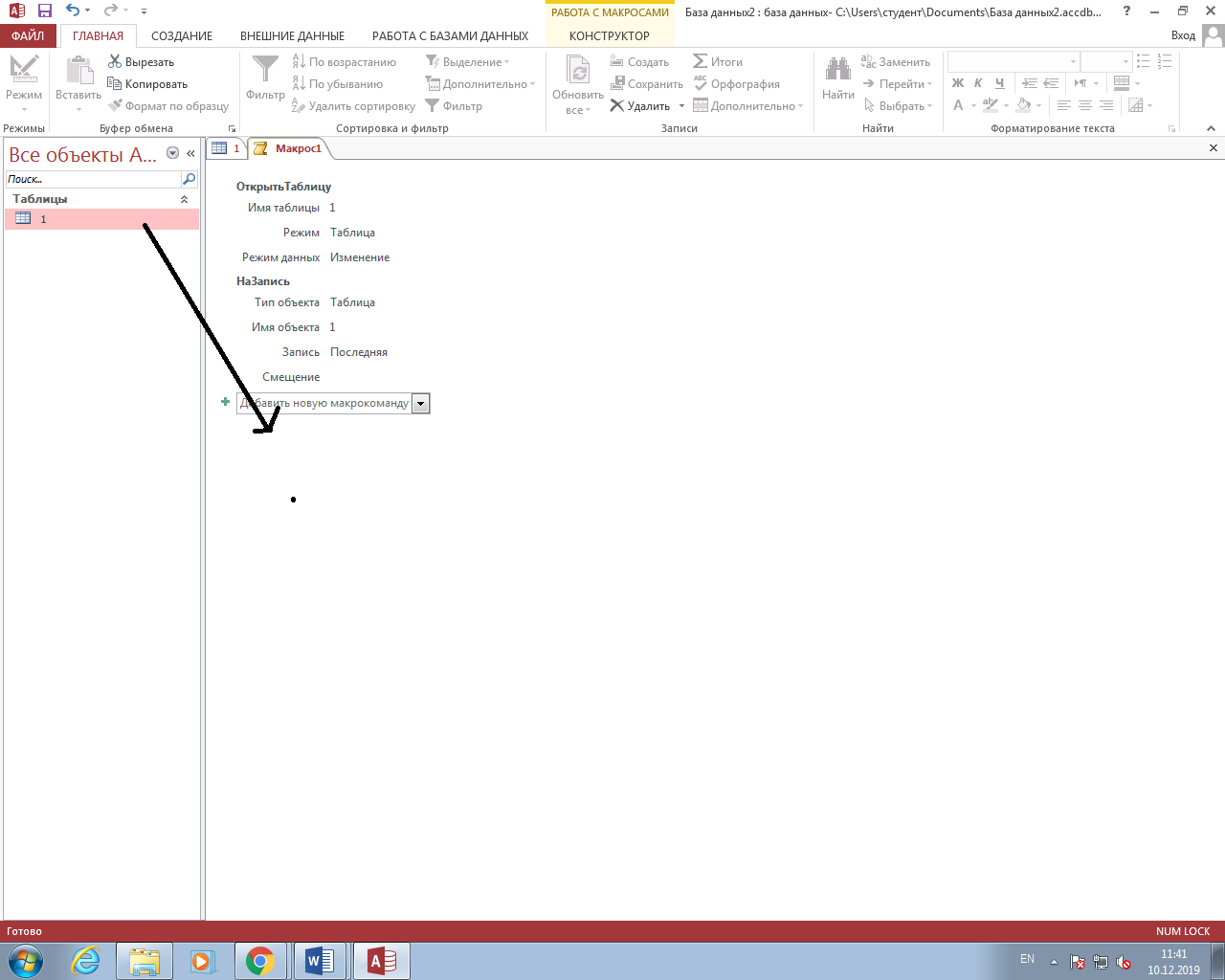


Рис. 11.2. Альтернативный способ создания макроса

Аналогично можно создавать макрокоманды, перетаскивая из окна База данных (Database) таблицы, запросы, отчеты и другие макросы. При этом в окне Конструктора макросов будут появляться соответствующие макрокоманды: ОткрытьТаблицу (ОрепТаblе), ОткрытьЗапрос (OpenQuery) , ОткрытьОтчет (OpenReport), ЗапускМакроса(RunMacro).

Для ввода аргументов макрокоманды чаще всего требуется выбирать значения из списков

Итак, созданная нами макрокоманда будет открывать форму "Клиенты" (Customers) в режиме Только чтение (Read Only) и показывать в ней только менеджеров по продажам. Как уже было сказано ранее, макрос может содержать последовательность из нескольких макрокоманд. Поэтому в этом же макросе в следующей строке мы выберем макрокоманду Сообщение (MsgBox), которая после открытия формы будет отображать диалоговое окно с сообщением о том, какие клиенты представлены в форме. Для этого введем в поле Сообщение (Message) в строке «Сообщение» пишем: Установлен фильтр Менеджеры по продажам, остальные поля можем оставить без изменения. АргументСигнал (Веер) определяет, будет ли вывод сообщения сопровождаться звуковым сигналом. АргументТип (Туре) определяет вид диалогового окна сообщения. Аргумент Заголовок (Title) позволяет задать текст, который будет выведен в заголовке окна сообщения.

Если база данных большая, отбор записей в форме может занять некоторое время, поэтому рекомендуется вывести на экран курсор в форме песочных часов, который будет показывать, что идет обработка данных. Для того чтобы сделать это, воспользуемся макрокомандой ПесочныеЧасы (Hourglass). Макрокоманды в макросе выполняются в той последовательности, в которой они записаны в столбце "Макрокоманда" (Action), т. е. сверху вниз. Макрокоманду ПесочныеЧасы (Hourglass) следует поставить первой, т. к. курсор должен измениться перед выводом формы. Таким образом, нам нужно вставить новую строку на панели описаний. Для этого создадим команду и перенесём с помощью мыши на 1 место В поле "Включить" (Hourglass On) панели аргументов появится значение Да (Yes), установленное по умолчанию, что нам и нужно. Чтобы вернуть первоначальный вид указателю мыши, в конце созданного макроса нужно ввести такую же макрокоманду ПесочныеЧасы (Hourglass), но со значением Нет (No) аргумента Включить (Hourglass On).

Строго говоря, эту макрокоманду лучше поставить перед макрокомандой вывода сообщения на экран. Сделать это элементарно, поскольку макрокоманды легко меняются местами с помощью мыши. Чтобы поменять две последние макрокоманды местами, выделите последнюю строку, щелкнув левой кнопкой мыши по команде и перетащите строку вверх. Горизонтальная линия показывает место нового положения строки. Отпустите кнопку мыши, когда эта линия окажется над предпоследней строкой.

Теперь остается только сохранить созданный макрос. Для этого нажать по макросы, на панели по центру, и нажать сохранить Появится окно сохранения макроса. Введите в поле ввода имя макроса, соблюдая правила именования объектов Access, например Менеджеры по продажам.

**Замечание**

В Microsoft Word и Excel имеются средства автоматической записи макросов. Эти средства очень удобны для изучения VBA, поскольку позволяют автоматически создать процедуру, включив запись макроса, выполнив необходимые действия и остановив запись. Тогда все действия пользователя, выполняемые после включения записи и до ее остановки, преобразуются в текст процедуры VBA. Затем этот текст можно просмотреть и отредактировать с помощью редактора VBA. В Microsoft Access средств автоматической записи макросов нет так же, как нет и средств автоматической генерации кода VBA (за исключением того, что создают мастера).

# Создание групп макросов

При разработке приложения с использованием макросов количество макросов может оказаться очень большим. Поэтому важно хорошо организовать доступ к нужным макросам. Для этого в одном объекте Макрос (Macro) можно объединить несколько макросов. Например, рекомендуется все макросы, связанные с событиями в форме или отчете, объединить в отдельный объект, соответствующий данной форме или отчету. Для того чтобы можно было это сделать, каждый макрос группы должен иметь свое имя, а имя объекта Макрос (Macro) будет являться именем группы макросов.

Чтобы создать группу макросов:

1. Откройте Конструктор макросов одним из тех способов, которые были описаны в предыдущем разделе.
2. Создать команду группа
3. Добавить нужные макросы внутрь группы

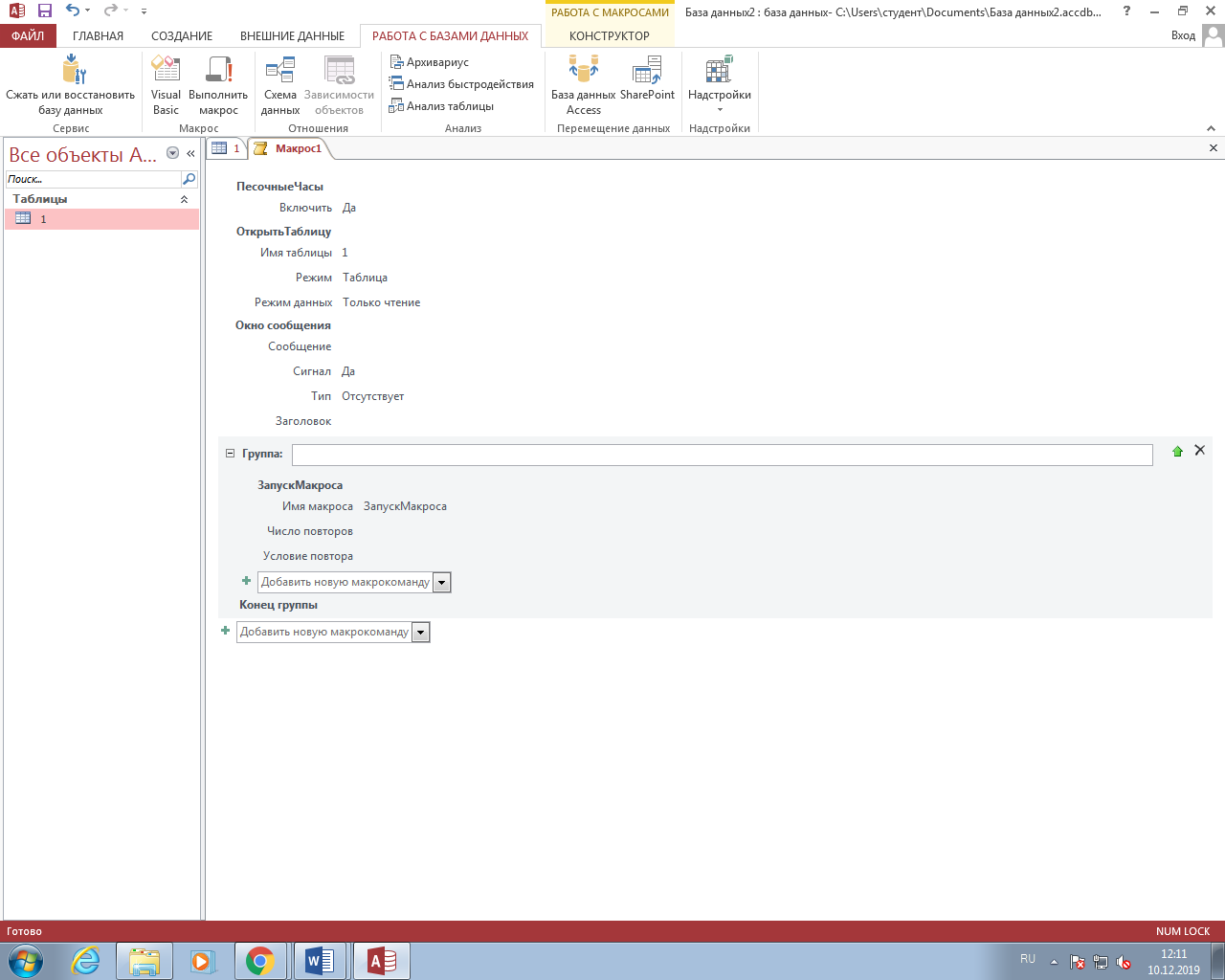


Рис. 11.3. Создание группы макросов

На рис. 11.З в окне Конструктора представлен макрос

Для запуска одного из макросов группы используется полное имя для ссылки на макрос. Полное имя макроса формируется таким образом: имя Группы. Имя Макроса.

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 13**

**НАИМЕНОВАНИЕ:** Основные приемы работы с отчетами. Создание и печать отчетов

### 1. ЦЕЛЬ РАБОТЫ: Изучить порядок конструирования отчетов. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3, ПК 1.7(09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.5, 11.6 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 Что такое отчет?

2.2 Каким способом можно создать отчет?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 MySQL

**5. ЗАДАНИЕ:**

### 5.1. Запустить базу данных

### 5.2. Создать однотабличный отчет

5.3. Создать отчет с подсчетом итогов

### 5.3. Просмотреть полученный отчет, отчитаться преподавателю

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

### 6.1 Через режим конструктора создать отчет. Ход выполнения действий по созданию отчета представлен в пункте Приложение

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Отчет

7.3. Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1 Что представляют собой отчеты в Access?

8.2 В чем их преимущества над другими методами вывода данных?

8.3 С чего начинается создание отчета?

8.4 Какие существуют мастера для создания отчета?

8.5 В чем отличие автоотчетов от других способов создания отчета?

8.6 Чем характеризуется режим Конструктора?

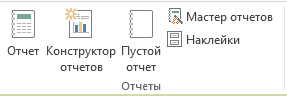
8.7 Назовите разделы отчета.

8.8 Назовите инструменты окна отчета.

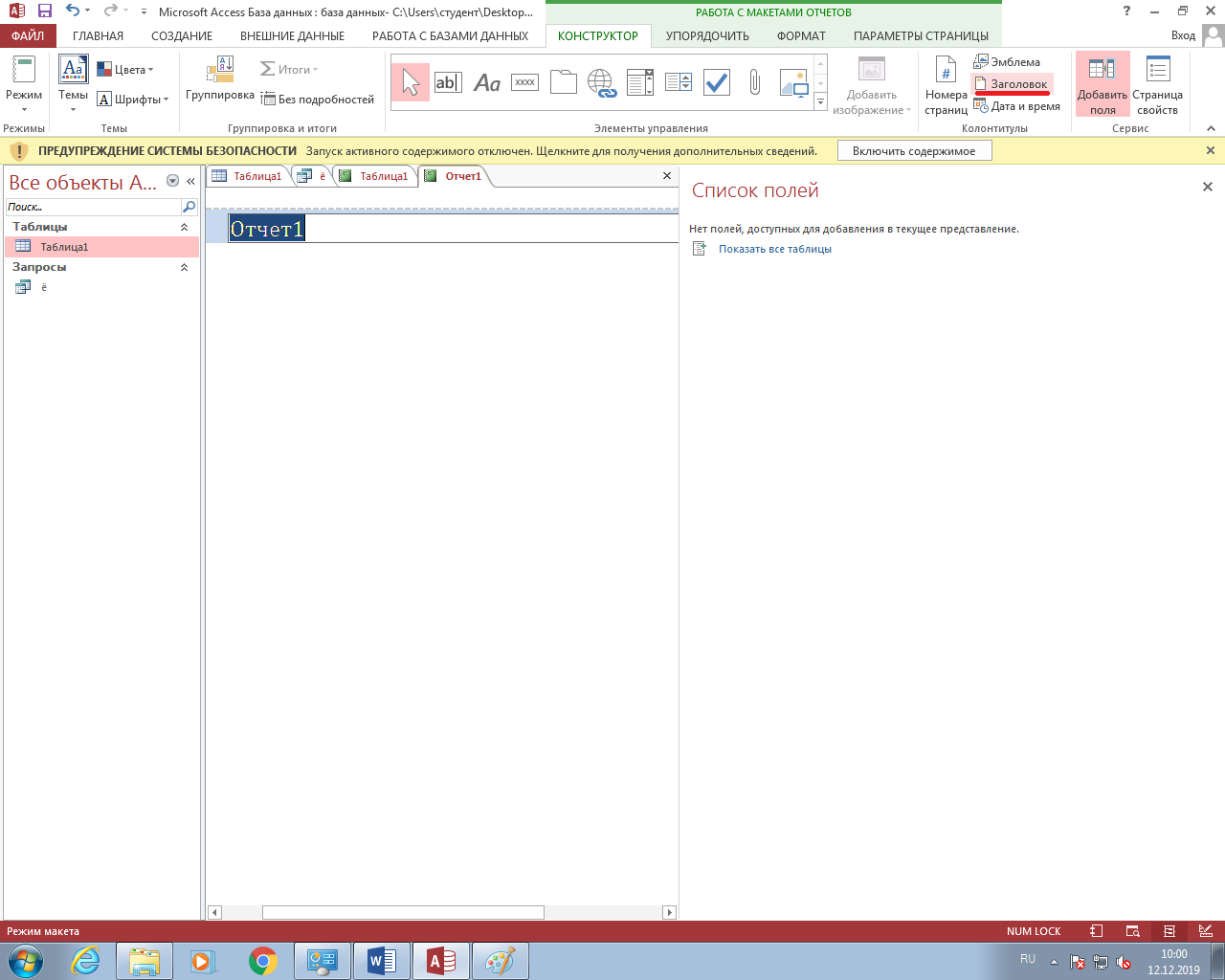
8.9 В каком разделе отчета выводятся итоги по группе?

**ПРИЛОЖЕНИЕ**

Для создания отчета в окне базы данных выбрать закладку Отчет и нажать кнопку Создать. В открывшемся окне Новый отчет выбрать Конструктор



### Отсутствующий раздел Заголовок отчета добавить с помощью кнопки Заголовок \ Примечание отчета.



**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ№ 14**

**НАИМЕНОВАНИЕ:** Интеграция БД с Access с другими приложениями

### 1. ЦЕЛЬ РАБОТЫ: Научиться интегрировать данные Access в другие приложения MS Office. Формирование компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10; овладеть знаниями и умениями для освоения ПК 2.1, ПК 2.2 (09.02.03), ПК 1.1, ПК 1.3, ПК 1.7(09.02.04), ПК 1.2, 1.5 (09.02.06), ПК 11.3, 11.5, 11.6 (09.02.07)

**2. ПОДГОТОВКА К ЗАНЯТИЮ:** по предложенной литературе изучить материал по теме практического занятия и ответить на следующие вопросы:

2.1 С какими приложениями можно производить интегрирование БД?

2.2 Для каких целей производят интегрирование одного приложения в лругое?

**3. ЛИТЕРАТУРА:**

3.1. Терехин Д.В. Основы проектирования баз данных. Учебное пособие – Самара, КС ПГУТИ 2020г.

**4.** **ПЕРЕЧЕНЬ ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**:

4.1 ПЭВМ, подключенные к сети Интернет

4.2 MySQL

**5. ЗАДАНИЕ:**

### 5.1. Запустить базу данных

### 5.2. Вставить в форму или отчет картинки, текст и электронную таблицу

### 5.3. Преобразовать какую-либо из таблиц БД в электронную таблицу Excel

**6. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

6.1 Изучить порядок вставки рисунков по приложению, осуществить вставку

6.2 Изучить по приложению пункт Вставка рабочего листа Excel в форму или отчет. Выполнить

6.3 Изучить по приложению пункт Вставка документов Word в таблицы, формы и отчеты Access. Выполнить

**7. СОДЕРЖАНИЕ ОТЧЕТА:**

7.1. Наименование и цель работы

7.2 Ответы на контрольные вопросы

**8. КОНТРОЛЬНЫЕ ВОПРОСЫ:**

8.1 Описать порядок интеграции объектов Access с другими компонентами MS Office

**ПРИЛОЖЕНИЕ**

# Вставка рисунков в формы и отчеты

В Microsoft Access существует возможность вставки рисунков в формы и отчеты. Использование рисунков может улучшить внешний вид создаваемых форм и отчетов. Например, в отчеты можно включать логотип фирмы, а в форме предоставлять возможность просмотра не только данных о сотрудниках фирмы, но и их фотографий. Для вставки рисунков в формы и отчеты используются специальные элементы управления — на панели объектов выбрать инструмент «рисунок», кнопка «добавить изображение», кнопка «добавить эмблему», свободная рамка объекта , присоединенная рамка объекта. Вставки рисунка зависит от того, как предполагается его использовать.

* Объект OLE можно хранить в одном из полей таблицы Access, например, в таблице "Сотрудники" (Employee) базы данных "Борей" хранятся фотографии сотрудников. Можно в полях таблицы Access хранить также документы Word. В таком случае для отображения этих объектов в формах или отчетах следует использовать элемент управления Присоединенная рамка объекта (Bound Object Frame).
* Объекты OLE могут храниться не в таблицах Access, а прямо в форме или отчете. С помощью такого способа в форме или отчете можно, например, сохранить логотип фирмы. В этом случае для отображения объекта OLE следует использовать элемент управления Свободная рамка объекта (Unbound Object Frame) или Рисунок (Image).
* Можно также для добавления рисунка в форму или отчет использовать свойство Рисунок(Picture) этой формы или отчета. Обычно так поступают, когда требуется включить фоновый рисунок (подложку), занимающий часть окна или все окно формы или отчета.

**Чтобы добавить фоновый рисунок в форму:**

1. Откройте форму в режиме Конструктора.
2. В окне свойств формы на вкладке Макет (Format) найдите свойство Рисунок (Picture). В поле ввода этого свойства введите путь и имя файла рисунка. Для этого можно нажать кнопку Построителя и выбрать в окне Выбор рисунка (Insert Picture) необходимый файл рисунка (рис. 14.1).

**Замечание**: Несколько стандартных файлов для фоновых рисунков находятся в папке MICROSOFT OFFICE\OFFICE\BITMAPS\STYLES.

1. Установите значение свойства Тип рисунка (Picture Type). Значений может быть два: Внедренный (Embedded) или Связанный (Linked). Если рисунок внедренный, он сохраняется в файле базы данных (mdb). Связанный рисунок хранится в отдельном файле на диске. О преимуществах и недостатках того и другого способа мы уже говорили выше. Если один и тот же рисунок вставляется во много форм или отчетов, то внедрение его приведет к "распуханию" файла mdb, т. к. храниться будет столько копий рисунка, сколько раз вы его внедряли. Если вы решите сделать его связанным, необходимо заботиться о том, чтобы его случайно не удалили или не переместили.
2. С помощью следующих свойств настройте отображение фонового рисунка.
   * Свойство Масштабы рисунка (Picture Size Mode) имеет три значения: Фрагмент (Clip), Вписать в рамку (Stretch), По размеру рамки (Zoom).

**Значение**: Фрагмент отображает рисунок в его исходном размере. Если размер рисунка превышает размер окна формы, то изображение рисунка обрезается. Значение Вписать в рамку изменяет размер рисунка так, чтобы он соответствовал окну формы. Использование данного значения может привести к искажению изображения рисунка. Значение По размеру рамки изменяет размер рисунка так, чтобы он соответствовал окну формы либо по ширине, либо по высоте. Использование данного значения не приведет ни к обрезанию рисунка, ни к искажению его пропорций.

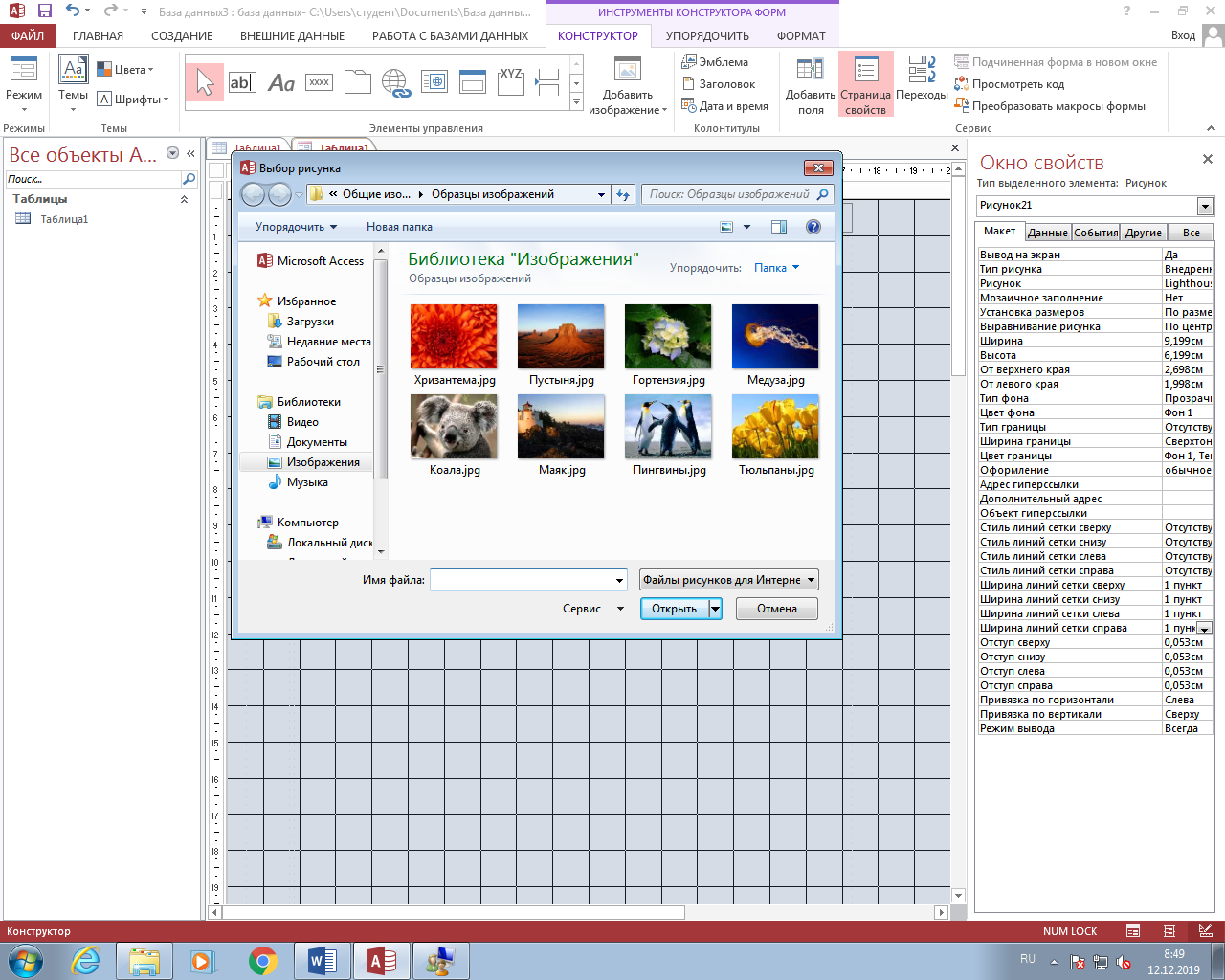


Рис. 14.1. Диалоговое окно Выбор рисунка

* + Если вставляется рисунок, занимающий не всю форму, свойство Выравнивание рисунка(Picture Alignment) позволяет определить положение его в форме: Сверху слева (Top Left),Сверху справа (Top Right), По центру (Center), Снизу слева (Bottom Left), Снизу справа(Bottom Right), По центру формы (Form Center).
  + Свойство Мозаичное заполнение (Picture Tiling) позволяет выводить повторяющееся изображение рисунка.

Пример использования рисунка в качестве фона в форме приведен на рис. 14.2.

Аналогично вставляется фоновый рисунок в отчет. Пример использования рисунка в качестве фона в отчете приведен на рис. 14.3.

Рисунок может быть вставлен в форму и в виде элемента управления формы. Для этого можно использовать два типа элементов управления: Рисунок (Image) и Свободная рамка объекта (Unbound Object Frame).

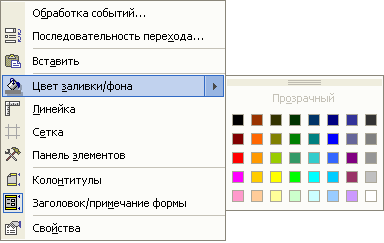


Рис. 14.2. Использование фонового рисунка в форме Microsoft Access

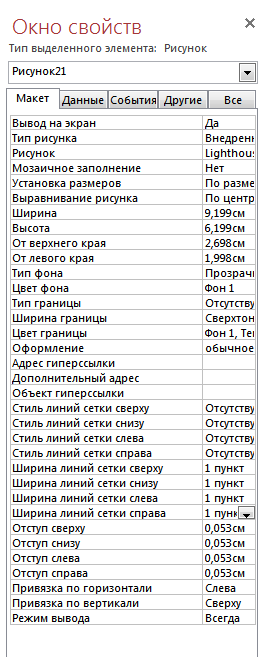


Рис. 14.3. Фоновый рисунок в отчете Microsoft Access

Если не требуется прямое изменение рисунка в форме или отчете, то лучше использовать Рисунок(Picture) (форма будет работать быстрее). Если же предполагается часто изменять рисунок, нужно вставить его с помощью свободной рамки объекта.

Чтобы вставить элемент управления Рисунок (Image), необходимо:

1. Открыть форму в режиме Конструктора формы или отчет в режиме Конструктора отчета.
2. Нажать кнопку Рисунок (Image) на панели элементов (кнопка Мастера элементов должна быть нажата, чтобы запустился Мастер рисунков).
3. Выбрать в форме или отчете место, в которое помещается рисунок, и щелкнуть на нем левой кнопкой мыши.
4. В раскрывшемся окне Вставка рисунка (Insert Picture) выбрать имя файла, содержащего добавляемый рисунок, и нажать кнопку ОК. Будет создан элемент управления, в котором помещается рисунок (рис. 14.4).
5. Установить значения свойств, которые влияют на то, как рисунок будет отображаться в форме или отчете. Эти свойства аналогичны соответствующим свойствам формы.

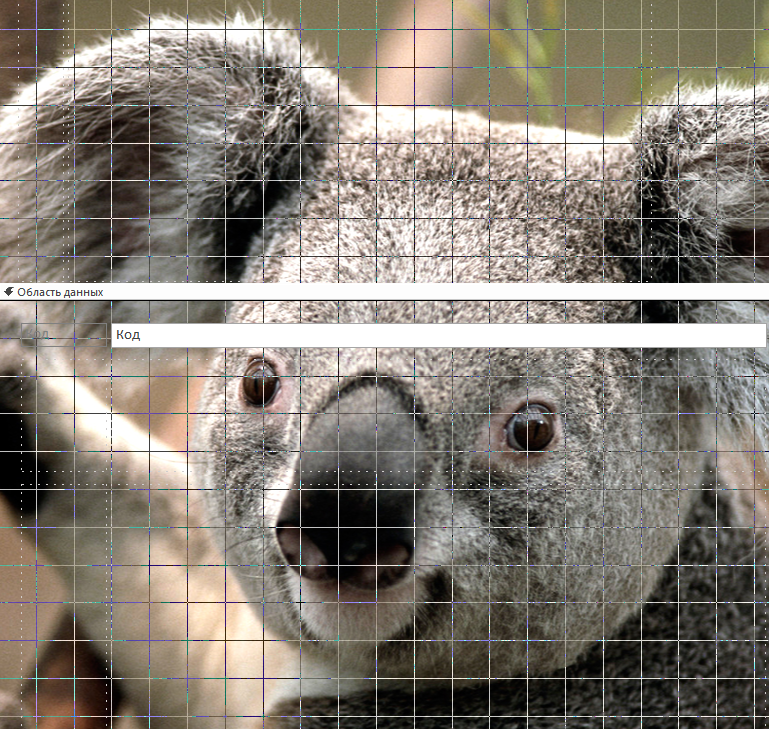


Рис. 14.4. Использование элемента Рисунок для вставки рисунка в форму

Чтобы вставить рисунок в форму или отчет с помощью свободной рамки объекта, необходимо:

1. Открыть форму в режиме Конструктора формы или отчет в режиме Конструктора отчета.
2. Нажать кнопку Свободная рамка объекта (Unbound Object Frame) на панели элементов.
3. Выбрать в форме или отчете место для вставляемого объекта и щелкнуть на нем левой кнопкой мыши. Появится диалоговое окно мастера Вставка объекта (Insert Object). С помощью мастера можно вставить уже готовый рисунок из файла или сначала создать его в соответствующем приложении, а затем вставить. Поэтому дальнейшие шаги могут быть разными.

Если вам еще предстоит создать рисунок, проделайте следующие операции.

1. В диалоговом окне Microsoft Access выбрать переключатель Создать новый (Create New), а затем в списке Тип объекта (Object Type) выбрать приложение, с помощью которого будет создаваться рисунок (рис. 14.5).

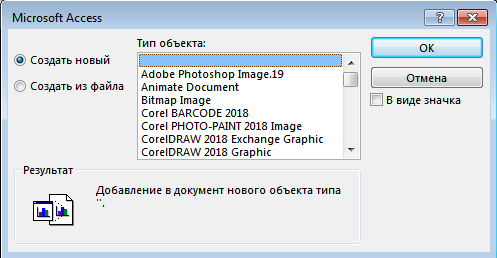


Рис. 14.5. Окно Microsoft Access, используемое для вставки рисунка с помощью свободной рамки объекта

1. Объект может отображаться в виде значка — это удобно, когда он содержит сведения, которые не должны все время выводиться на экран. В таком случае нужно просто установить флажок В виде значка (Display as Icon).
2. Нажать кнопку ОК.
3. Создать объект с помощью приложения, выбранного на шаге 1.
4. Закончив создание объекта, выбрать команду Закрыть (Close) в меню Файл (File) приложения, чтобы вернуться в Microsoft Access. Microsoft Access создает элемент управления Свободная рамка объекта (Unbound Object Frame) и отображает в ней рисунок.

Если же файл с рисунком у вас уже готов, чтобы вставить его прямо из файла, вам необходимо:

1. В диалоговом окне Microsoft Access выбрать переключатель Создать из файла (Create From File), а затем указать путь к файлу. Если точный путь вы не знаете, используйте кнопку Обзор (Browse) (рис. 14.6).

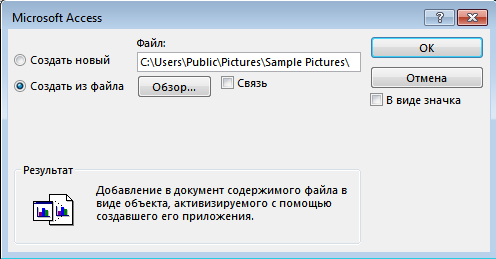


Рис. 14.6. Окно Microsoft Access при вставке рисунка из файла в отчет с помощью свободной рамки объекта

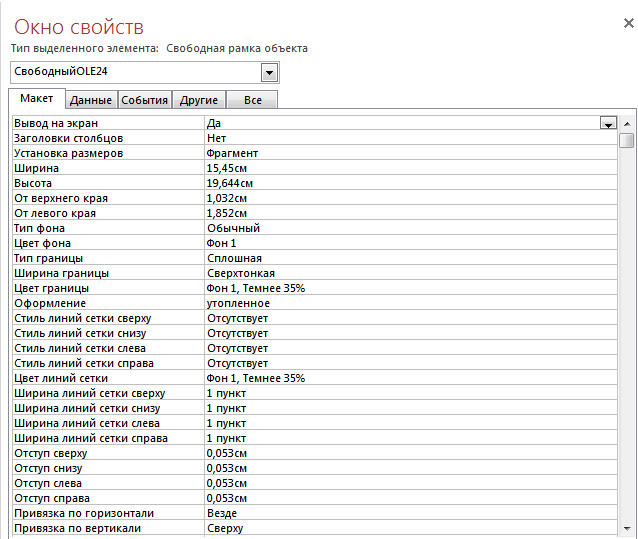


Рис. 14.7. Вставка рисунка Microsoft Paint в отчет с помощью свободной рамки объекта

1. Если нужно отобразить объект в виде значка, установить флажок В виде значка (Display As Icon).
2. Для нахождения связи с объектом установить флажок Связь (Link). Если флажок сброшен (по умолчанию), объект будет внедрен.
3. Нажать кнопку ОК.

Пример вставки рисунка, созданного с использованием Microsoft Paint, в форму с помощью свободной рамки объекта приведен на рис. 14.7.

После вставки объекта можно изменять его размеры и пропорции с помощью свойства Установка размеров (Size Mode). В большинстве случаев изменения в свободный объект могут вноситься в режиме Конструктора формы или отчета. Для того чтобы иметь возможность изменять объект в режиме формы, необходимо задать для свойства Доступ (Enabled) свободной рамки объекта значение Да (Yes), а для свойства Блокировка (Locked) — значение Нет (No). Чтобы отредактировать объект, дважды щелкните на нем мышью. Для этого только необходимо, чтобы на вашем компьютере было установлено приложение, в котором может редактироваться файл данного типа. Пример редактирования внедренного точечного рисунка представлен на рис. 14.8.

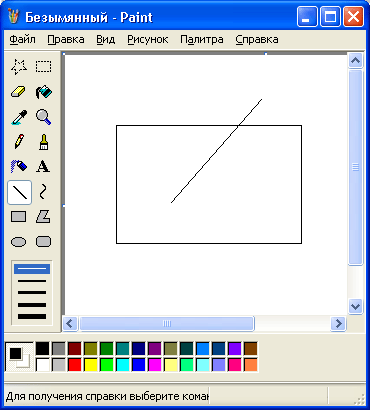


Рис. 14.8. Пример редактирования точечного рисунка с помощью Microsoft Paint

Для того чтобы выйти из режима редактирования, достаточно щелкнуть мышью за пределами рисунка.

В отличие от других элементов управления в форме, которые связаны с записями в таблице-источнике, рисунок в свободной рамке объекта не меняется при перемещении по записям в форме — он связан с самой формой, а не с данными, которые отображаются в форме. Рисунки, которые хранятся в записях таблицы, отображаются в форме с помощью присоединенной рамки объекта.

Для хранения рисунков и других объектов OLE в таблицах Access используется специальный тип поля:Объект OLE (OLE Object). Элемент управления Присоединенная рамка объекта (Bound Object Frame) создается одним из стандартных способов:

* перемещением с помощью мыши нужного поля из списка полей таблицы-источника. В этом случае будет сразу создан элемент управления, связанный с полем в таблице;
* с помощью кнопки Присоединенная рамка объекта (Bound Object Frame) на панели элементов. В данном случае будет создан элемент управления, для которого нужно сразу установить свойство Данные (RecordSource) — имя поля, содержащего объект OLE (рис. 14.9).

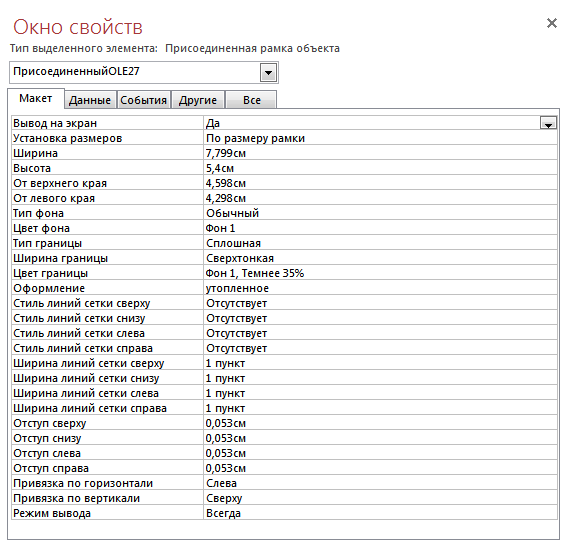


Рис. 14.9. Создание присоединенной рамки объекта в режиме Формы

Вставить рисунок в поле таблицы можно как в таблице, открытой в режиме Таблицы, так и в форме. Для этого нужно:

1. Открыть таблицу в режиме Таблицы или форму в режиме Формы.
2. Перейти на запись, в которую требуется вставить рисунок, и выбрать поле объекта OLE.
3. Выполнить команду Объект (Object) из меню Вставка (Insert).
4. В диалоговом окне Microsoft Access (см. рис. 15.5) выбрать переключатель Создать новый(Create New), а затем в списке Тип объекта (Object Type) выбрать тип приложения, в котором будет создаваться объект. Если файл с рисунком уже готов, то нужно выбрать переключательСоздать из файла (Create From File), a затем указать путь к файлу.
5. Для того чтобы отобразить объект в виде значка, установить флажок В виде значка (Display As Icon).
6. Нажать кнопку ОК.
7. Если вы создаете новый рисунок, то нужно создать его с помощью приложения выбранного на шаге 4.
8. Закончив создание объекта, выбрать команду Закрыть (Close) в меню Файл (File) приложения для возвращения в Microsoft Access.

Рисунок будет вставлен в запись в таблице. При этом в режиме Формы рисунок или иной объект автоматически будут отображаться на экране. В режиме Таблицы будет отображаться только тип объекта, например Растровый рисунок.

# Вставка рабочего листа Excel в форму или отчет

С помощью таких элементов управления, как Свободная рамка объекта и Присоединенная рамки объекта, в формах и отчетах Access можно отображать не только различные графические объекты, но и листы Microsoft Excel.

Так же как и рисунок, лист Microsoft Excel можно внедрить в форму или отчет или связать его с ними. При этом вставленный лист автоматически сохраняется в файле базы данных и всегда является доступным. Если обновить такой объект в форме или отчете, он будет изменен только в базе данных. Если лист Microsoft Excel связан, его можно просматривать и вносить изменения в форме или отчете. При этом изменения сохраняются в исходном файле объекта, а не в файле базы данных. Файл объекта можно обновлять независимо от базы данных. Последние изменения будут выведены на экран при следующем открытии формы или отчета. Связывание объектов удобно при работе с очень большими файлами, которые нежелательно включать в файл базы данных, а также с файлами, используемыми в нескольких формах и отчетах. Если связанный файл объекта перемешен, необходимо повторно установить с ним связь.

Рассмотрим возможность вставки листа Microsoft Excel в форму и отчет Microsoft Access с помощью свободной рамки объекта. Чтобы вставить новый лист Microsoft Excel в форму Microsoft Access, необходимо:

1. Открыть форму или отчет в режиме Конструктора.
2. Добавить в нее элемент управления Свободная рамка объекта (Unbound Object Frame). При этом в окне Microsoft Access выбрать переключатель Создать новый (Create New), а в спискеТип объекта (Object Type) выбрать Лист Microsoft Excel (Microsoft Excel Worksheet) (рис. 14.10). Если файл XLS уже создан, выберите переключатель Создать из файла (Create From File) и введите путь и имя нужного файла.

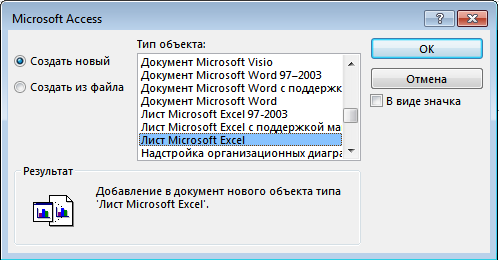


Рис. 14.10. Окно Microsoft Access при вставке нового листа Microsoft Excel в форму Microsoft Access с помощью свободной рамки объекта

1. Когда в форме появится элемент управления, в котором отображается пустая таблица Excel, введите в нее данные (это можно сделать прямо в режиме Конструктора) (рис. 14.11). Если вы вставляли готовый файл, то сразу будет отображена таблица Excel (рис. 14.12).

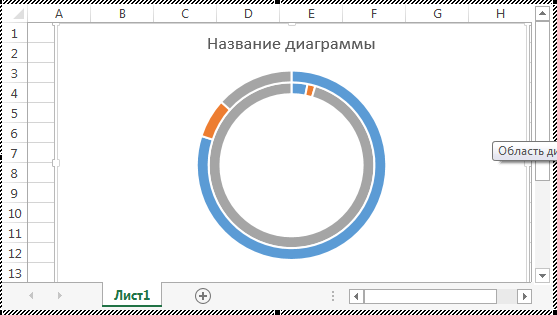


Рис. 14.11. Ввод данных на лист Microsoft Excel в режиме Конструктора форм Microsoft Access

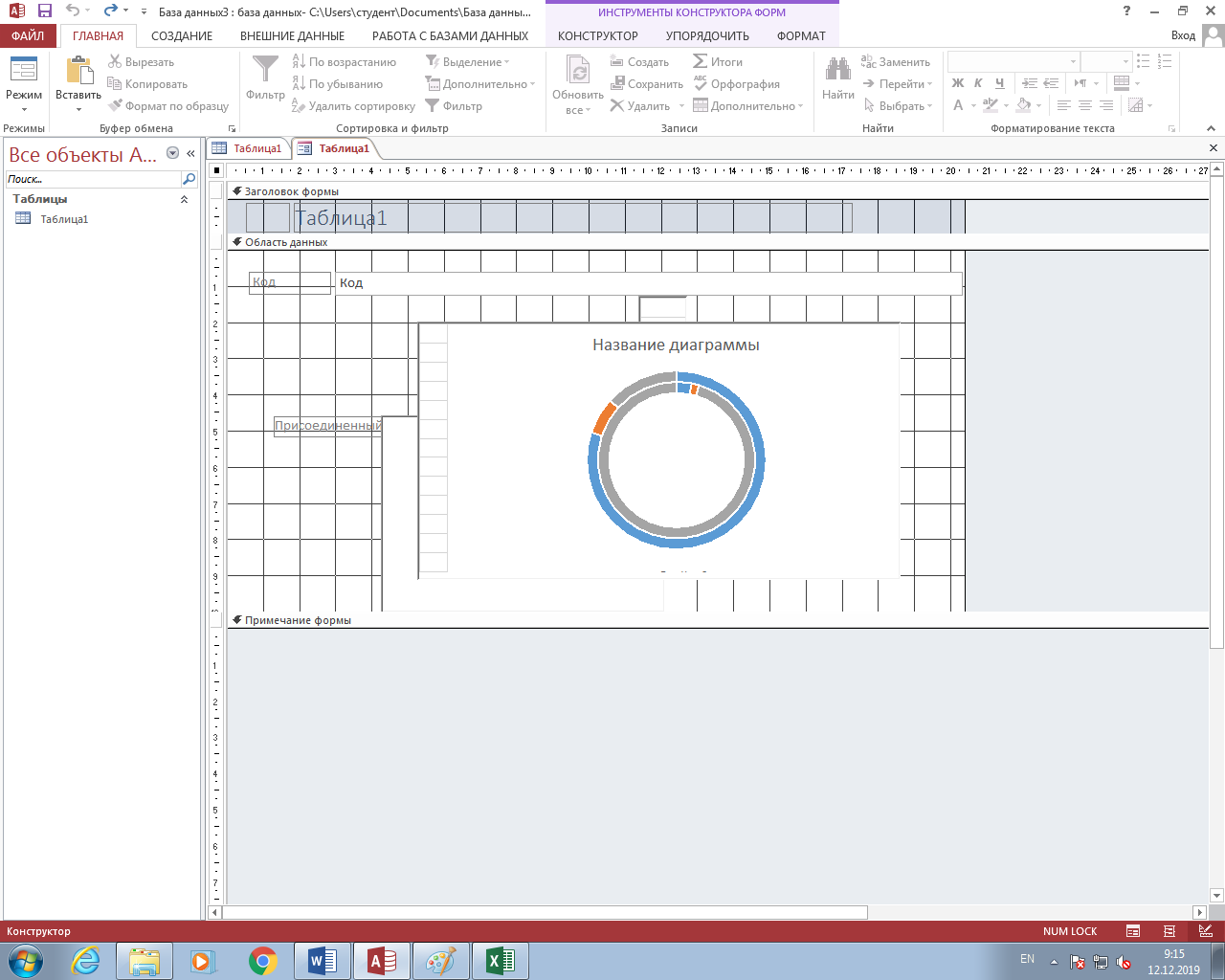


Рис. 14.12. Использование свободной рамки объекта для вставки существующего листа Microsoft Excel в форму Microsoft Access

1. Чтобы сделать доступным изменение данных на листе в режиме Формы, установите значение свойства Доступ (Enabled) для вставленного элемента управления равным Да (Yes), а свойство Блокировка (Locked) — Нет (No).

Увидеть данные, расположенные на листе Microsoft Excel, можно, открыв форму в режиме Формы. Чтобы иметь возможность изменять или добавлять данные на лист, нужно дважды щелкнуть кнопкой мыши на листе Microsoft Excel в форме (рис. 14.13).

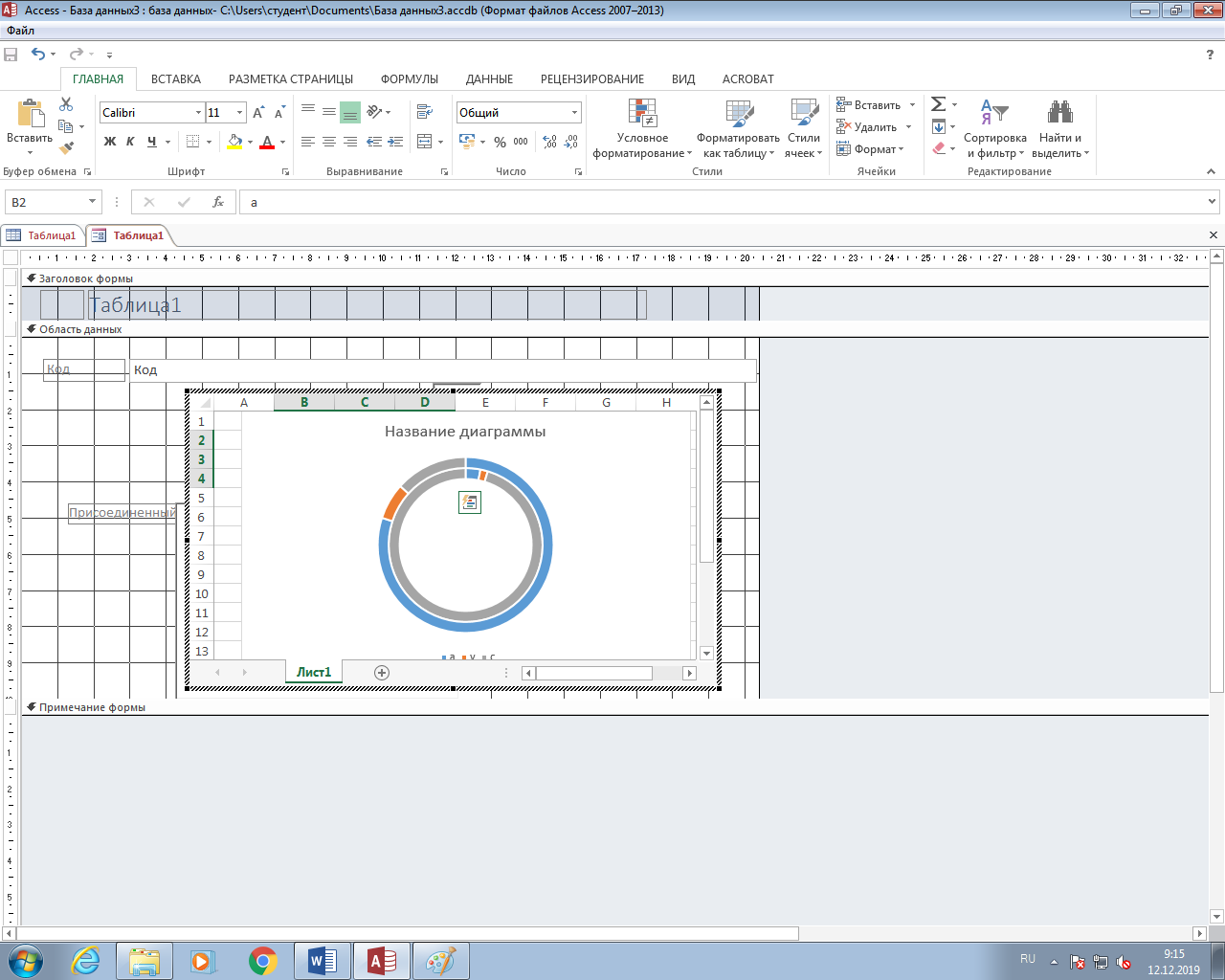


Рис. 14.13. Ввод данных на лист Microsoft Excel в режиме Формы

При работе с отчетами Microsoft Access следует выполнять аналогичную пошаговую процедуру. Увидеть данные, расположенные на листе Microsoft Excel, можно, открыв отчет в режиме просмотра.

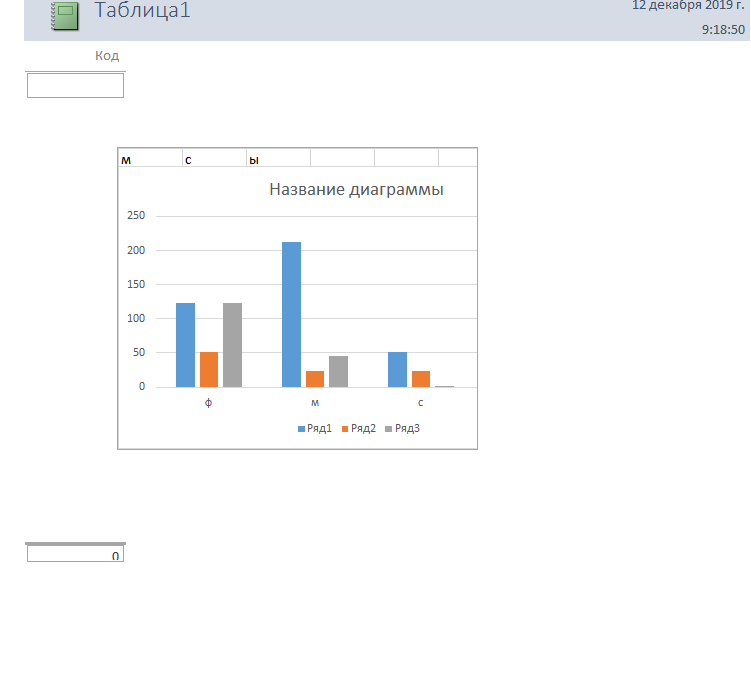


Рис. 14.14. Использование присоединенной рамки объекта для вставки листа Microsoft Excel в формуMicrosoft Access

Рассмотрим теперь возможность вставки листов Microsoft Excel с помощью присоединенной рамки объекта. В этом случае листы Excel будут храниться в записях таблицы Access (более разумным представляется хранить не сами таблицы, а ссылки на них, т. е. не внедрять листы Excel, а связывать их с таблицей Access). Сначала в таблице должно быть создано поле типа Объект OLE (OLE Object), которое будет содержать листы Excel. Затем в форме, отображающей записи этой таблицы, должен быть создан элемент Присоединенная рамка объекта (Bound Object Frame), связанный с этим полем. Вставка листа Microsoft Excel в поле таблицы или формы может быть выполнена с помощью команды меню Вставка, Объект (Insert, Object), как это уже описывалось ранее. На рис. 14.14 и 14.15 представлены примеры вставки листа Microsoft Excel в форму и отчет Access.

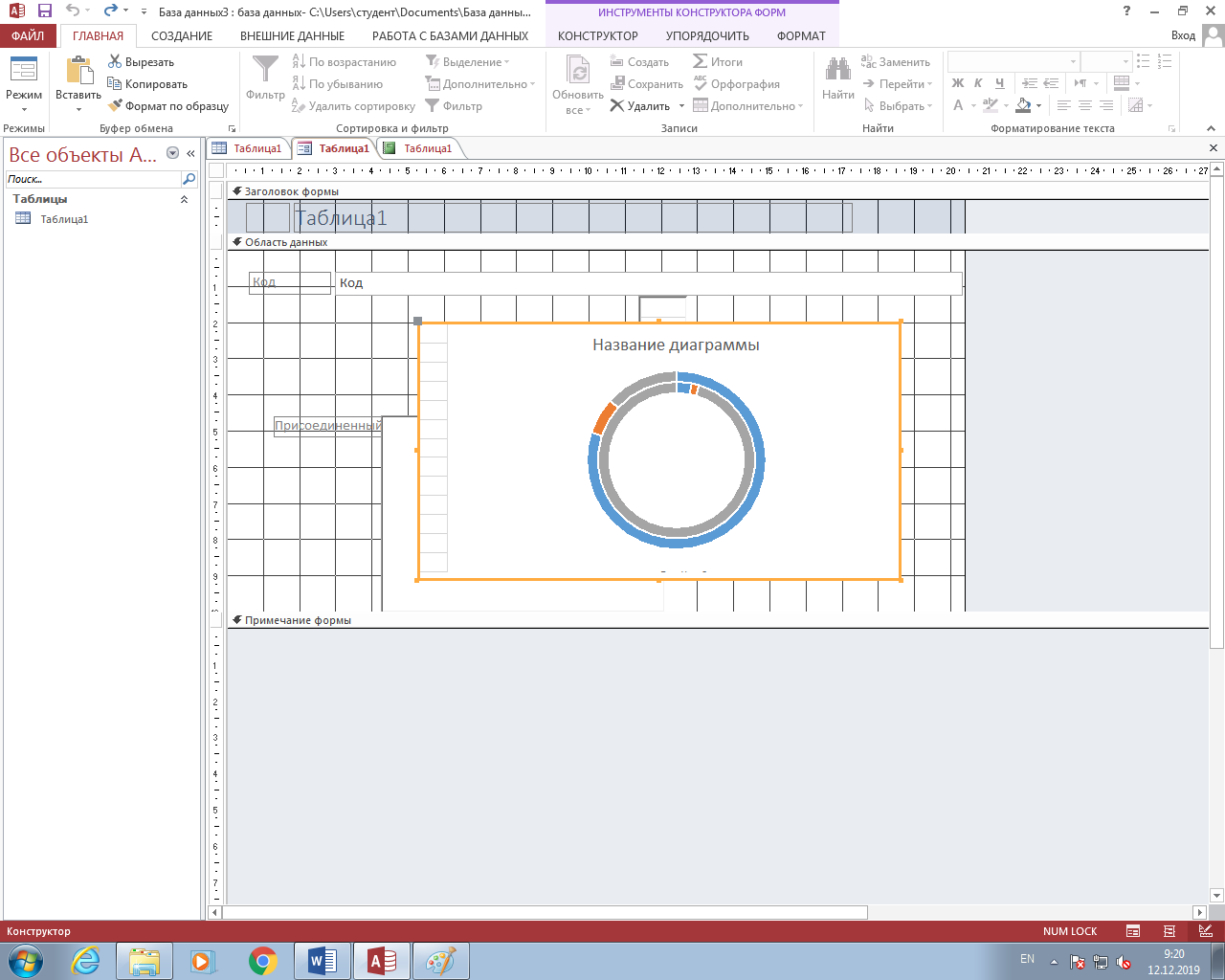


Рис. 14.15. Использование присоединенной рамки объекта для вставки листов Microsoft Excel в отчет Microsoft Access

# Вставка документов Word в таблицы, формы и отчеты Access

Использование документов Microsoft Word в таблицах, формах и отчетах Microsoft Access осуществляется аналогично использованию листов Microsoft Excel или рисунков. Документ Microsoft Word рассматривается как объект OLE, который может быть помещен в таблицу или, с помощью свободной или присоединенной рамки объекта, в форму или отчет Microsoft Access.

Рассмотрим пример, в котором Microsoft Access используется для учета документов Microsoft Word. Для реализации учета документов Word с возможностью их просмотра и редактирования необходимо создать таблицу, в которую должны быть включены поля для порядкового номера документа, названия документа, шифра документа, самого документа и примечаний (табл. 14.2).

Таблица 14.2. Поля таблицы и их типы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | |  | |  |
|  | **Имя поля** | | | | **Тип поля** | | **Назначение** | | |  |
|  | "Номер" | | | | Счетчик (AutoNumber) | | Порядковый номер документа | | |  |
|  | "Наименование" | | | | Текстовый (Text) | | Название документа | | |  |
|  | "Код" | | | | Текстовый (Text) | | Шифр документа | | |  |
|  | "Текст" | | | | Поле объекта OLE (OLE Object) | | Документ | | |  |
|  | "Примечание" | | | | Поле Memo (Memo) | | Примечания | | |  |
|  |  |  | |  | |  |

Для создания таблицы необходимо:

1. Создать в режиме Конструктора таблицу, состоящую из "указанных выше полей (рис. 14.16).

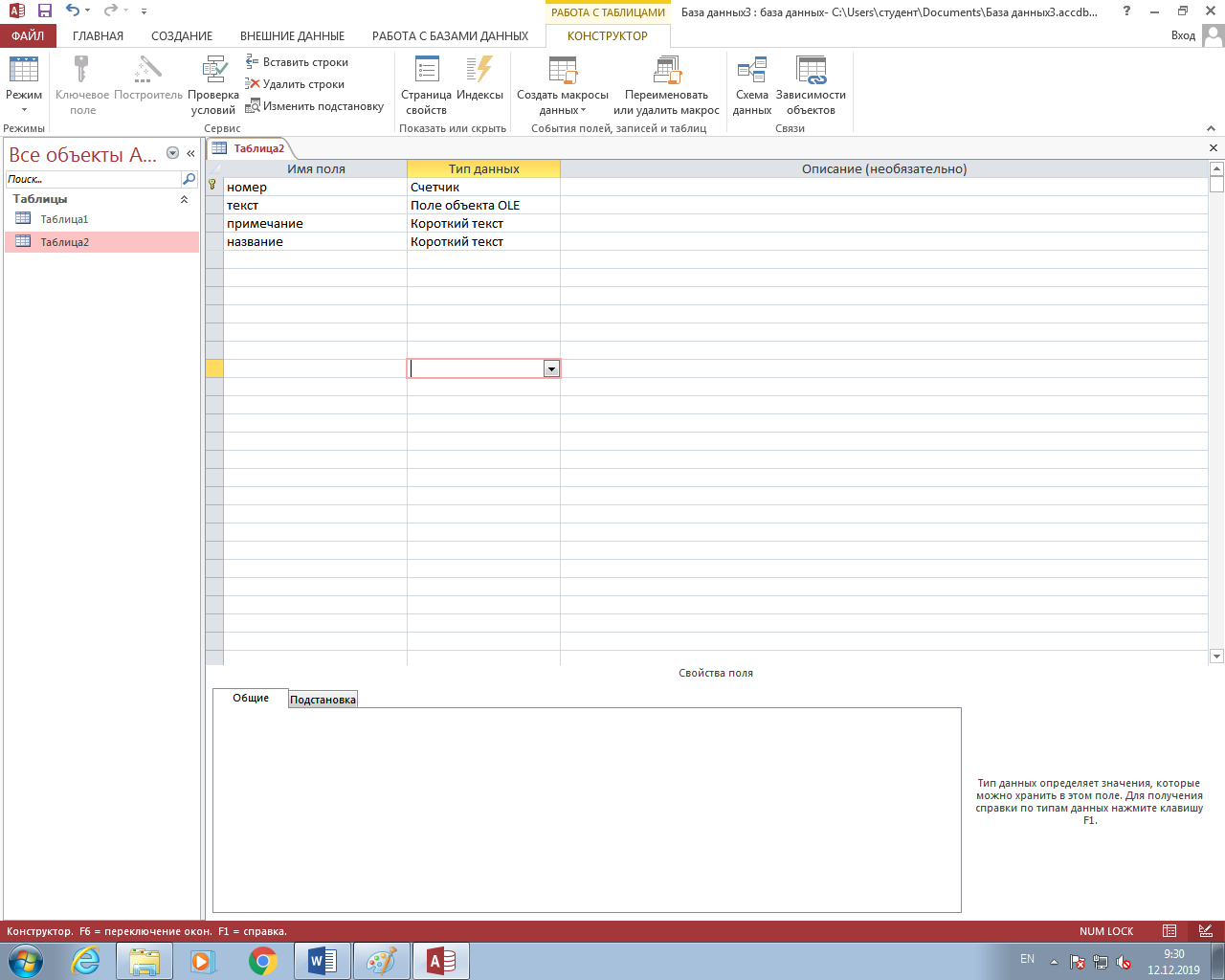


Рис. 14.16. Таблица "Учет документов" в окне Конструктора

1. Сохранить таблицу под именем "Учет документов".
2. Открыть таблицу "Учет документов" в режиме Таблицы.
3. Заполнить поля таблицы.

Для вставки документа Word в качестве объекта OLE требуется установить указатель на нужное поле, выполнить команду Объект (Object) из меню Вставка (Insert), в окне Microsoft Access выбрать переключатель Создать из файла (Create From File) и в поле Файл (File) указать имя и путь к файлу. Так как в нашем примере учитываются уже созданные документы, которые могут изменяться не только из Microsoft Access, необходимо установить связь с этими документами, для чего установите флажокСвязь (Link) в окне Microsoft Access. Флажок В виде значка (Dysplay As Icon) в данном случае не устанавливается, т. к. нужно иметь возможность быстрого просмотра документов (рис. 14.17).

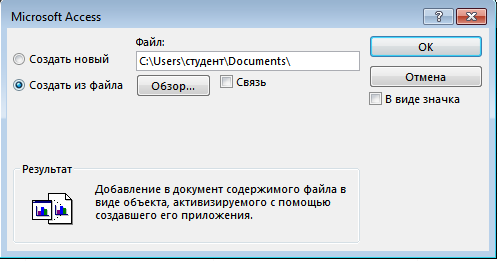


Рис. 14.17. Окно Microsoft Access при вставке документа Word из файла в таблицу в виде объекта OLE

В результате создана таблица "Учет документов", приведенная на рис. 14.18.

После создания таблицы, в которой содержатся данные о документах, необходимо создать форму для удобного просмотра, редактирования и добавления новой информации о документах. Чтобы создать форму, необходимо:

1. В окне базы данных раскрыть список форм и нажать на кнопку Создать (New).
2. В окне Новая форма (New Form) в поле со списком Выберите таблицу или запрос, на основе которого будет построена форма (Choose the table or query where the object's data comes from) выбрать таблицу "Учет документов", выбрать Конструктор (Design View) и нажать кнопку ОК.

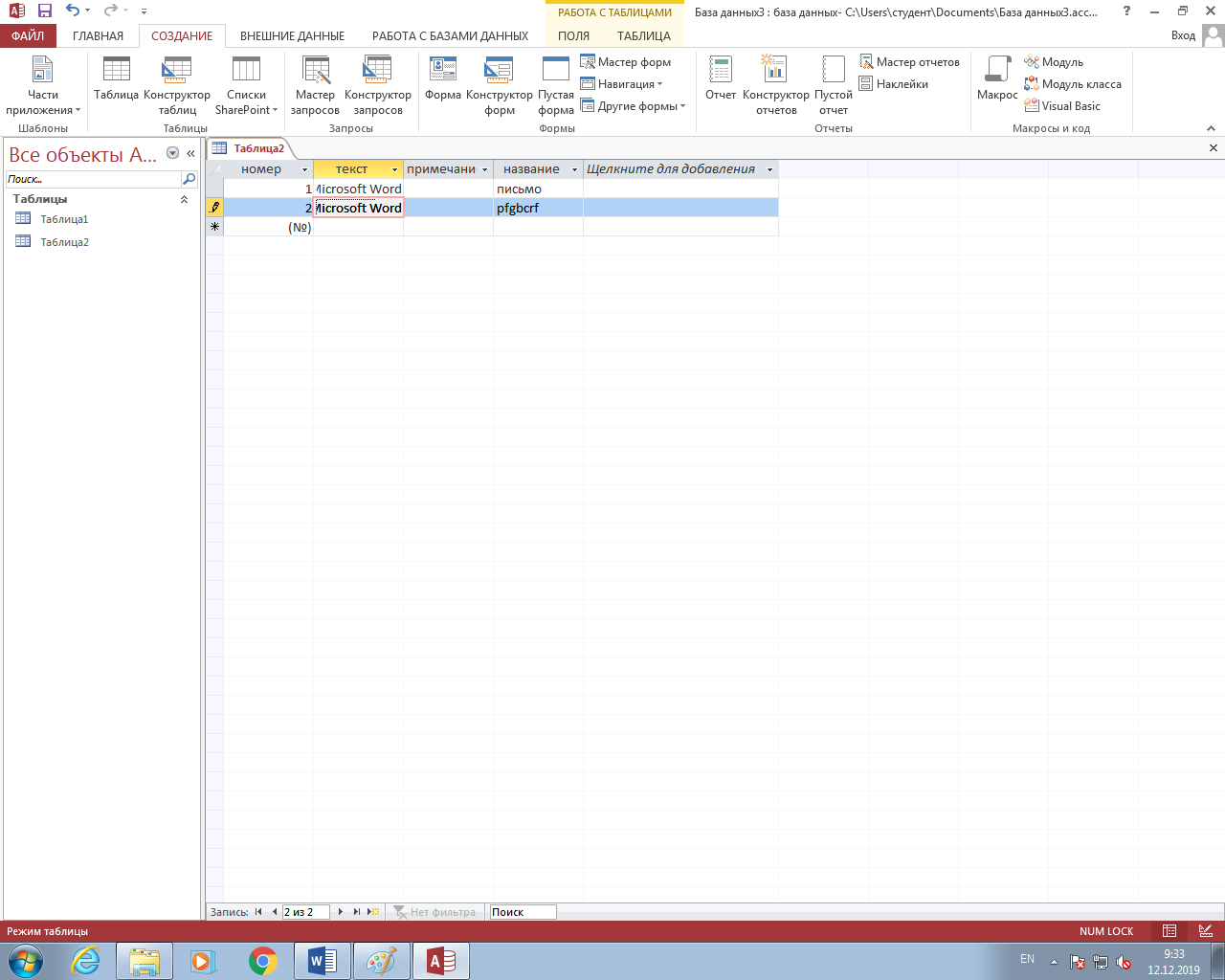


Рис. 14.18. Представление данных в таблице "Учет документов" в режиме Таблицы

1. Нажать кнопку Список полей (Field List) и перенести поля из списка в форму.
2. Отредактировать элементы в форме, изменив при необходимости их размеры и положение в форме. Пример расположения элементов приведен на рис. 14.19.
3. Закрыть форму, сохранив ее под именем "Учет документов".

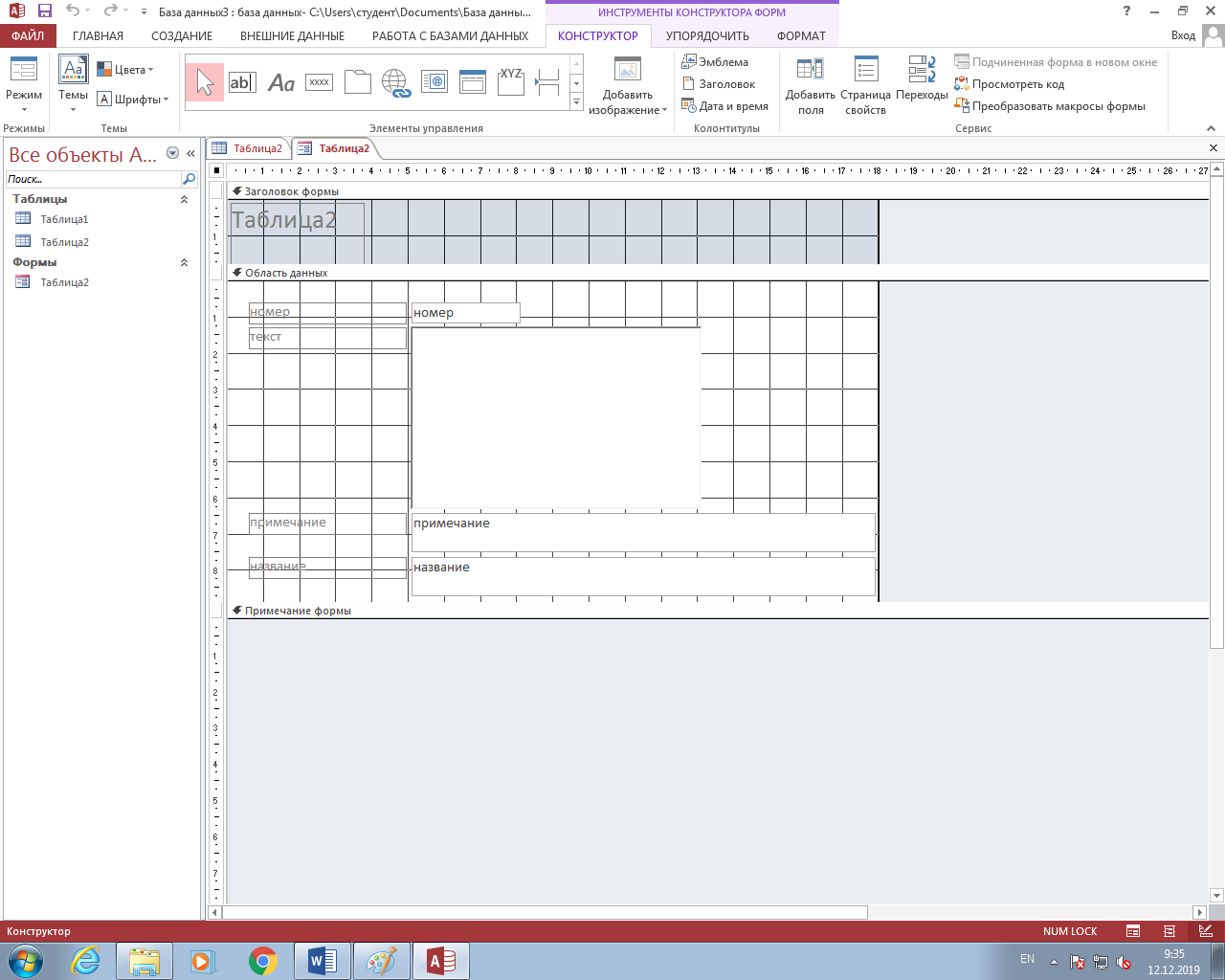


Рис. 14.19. Расположение элементов в форме "Учет документов"

Для просмотра содержимого таблицы необходимо открыть форму в режиме Формы (рис. 14.20). Чтобы получить возможность редактирования документа, требуется дважды щелкнуть левой кнопкой мыши на документе в форме. При этом откроется приложение Microsoft Word с выбранным документом.

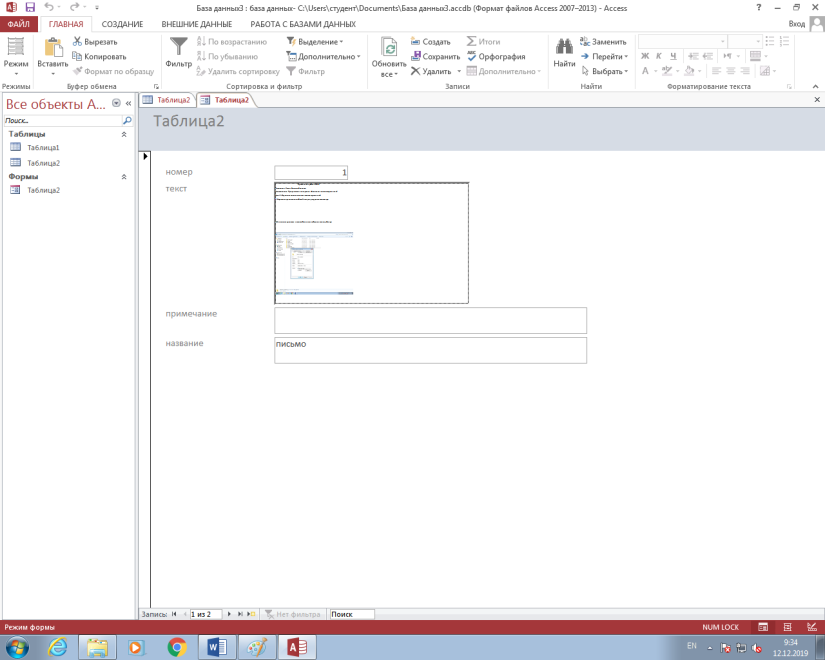


Рис. 14.20. Форма "Учет документов" в режиме Формы

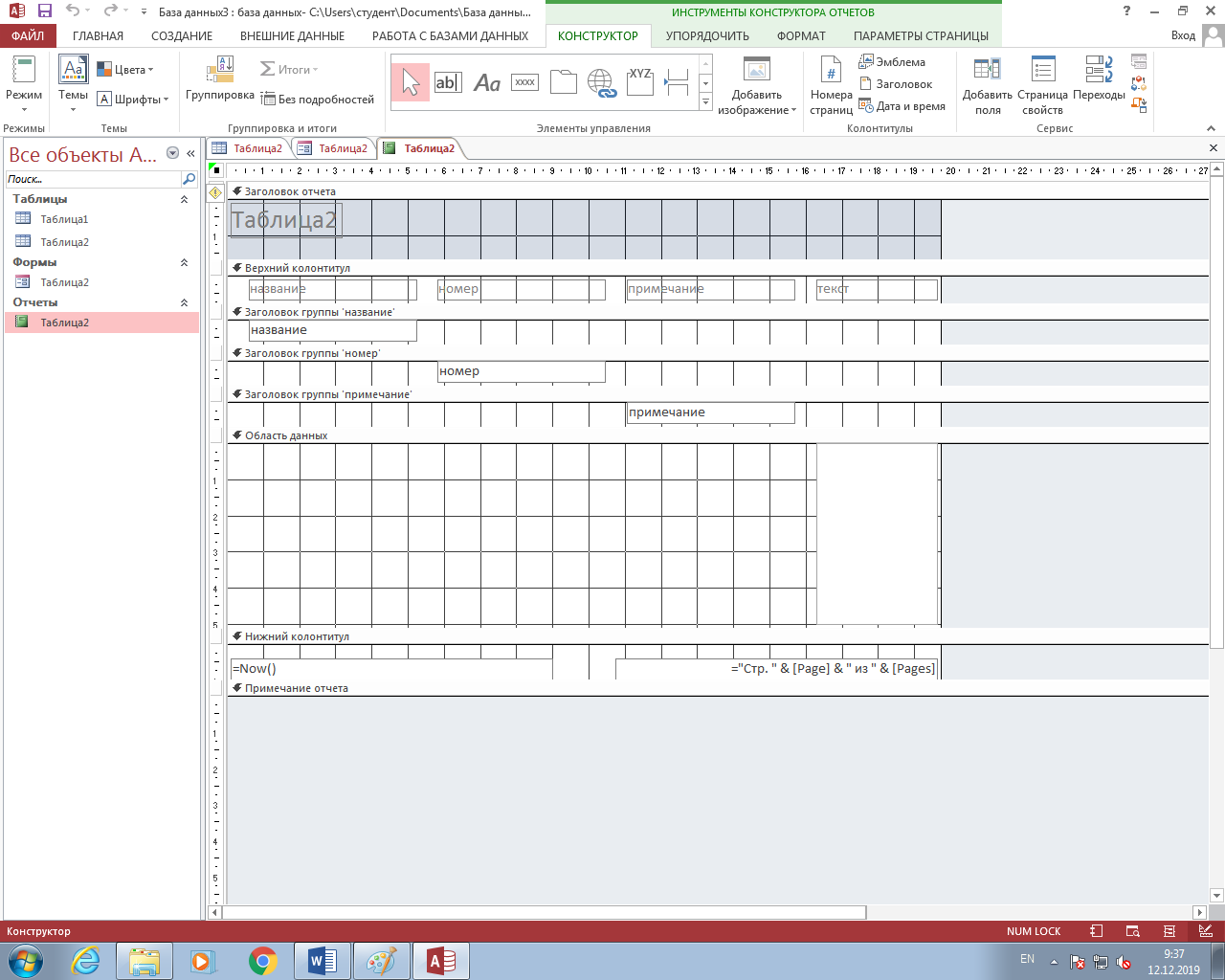


Рис. 14.21. Расположение элементов отчета "Учет документов" в режиме Конструктора

Для создания отчета необходимо:

1. В окне базы данных раскрыть список отчетов и нажать кнопку Создать (New).
2. В окне Новый отчет (New Report) в поле со списком Выберите таблицу или запрос, на основе которого будет построен отчет (Choose the table or query where the object's data comes from) выбрать таблицу "Учет документов", выбрать Конструктор (Design View) и нажать кнопку ОК.
3. Нажать кнопку Список полей (Field List) и перенести поля из списка в секцию Область данных (Detail) отчета.
4. Отредактировать элементы в отчете, изменив при необходимости их размеры и положение в форме. Пример расположения элементов приведен на рис. 14.21.
5. Закрыть отчет, сохранив его под именем "Учет документов".

Чтобы просмотреть содержимое отчета, необходимо открыть его в режиме просмотра (рис. 14.22).

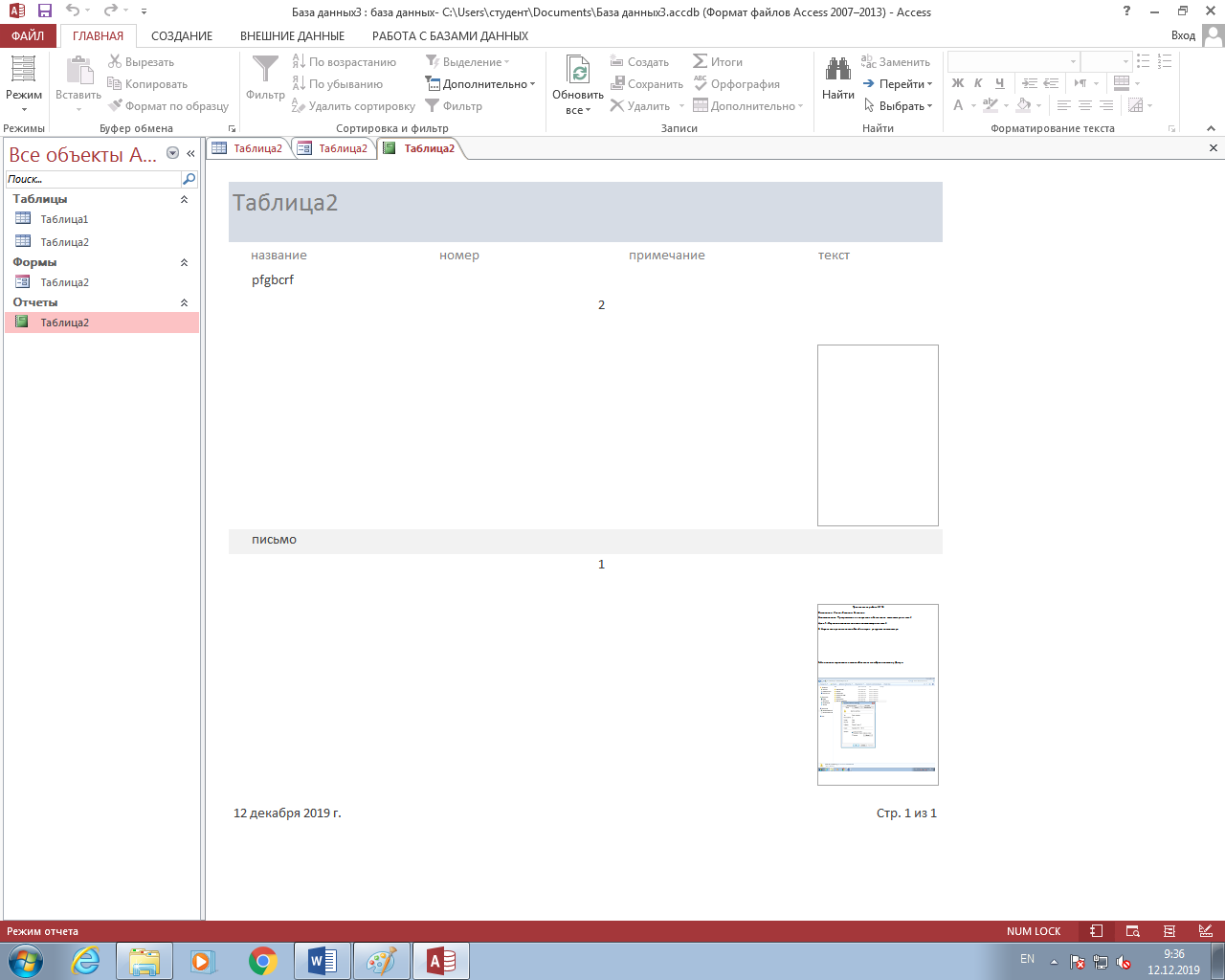


Рис. 14.22. Отчет "Учет документов" в режиме просмотра

При изменении содержимого документов Word эти изменения будут отображаться в форме и отчете при их следующем открытии.

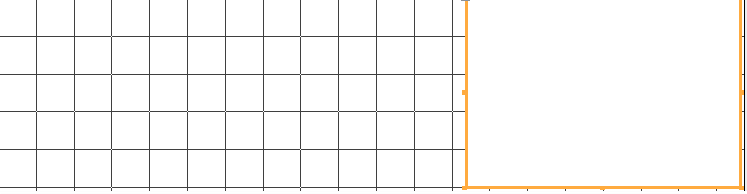


Рис. 14.23. Представление документа Word в отчете в виде значка

При желании можно вставлять документы Word в виде значка. Для этого в окне Вставка объекта(Insert Object) необходимо устанавливать флажок В виде значка (Display As Icon). В этом случае в форме и отчете не будут отображаться данные документов, а будут указываться только значки, изображающие документы. Для обращения к нужному документу из формы достаточно дважды щелкнуть левой кнопкой мыши на значке соответствующего документа. При этом откроется приложение Microsoft Word с соответствующим документом в окне. Примеры отображения документов в виде значка в форме и отчете приведены, соответственно, на рис. 14.23 и рис. 14.24.