## Министерство образования Иркутской области

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Ангарский политехнический техникум»

РАССМОТРЕНА

на заседании предметной (цикловой)

комиссии и рекомендована к применению

Председатель ПЦК

/О.Ю.Лезнова/

« » 20\_\_г.

# МЕТОДИЧЕСКАЯ РАЗРАБОТКА

## по выполнению внеаудиторной самостоятельной работы (СРС)

учебная дисциплина общепрофессионального цикла

**ОП.07. Основы проектирования баз данных**

по специальности **09.02.04 Информационные системы (по отраслям)**

Преподаватель Макарова Наталья Алексеевна,

высшая квалификационная категория

Ангарск 2018

## Содержание

1. Тематический план организации СРС
2. Методические рекомендации (указания) к выполнению каждой СРС

2

# Тематический план организации самостоятельной работы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Тема** | **Количество**  **часов** | **Цель самостоятельной**  **работы, включая ОК и ПК** | **Задания для самостоятельной работы** | **Дидактическое обеспечение и рекомендуемая литература (включая Интернет-ресурсы)** |
| 1 | Основы реляционной алгебры, решение задач. | 4 | Формирование умений и знаний потеме: «Тема 1.1  Основные  понятия баз данных. Взаимосвязи в моделях и реляционный подход к построению моделей»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Выполнение  индивидуальных  заданий | Федорова Г.Н. Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Н. Федорова. – М.: Издательский центр «Академия», 2018. – 224 с. |
| 2 | Установка и нормализация отношений в базе данных. | 6 | Формирование умений и знаний потеме: «Тема 1.1  Основные  понятия баз данных. Взаимосвязи в моделях и реляционный подход к построению моделей»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Выполнение  индивидуальных  заданий |
| 3 | Формирование запросов и отчетов. | 4 | Формирование умений и знаний по теме: «Тема 1.2  Применение языка VBA в разработке приложений»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Выполнение заданий по вариантам |
| 4 | Создание процедур обработки событий с применением языка VBA. | 6 | Формирование умений и знаний по теме: «Тема 1.2  Применение языка VBA в разработке приложений»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Выполнение заданий  по вариантам |
| 5 | Использование алиасов, агрегатных функций с группировками. | 8 | Формирование умений и знаний по теме: «Тема 2.1  Организация запросов SQL»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Выполнение  индивидуальных  заданий |
| 6 | Формирование запросов SQL, создание форм и отчетов. | 8 | Формирование умений и знаний по теме: «Тема 2.1  Организация запросов SQL»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Выполнение  индивидуальных  заданий |
| 7 | Создание и заполнение таблиц базы данных. | 8 | Формирование умений и знаний по теме: «Тема 3.2  Курсовое проектирование»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Создание и заполнение таблиц базы данных |
| 8 | Создание запросов. | 8 | Формирование умений и знаний по теме: «Тема 3.2  Курсовое проектирование»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Создание запросов |
| 9 | Создание форм. | 8 | Формирование умений и знаний по теме: «Тема 3.2  Курсовое проектирование»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Создание форм |
| 10 | Создание макросов для автоматизации работы с БД. | 7 | Формирование умений и знаний по теме: «Тема 3.2  Курсовое проектирование»  создание условий для формирования компетенций: ОК2, ОК4, ОК5, ОК 8, ОК 9 | Создание макросов для автоматизации работы с БД |

# 

### Методические рекомендации (указания) к выполнению самостоятельной работы №1

1. Тема самостоятельной работы: Основы реляционной алгебры, решение задач.
2. Количество часов: 4ч.
3. Задания для самостоятельной работы: Выполнение индивидуальных заданий
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 1.1 Основные понятия баз данных. Взаимосвязи в моделях и реляционный подход к построению моделей», создание условий для формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

1. Рекомендации преподавателя:

**Примеры использования реляционной алгебры для выражения словесных запросов в виде формулы**

Дана БД предприятия, состоящая из трех таблиц

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ПОСТАВЩИК** | |  |  |  |  | **ПД** |  |  |
| **ПНОМ** | **ПФАМ** | **СТАТУС** | **ГОРОД** |  |  | **ПНОМ** | **ДНОМ** | **ШТ** |
| П1 | Иванов | 20 | Воронеж |  |  | П1 | Д1 | 300 |
| П2 | Петров | 15 | Москва |  |  | П1 | Д2 | 200 |
| П3 | Сидоров | 10 | Москва |  |  | П1 | Д3 | 400 |
| П4 | Зайцев | 30 | Воронеж |  |  | П1 | Д4 | 200 |
| П5 | Волков | 20 | Киев |  |  | П1 | Д5 | 100 |
|  |  |  |  |  |  | П1 | Д6 | 100 |
| **ДЕТАЛЬ** | |  |  |  |  | П2 | Д1 | 300 |
| **ДНОМ** | **ДНАЗВ** | **ЦВЕТ** | **ВЕС** | **ГОРОД** |  | П2 | Д2 | 400 |
| Д1 | Гайка | Красный | 12 | Воронеж |  | П3 | Д3 | 200 |
| Д2 | Болт | Зеленый | 17 | Москва |  | П4 | Д2 | 200 |
| Д3 | Шайба | Голубой | 17 | Минск |  | П4 | Д4 | 300 |
| Д4 | Шайба | Красный | 14 | Воронеж |  | П4 | Д5 | 400 |
| Д5 | Шуруп | Голубой | 12 | Москва |  |  |  |  |
| Д6 | Гвоздь | Красный | 19 | Воронеж |  |  |  |  |

Построить выражения реляционной алгебры, которые позволят получить ответ на вопросы:

1. **Получить имена поставщиков, которые поставляют деталь Д2**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ПНОМ** | **ДНОМ** | **ШТ** | **ПФАМ** | **СТАТУС** | **ГОРОД** |
| П1 | Д1 | 300 | Иванов | 20 | Воронеж |
| П1 | Д2 | 200 | Иванов | 20 | Воронеж |
| П1 | Д3 | 400 | Иванов | 20 | Воронеж |
| П1 | Д4 | 200 | Иванов | 20 | Воронеж |
| П1 | Д5 | 100 | Иванов | 20 | Воронеж |
| П1 | Д6 | 100 | Иванов | 20 | Воронеж |
| П2 | Д1 | 300 | Петров | 15 | Москва |
| П2 | Д2 | 400 | Петров | 15 | Москва |
| П3 | Д3 | 200 | Сидоров | 10 | Москва |
| П4 | Д2 | 200 | Зайцев | 30 | Воронеж |
| П4 | Д4 | 300 | Зайцев | 30 | Воронеж |
| П4 | Д5 | 400 | Зайцев | 30 | Воронеж |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ПНОМ** | **ДНОМ** | **ШТ** | **ПФАМ** | **СТАТУС** | **ГОРОД** |
| П1 | Д2 | 200 | Иванов | 20 | Воронеж |
| П2 | Д2 | 400 | Петров | 15 | Москва |
| П4 | Д2 | 200 | Зайцев | 30 | Воронеж |



|  |
| --- |
| **ПФАМ** |
| Иванов |
| Петров |
| Зайцев |



Выражение:



1. **Получить имена поставщиков, которые поставляют по крайней мере одну красную деталь**





1. **Получить имена поставщиков, которые поставляют все детали**





1. **Получить номера поставщиков, поставляющих по крайней мере все те детали, которые поставляет поставщик П2**



1. **Получить имена поставщиков, которые не поставляют деталь Д2**





**Задания для самостоятельной работы**

По словесному запросу к базе данных предприятий составить соответствующее выражение в терминах реляционной алгебры, необходимое для этого запроса.

Дана БД агентства недвижимости, состоящая из четырех таблиц.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Отношение Владелец** | |  |  |  |
| **НомерВлад** | **ФИО** | **Улица** | **Дом, Кв** | **Тел** |
| 11 | Николаев И.П. | ул.Комсомольская | д.165, кв 56 | 42-26-58 |
| 12 | Антонов А.П. | ул.Ленина | д.1а | 25-78-96 |
| 21 | ООО "Весна" | пр.Победы | д.80, кв.1 | 73-24-15 |
| 22 | ЗАО "Персей" | Ул.Горького | д.25б | 55-66-59 |

НомерВлад – номер владельца недвижимости;

ФИО – Фамилия, имя, отчество либо наименование владельца;

Улица, Дом,Кв – место жительства либо расположения владельца;

Тел – телефон.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Отношение Объект** | | |  |  |  |  |
| **НомерОб** | **Тип** | **Улица** | **Дом, Кв** | **КолКом** | **РазмерПлаты** | **НомерВлад** |
| 105 | дом | ул. Железнодорожная | д.76 | 3 | 200 | 11 |
| 107 | дом | ул. Комсомольская | д.2 | 3 | 185 | 12 |
| 108 | дом | ул.Кирпичная | д.15 | 2 | 150 | 21 |
| 15 | кв-ра | ул.Комсомольская | д.157, кв.12 | 2 | 450 | 22 |
| 14 | кв-ра | пр.Мира | д.17, кв.22 | 1 | 120 | 11 |
| 16 | кв-ра | Пр.Мира | д.122, кв.11 | 1 | 200 | 22 |
| 17 | кв-ра | ул.Ленина | д.456,кв.5 | 1 | 200 | 12 |
| 18 | кв-ра | ул.А.Буюклы | д.25, кв.13 | 3 | 600 | 12 |
| 19 | кв-ра | ул.Чехова | д.68, кв.23 | 2 | 400 | 11 |
| 20 | кв-ра | ул.Горького | д.17, кв.135 | 1 | 250 | 21 |
| 110 | дом | ул.Колхозная | д.12 | 2 | 150 | 21 |
| 111 | дом | ул.Колхозная | д.22 | 2 | 180 | 11 |
| 112 | дом | ул.Лермонтова | д.16 | 3 | 300 | 12 |

НомерОб – номер объекта недвижимости;

Улица, Дом,Кв – место расположения объекта недвижимости;

Тип – дом, кв-ра, коттедж или другой вид жилья;

КолКом – количество комнат;

РазмерПлаты – арендная плата за месяц ;

НомерВлад – номер владельца, которому принадлежит объект.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Отношение Клиент** | |  |  |  |  |  |
| **НомКлиента** | **Имя** | **Улица** | **Дом, Кв** | **Тел** | **ПредпочтТип** | **МаксПлата** |
| 13-03 | Петров В.И | ул. Украинская | д.67, кв.69 | 73-15-69 | 1-к | 120 |
| 17-02 | Васильев Н.А | ул.Бумажная | д.25 | 55-68-96 | 2-к | 250 |
| 01-02 | Ветров Г.Б. | пр.Мира | д.19, кв.13 | 71-11-87 | дом | 150 |
| 22-03 | Долгова Н.Н. | пр.Мира | д.193, кв.58 | 42-63-78 | 1-к | 100 |
| 27-02 | Перова Е.И | ул.Ленина | д.125, кв.69 | 73-56-42 | 2-к | 200 |
| 17-03 | Антонова Н.А | ул.Бумажная | д.52 | 42-12-45 | 2-к | 250 |
| 18-02 | Степанов Г.Б. | пр.Мира | д.369,кв.95 | 25-32-12 | 2-к | 200 |

НомерКлиента – номер клиента;

Имя– имя клиента;

Улица, Дом,Кв – место жительства клиента;

Тел – телефон;

ПредпочтТип – пожелание клиента относительно количества комнат;

МаксПлата – пожелания клиента по арендной плате за месяц ;

**Отношение Договор**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **НомДог** | **ДатаПодпис** | **СрокДейств** | **НомерОб** | **НомКлиента** | **Комиссионные** |
| Д-01-2004-1 | 17.01.2004 | 25.04.2004 | 14 | 13-03 | 36,00 |
| Д-01-2004-2 | 18.01.2004 | 18.07.2004 | 108 | 17-03 | 90,00 |
| Д-01-2004-3 | 25.01.2004 | 25.04.2004 | 107 | 27-02 | 57,00 |
| Д-04-2004-15 | 26.04.2004 | 26.07.2004 | 14 | 22-03 | 36,00 |
| Д-02-2004-4 | 02.02.2004 | 02.02.2005 | 110 | 18-02 | 180,00 |
| Д-02-2004-5 | 05.02.2004 | 05.09.2004 | 111 | 18-02 | 126,00 |
| Д-04-2004-16 | 25.04.2004 | 25.10.2004 | 107 | 27-02 | 114,00 |
| Д-02-2004-6 | 16.02.2004 | 16.09.2004 | 20 | 17-03 | 175,00 |
| Д-02-2004-7 | 21.02.2004 | 21.05.2004 | 16 | 17-02 | 60,00 |
| Д-02-2004-8 | 29.02.2004 | 29.08.2004 | 17 | 17-03 | 120,00 |
| Д-03-2004-9 | 01.03.2004 | 01.09.2004 | 105 | 27-02 | 120,00 |
| Д-03-2004-10 | 03.03.2004 | 03.06.2004 | 112 | 17-02 | 90,00 |

НомДог – номер договора;

ДатаПодпис – дата подписания договора;

СрокДейств – срок действия договора;

НомерОб – номер объекта недвижимости;

НомерКлиента – номер клиента;

Комиссионные – плата агентству за оформление сделки.

**Вариант1**

1. Составить список всех домов.
2. Получить список клиентов, арендующих объект с номером 14.
3. Определить тех клиентов, которые арендуют такие же по типу объекты недвижимости, что и клиент Долгова Н.Н.
4. Определить адреса квартир, при заключении договоров по которым комиссионные составили больше 100у.е.
5. Получить номера объектов, арендная плата по которым составляет не более 150 у.е. или которые расположены на той же улице, где живет Ветров Г.Б.

**Вариант2**

1. Перечислить все 1-к квартиры, арендная плата по которым меньше 200у.е.
2. Выбрать адреса домов, которые бы устроили клиента Ветрова Г.Б.
3. Составить список номеров объектов, по которым не было заключено ни одной сделки.
4. Выбрать информацию по договорам, заключенным с клиентом Антоновой Н.А.
5. Вывести всю информацию по договорам, включая имя клиента, адрес, тип жилья.

**Вариант3**

1. Перечислить имена и адреса клиентов.
2. Составить список владельцев, заключивших договора в марте 2004года.
3. Определить владельцев, которые предлагают для аренды такие же объекты, как и владелец с номером 11.
4. Выбрать объекты, расположенные на той же улице, где живут их владельцы.
5. Вывести список типов жилья с указанием их стоимости.

**Вариант4**

1. Перечислить номера объектов, арендованных до 01.01.2005г
2. Составить список владельцев, которые являются собственниками домов.
3. Вывести список владельцев, являющихся собственниками и дома и квартиры.
4. Вывести список клиентов и объектов, находящихся на одной улице.
5. Определить владельца объекта, для которого заключен договор с размером комиссионных 180 у.е.

**Вариант5**

1. Составить список адресов объектов недвижимости, находящихся в собственности владельца с номером 21.
2. Выбрать фамилии людей, которым подойдет жилье по адресу пр.Мира, 17-22.
3. Получить номера объектов недвижимости, размер арендной платы которых равен максимально возможной плате клиента с номером 18-02.
4. Определить имена владельцев, заключивших хотя бы одну сделку.
5. Вывести номера объектов, принадлежащих владельцу Николаеву И.П., размер платы по которым не меньше 200у.е.

Дана БД производственного предприятия, состоящая из четырех таблиц.

|  |  |  |  |
| --- | --- | --- | --- |
| **ПРЕДПРИЯТИЕ** | |  |  |
| **Пред#** | **Название** | **Рейтинг** | **Город** |
| 180 | Электроника | 230 | Воронеж |
| 230 | Гормолзавод | 300 | Москва |
| 150 | Сельмаш | 140 | Воронеж |
| 190 | Хлебозавод | 300 | Курск |
| 270 | Рудгормаш | 240 | Москва |

где

Пред# – номер предприятия, номер общий по некоторым группам городов;

Название – название предприятия;

Рейтинг – рейтинг предприятия по некоторым показателям;

Город – город, в котором находится предприятие.

|  |  |  |  |
| --- | --- | --- | --- |
| **ПРОДУКЦИЯ** | |  |  |
| **Прод#** | **Наименование** | **Количество** | **ГородВыпуска** |
| 10 | Магнитофоны | 12000 | Воронеж |
| 20 | Кровати | 15000 | Москва |
| 30 | Тракторы | 20000 | Воронеж |
| 40 | Кухни | 30000 | Орел |
| 50 | Продукты | 10000 | Воронеж |

где

Прод# – номер продукции;

Наименование – наименование продукции;

Количество – стоимость продукции, выпускаемой в год в данном городе;

ГородВыпуска – город, в котором указанная продукция выпускается.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Работник** | |  |  |  |
| **ТН** | **Фамилия** | **ГородПрожив** | **День\_рожд** | **Пред#** |
| 55 | Иванов | Воронеж | 15.03.02 | 180 |
| 10 | Петров | Москва | 17.02.95 | 230 |
| 100 | Сидоров | Воронеж | 03.12.93 | 150 |
| 190 | Иванов | Курск | 18.04.91 | 190 |

где

ТН – номер личности;

Фамилия – фамилия человека;

ГородПрожив – город проживания;

День\_рожд – дата рождения данного человека;

Пред# – номер предприятия, где работает данная личность.

|  |  |  |  |
| --- | --- | --- | --- |
| **ПРЕД\_ПРОД** | | | |
| **Пред#** | **Прод#** | **Год** | **Выработка** |
| 150 | 30 | 2000 | 150 |
| 180 | 10 | 2000 | 100 |
| 190 | 50 | 2001 | 50 |
| 230 | 50 | 2001 | 120 |
| 270 | 20 | 2002 | 50 |

где

Пред# – номер предприятия;

Прод# – номер продукции;

Год – год выпуска продукции;

Выработка (тыс.руб) – количество продукции данного предприятия.

## 

**Вариант6**

1. Получить названия предприятий, производящих продукцию с номером 30.
2. Выбрать информацию обо всех предприятиях, в т.ч. о работниках и продукции.
3. Выбрать фамилии людей, которые работают на хлебозаводе.
4. Определить номера предприятий из Воронежа с рейтингом выше 200.
5. Выбрать имена предприятий, производящих все виды продукции.

**Вариант7**

1. Получить имена предприятий, производящих продукцию всех сортов.
2. Выбрать название продукции, у которой количество потребления в городе находится в диапазоне от 12000 до 15000.
3. Выбрать фамилии людей, у которых город проживания совпадает с городом нахождения предприятия.
4. Найти номера работников, работающих на одном предприятии.
5. Определить название предприятий, которые не производят продукцию с номером 50.

**Вариант8**

1. Получить номера предприятий, производящих по крайней мере ту продукцию, которую выпускает предприятие с номером 190.
2. Выбрать название предприятий, у которых выработка продукции в 2001 г. на единицу работающего составила более 100 тыс. руб.
3. Определить фамилии людей, работающих на предприятиях в г.Воронеже.
4. Определить имена предприятий, производящих продукцию с номером «10».
5. Определить номера предприятий, производящих по крайней мере все виды продукции, производимые предприятием с номером 270.

**Вариант9**

1. Выбрать все пары названий городов, для которых предприятие и работники находятся в одном городе.
2. Выбрать название предприятий, которые производят продукты.
3. Определить название предприятий, производящих продукцию с номером «50» в 2001 году.
4. Определить номера предприятий, имеющих в списке работающих по крайней мере одного «Иванова».
5. Получить номера продукции, которая имеет количество более 15000 или производится предприятием с номером 270.

**Вариант10**

1. Получить имена предприятий, не производящих продукцию с номером 50.
2. Выбрать названия городов, для которых предприятие из первого города, а интересующая продукция во втором городе.
3. Определить название продукции с номером «30», имеющей выработку на единицу работающего > 100 тыс. руб.
4. Найти названия предприятий, производящих по крайней мере одну продукцию с номером 50.
5. Найти названия предприятий, выпускающих одинаковую продукцию.
6. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.

### Методические рекомендации (указания) к выполнению самостоятельной работы №2

1. Тема самостоятельной работы: Установка и нормализация отношений в базе данных.
2. Количество часов: 6ч.
3. Задания для самостоятельной работы: Выполнение индивидуальных заданий
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 1.1 Основные понятия баз данных. Взаимосвязи в моделях и реляционный подход к построению моделей», создание условий для формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

1. Рекомендации преподавателя:

**Краткие теоретические сведения**

Прежде чем приступить к созданию таких объектов базы данных, как таблицы, формы и отчеты, нужно разработать их проект. Главное назначение проекта — выработка четкого пути, по которому нужно следовать при его реализации. База данных–достаточно сложный объект, и время, затраченное на ее планирование, может значительно сократить сроки ее разработки. Отсутствиепродуманнойструктурыбазыданныхприводиткнеобходимостипостояннойпеределкииперенастраиваниюобъектовбазыданных, таких, как формы и таблицы.

Проектированиебазыданныхцелесообразноначатьскраткогоописанияотчетов, списков и других документов, которые необходимо получить с помощью БД. Далее следует разработать эскиз объектов, требуемых для получения необходимых результатов и определить связи между этими объектами.

При разработке эскиза необходимо ответить на следующие вопросы:

Какими данными мы располагаем?

Какие данные будут содержать таблицы?

Какой тип и какие свойства должны иметь данные в каждом поле таблицы?

Как эти таблицы будут связаны друг с другом?

Законченный план должен содержать подробное описание всех таблиц (имена полей, типы данных и их свойства), а также связей между ними.

Проектирование предусматривает этапы создания проекта базы данных от концепции до реального воплощения. Этапы проектирования базы данных:

1. Исследование предметной области и формулировка основных допущений (накладываемых условий). На этом этапе составляется список всех форм и отчетов, которые могут быть затребованы пользователями вашей БД.

2. Анализ данных. Составить перечень всех элементов данных, входящих в формы и отчеты и сгруппировать их в таблицы БД.

3. Установить, какие взаимосвязи существуют между элементами данных. Определить первичные и вторичные (внешние) ключи отношений. Организовать поля данных в таблицах, причем это необходимо сделать, следуя 4-м правилам нормализации:

Правило 1: Каждое поле таблицы должно представлять уникальный тип информации. Это правило означает, что необходимо избавиться от повторяющихся полей и разделить составные поля на отдельные элементы данных.

Правило 2: Каждая таблица должна иметь уникальный идентификатор или первичный ключ, который может состоять из одного или нескольких полей.

Правило 3: В таблице не должно быть данных, не относящихся к объекту, определяемому первичным ключом.

Правило 4: Независимость полей. Это правило означает возможность изменять значения любого поля (не входящего в первичный ключ) без воздействия на данные других полей.

Результатом 3 этапа должна явиться группа таблиц, удовлетворяющих правилам нормализации. На этом же этапе необходимо установить связи между таблицами.

**Порядок выполнения работы**

В качестве предметной области выбрана деятельность фирмы по продаже автомобилей марки Toyota. Поставлена задача: упорядочить информацию о клиентах фирмы, ассортименте продукции и сформированных заказах.

Этапы 1-3 проектирования БД изучить теоретически, 4-5 выполнить практически.

Оформить отчет о выполнении всех этапов задания

Ответить на контрольные вопросы

**1-й этап. Определение цели проектирования БД.**

Проектируемая реляционная БД должна содержать структурированную информацию о клиентах, продаваемых автомобилях и заказах.

Проектируемая БД должна поддерживать выполнение, как минимум, следующих основных функций: ввод и обновление информации, просмотр и удаление.

**2-й этап. Разработка информационно-логической модели предметной области.**

Вся информация о предметной области может быть логично разделена на 3 таблицы:

Клиенты, Автомобили, Заказы.

При этом выполняются основные требования к содержанию таблиц:

1. Каждая таблица содержит информацию только на одну тему.
2. Информация в таблицах не дублируется.
3. Для связи между таблицами заданы **первичные ключи**, однозначно определяющие каждую запись в таблице.

Содержание базовых таблиц приведено ниже:

|  |  |  |
| --- | --- | --- |
| **Таблица Клиенты** | **Таблица Автомобили** | **Таблица Заказы** |
| 1. Код клиента (ключ) | 1. Код модели (ключ) | 1. Код заказа (ключ) |
| 1. Фамилия | 1. Модель | 1. Код клиента |
| 1. Имя | 1. Мощность двигателя | 1. Код Модели |
| 1. Отчество | 1. Цвет | 1. Дата заказа |
| 1. Адрес | 1. Количество дверей | 1. Скидка, % |
| 1. Телефон | 1. Заводская цена | 1. Оплачено |
|  | 1. Издержки (транспортные, предпродажные) |  |
|  | 1. Специальная модель |  |
|  | 1. Дополнительное оснащение |  |

При разработке полей для каждой таблицы необходимо учитывать:

· Каждое поле должно быть связано с темой таблицы.

· Не включать в таблицу данные, которые являются результатом вычисления.

· Информацию следует разбивать на наименьшие логические единицы (Например, поля «Индекс», «Страна», «Населенный пункт», «Почтовый адрес», а не общее поле «Адрес»).

**3-й этап. Определение отношений между таблицами.**

Поскольку для проектируемой БД выполнены требования нормализации, между таблицами **Клиенты-Заказы** и **Автомобили-Заказы** могут быть установлены **одно-многозначные отношения***(1 : ∞ )*, которые поддерживаются реляционной СУБД.

**Связь** между таблицами устанавливается **с помощью ключей** Код клиента и Код модели, которые в главных таблицах Клиенты и Автомобили являются первичными, а в таблице-связке Заказы - внешними.



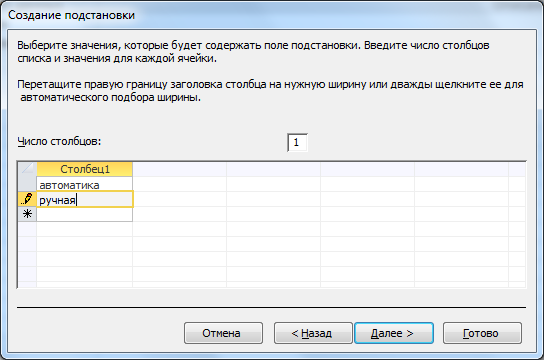
**4-й этап. Создание таблиц БД средствами СУБД MS Access.**

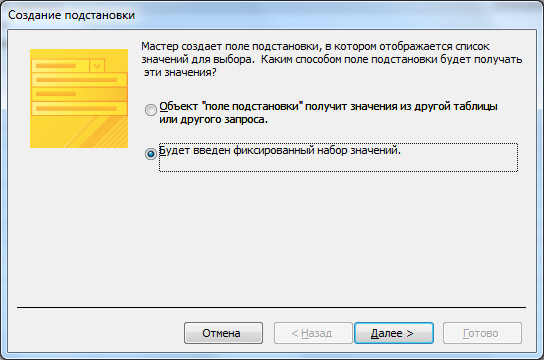
4.1. Загрузить СУБД MS Access. Создать **Новую базу данных.**

4.2. Создать макет таблицы Автомобили в режиме Конструктора, используя нижеприведенные данные об именах полей, их свойствах и типах данных.

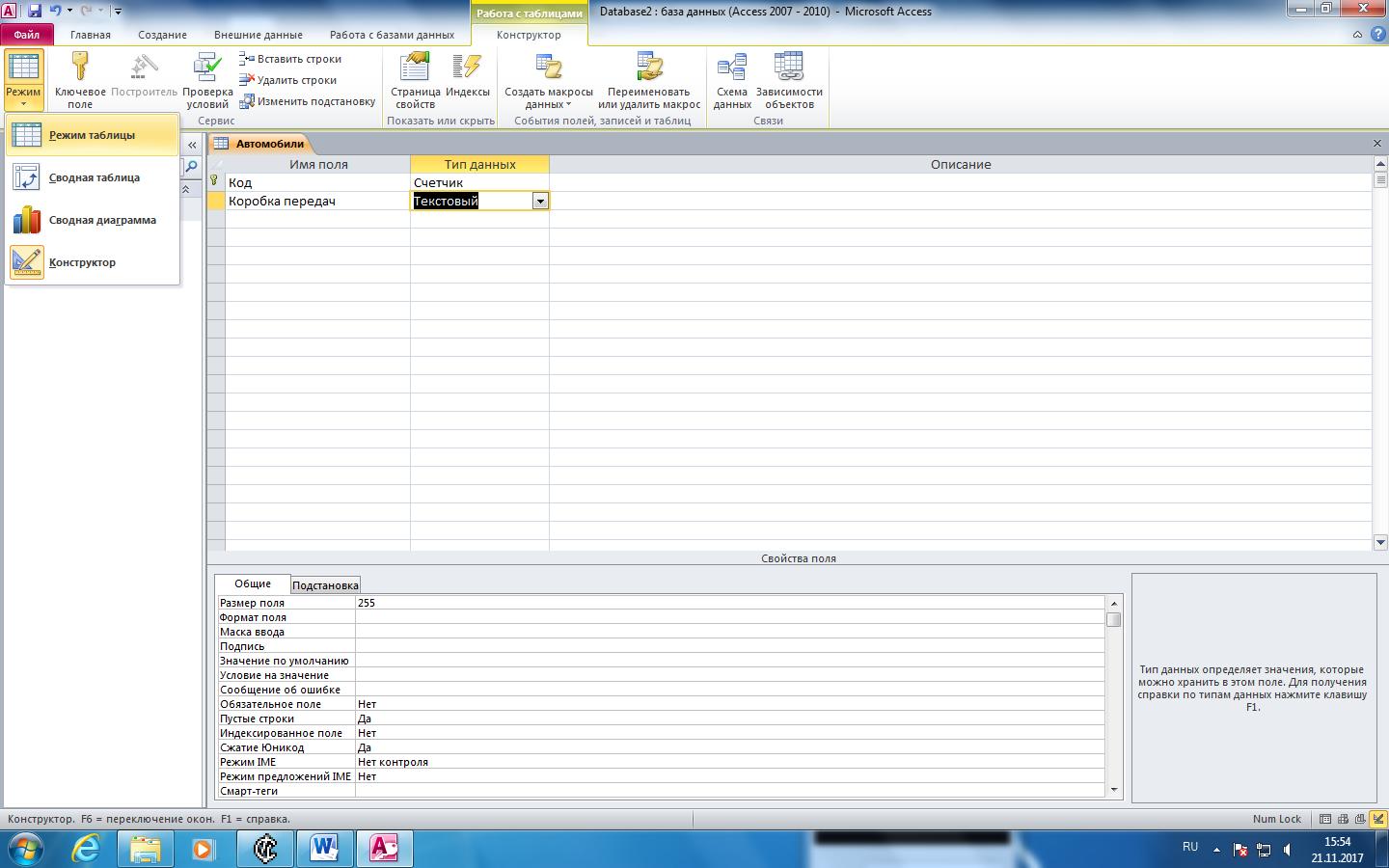
\*Все поля, за исключением поля Другое оснащение, должны быть обязательными для заполнения (Свойство Обязательное поле: Да).

| **Имя поля** | **Тип данных** | **Описание** | **Свойства поля** |
| --- | --- | --- | --- |
| **Код модели** | Числовой | Ключевое поле, код модели по заводскому каталогу | Индексированное поле: Да/Совпадения не допускаются  *Ключевое поле задается в меню Правка/Ключевое поле* |
| **Модель** | Текст | Тип кузова | Размер поля: 20,  Значение по умолчанию: Corolla  Индексированное поле: Да/Совпадения допускаются (одна и та же модель может встречаться в БД многократно с различными вариантами оснащения) |
| **Мощность** | Текст | Мощность двигателя (кВт/л.с.) | Размер поля: 10  Индексированное поле: Нет |
| **Цвет** | Текст | Цвет кузова | Размер поля: 20  Индексированное поле: Нет |
| **Количество дверей** | Числовой | 2 или 4 | Размер поля: Байт  Формат: Основной  Число десятичных знаков: 0  Значение по умолчанию: 4  Условие на значение: 2 Or 4  Сообщение об ошибке: Данное поле может содержать только 2 или 4  Индексированное поле: Нет |
| **Коробка передач** | \*Мастер подстановок, фиксированный набор значений | Автоматика или ручная | Размер поля: 12  Значение по умолчанию: Ручная  Условие на значение: “Ручная” Or “Автоматика”  Сообщение об ошибке: Допустимы только значения “Ручная” или “Автоматика”  Индексированное поле: Нет |
| **Обивка** | \*Мастер подстановок, фиксированный набор значений | Велюр, кожа, ткань | Размер поля: 10  Индексированное поле: Нет |
| **Другое оснащение** | Memo | Дополнительные аксессуары | Значение по умолчанию: Радио/плейер, раздвижная крыша |
| **Заводская цена** | Денежный | Заводская продажная нетто-цена | Формат: Денежный  Число десятичных знаков: Auto  Индексированное поле: Нет |
| **Транспортные издержки** | Денежный | Издержки на доставку | Формат: Денежный  Число десятичных знаков: Auto  Индексированное поле: Нет |
| **Предпродажные издержки** | Денежный | Издержки на предпродажную подготовку | Формат: Денежный  Число десятичных знаков: Auto  Значение по умолчанию: 105  Индексированное поле: Нет |
| **Специальная модель** | Логический | Спец. модель или стандартная | Формат: Да/Нет |

\*Мастер подстановокпозволяет создать для указанных полей список, из которого выбирается значение.



4.3. Перейти в режим Таблицы, сохранив созданный макет таблицы под именем Автомобили



4.4. Добавить в таблицу Автомобили 3 записи:

|  |  |  |  |
| --- | --- | --- | --- |
| **Код модели** | 12580 | 12653 | 12651 |
| **Модель** | Corolla Liftback | Corolla CompactGT | Corolla CompactXL |
| **Мощность** | 69/90 | 100/139 | 90/135 |
| **Цвет** | Бутылочное стекло | Черный | Небесно-голубой |
| **Количество дверей** | 4 | 2 | 2 |
| **Коробка передач** | Автоматика | Ручная | Ручная |
| **Обивка** | Ткань | Кожа | Велюр |
| **Другое оснащение** | Радио/плейер, раздвижная крыша, лаковое покрытие “Металлик” | Радио/плейер, раздвижная крыша, алюмин. дворники | Электро-подъемник окон, раздвижная крыша |
| **Заводская цена** | 39200 | 41100 | 37900 |
| **Транспортные издержки** | 1200 | 975 | 1050 |
| **Предпродажные издержки** | 105 | 105 | 105 |
| **Специальная модель** | Нет | Да | Да |

4.5. Создать макет таблицы Клиенты в режиме Конструктора.

\*Обязательные поля Код клиента, Фамилия, Страна.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Описание** | **Свойства поля** (определяют правила сохранения, отображения и обработки данных в поле) |
| **Код клиента** | Счетчик | Ключевое поле, уникальный номер клиента в БД | Индексированное поле: Да/Совпадения не допускаются  *Ключевое поле задается в меню Правка/Ключевое поле* |
| **Фамилия** | Текст | Фамилия | Размер поля: 40,  Индексированное поле: Да/Совпадения допускаются |
| **Имя** | Текст | Имя | Размер поля: 20,  Индексированное поле: Да/Совпадения допускаются |
| **Отчество** | Текст | Отчество | Размер поля: 40,  Индексированное поле: Да/Совпадения допускаются |
| **Индекс** | Числовой | Почтовый индекс | Размер поля: Длинное целое,  Индексированное поле: Да/Совпадения допускаются |
| **Страна** | Текст | Название страны | Размер поля: 20,  Индексированное поле: Да/Совпадения допускаются |
| **Населенный пункт** | Текст | Название населенного пункта | Размер поля: 40,  Индексированное поле: Да/Совпадения допускаются |
| **Почтовый адрес** | Текст | Почтовый адрес | Размер поля: 50,  Индексированное поле: Нет |
| **Телефон** | Текст | Контактный телефон | Размер поля: 20,  Индексированное поле: Нет |

4.6. Добавить в таблицу Клиенты 3 записи. (Перейти в режим Таблицы, сохранив макет таблицы под именем Клиенты)

4.7. Создать в режиме Конструктора макет таблицы Заказы.

\*Все поля, за исключением поля Скидка, являются обязательными для заполнения.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Описание** | **Свойства поля** (определяют правила сохранения, отображения и обработки данных в поле) |
| **Код заказа** | Счетчик | Ключевое поле, уникальный номер заказа | Индексированное поле: Да/Совпадения не допускаются  *Ключевое поле задается в меню Правка/Ключевое поле* |
| **Код модели** | Числовой, \*Мастер подстановок | Внешний ключ, для связи с таблицей Автомобили | Размер поля: Длинное целое  Индексированное поле: Да, допускаются совпадения |
| **Код клиента** | Числовой,  \*Мастер подстановок | Внешний ключ, для связи с таблицей Клиенты | Размер поля: Длинное целое  Индексированное поле: Да, допускаются совпадения |
| **Дата заказа** | Дата/время | Дата формирования заказа ДД.ММ.ГГ | Формат: Краткий формат даты  Индексированное поле: Да/Совпадения допускаются |
| **Скидка** | Числовой | Размер скидки в % | Размер поля: Одинарное с плавающей точкой  Формат: Процентный  Условие на значение: Between 0 And 1 |

\* Используя Мастер подстановок, сформировать для полей Код клиента и Код модели список выбора из таблиц Клиенты и Автомобили(сбросить флажок Скрыть ключевое поле).

4.8. Добавить 5 записей в таблицу Заказы.

**5-й этап. Создание схемы данных БД (связей между таблицами).**

5.1. Выполнить опцию Схема данных из вкладки Работа с базами данных. Вдиалогом окне Добавление таблицы последовательно добавить все три таблицы. Закрыть диалоговое окно.

5.2. Установить связь между таблицами Клиенты-Заказы, Автомобили-Заказы: выделить ключевое поле в главной таблице (Клиенты или Автомобили) и перетащить его на соответствующее поле таблицы-связки Заказы. Обеспечить целостность данных.

5.3. Сохранить макет схемы данных.

**Контрольные вопросы**

1. Перечислите основные этапы проектирования реляционной БД. Кратко поясните содержание каждого этапа.
2. Какие требования предъявляют к содержанию таблиц реляционной БД?
3. Приведите порядок создания макета таблицы в режиме Конструктора.
4. Для чего в каждой таблице задается первичный ключ? В чем различие между первичным и внешним ключом?
5. Порядок формирования схемы БД.
6. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.

### Методические рекомендации (указания) к выполнению самостоятельной работы №3

1. Тема самостоятельной работы: Формирование запросов и отчетов.
2. Количество часов: 4ч.
3. Задания для самостоятельной работы: выполнение заданий по вариантам
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 1.2 Применение языка VBA в разработке приложений», создание условий для

формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

1. Рекомендации преподавателя:

### Общие теоретические сведения

Современные информационные системы характеризуются большими объемами хранимых данных, их сложной организацией, а также высокими требованиями к скорости и эффективности обработки этих данных. Это становится возможным при использовании специальных программных средств – систем управления базами данных (СУБД).

**База данных (БД)** – это поименованная совокупность данных относящихся к определенной предметной области.

**Система управления базами данных (СУБД)**– это комплекс программных и языковых средств, необходимых для создания обработки баз данных и поддержания их в актуальном состоянии.

Почти все современные СУБД основаны на реляционной модели данных. Название "реляционная" связано с тем, что каждая запись в такой базе данных содержит информацию, относящуюся (related) только к одному объекту. Все данные в реляционной БД представлены в виде таблиц. Каждая строка таблицы содержит информацию только об одном объекте и называется **записью**. Столбец таблицы содержит однотипную для всех записей информацию и называется **полем**. Для успешного функционирования базы данных важна правильная организация данных в ней. При определении структуры данных в базе выделяют следующие основные понятия.

**Класс объектов** - совокупность объектов, обладающих одинаковым набором свойств. ***Например***, в базе данных о ВУЗе классами объектов могут быть студенты, преподаватели, предметы.

**Свойство (атрибут)**- определенная часть информации о некотором объекте. Хранится в виде столбца (поля) таблицы. ***Например***, фамилия, имя, отчество - это свойства для объекта Студент.

**Связь (отношение)** - способ, которым связана информация о разных объектах.

#### Типы связей между объектами

Основным структурным компонентом базы данных, как правило, является таблица. При определении состава таблиц следует руководствоваться правилом: в каждой таблице должны храниться данные только об одном классе объектов.

Если в базе данных должна содержаться информация о разных классах объектов, то она должна быть разбита на отдельные таблицы. Связь между таблицами осуществляется с помощью общих полей.

Связи между любыми двумя таблицами относятся к одному из трех типов: один-к-одному (1:1), один-ко-многим (1:М) и многие-ко-многим (М:М).

При установке связи типа **"один-к-одному"** (1:1) каждой записи в одной таблице соответствует не более одной записи в другой таблице.

Связь типа **"один-ко-многим"** (1:М) означает, что каждой записи в одной таблице соответствует несколько записей в связанной таблице. Этот наиболее распространенный тип связей. Для его реализации используются две таблицы. Одна из них представляет сторону "один", другая – сторону "много".

Связь типа **"много-ко-многим"** (М:М) используется, когда множеству записей в одной таблице соответствует множество записей в связанной таблице. Большинство современных СУБД непосредственно не поддерживают такой тип связи. Для ее реализации такая связь разбивается на две связи типа один-ко-многим. Соответственно, для хранения информации потребуется уже три таблицы: две со стороны "много" и одна со стороны "один". Связь между этими тремя таблицами также осуществляется по общим полям.

#### Структура MS Access

**MS Access** – это функционально полная реляционная СУБД, работающая в среде Windows. В Access база данных включает в себя все объекты, связанные с хранимыми данными (таблицы, формы, отчеты, запросы, макросы, модули). Все объекты Access хранятся в одном файле с расширением .accdb. В таблицах хранятся данные, которые можно просматривать, редактировать, добавлять. Используя формы, можно выводить данные на экран в удобном виде, просматривать и изменять их. Запросы позволяют быстро выбирать необходимую информацию из таблиц. С помощью отчетов можно создавать различные виды документов, для вывода на печать, макросы и модули позволяют автоматизировать работу с базой данных.

Запуск Access осуществляется двойным щелчком мыши по значку «MS Access» на рабочем столе, или в подменю «Microsoft Office» меню «Пуск».

После запуска на экране появится окно Приступая к работе с Microsoft Office Access, с помощью которого можно создать новую базу данных, выбрать нужный шаблон базы данных из Интернета или открыть локально расположенную базу данных.

При нажатии на кнопку Новая база данных правая часть окна изменится. В ней необходимо указать название файла новой базы данных и её будущее местоположение. Затем нажать кнопку Создать (рис. 1).

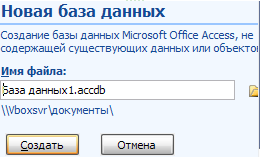


 Рис. 1. Создание новой базы данных MS Access 2007

Откроется окно новой базы данных с новой таблицей для дальнейшей работы. Верхняя строка представляет собой Ленту MS Office 2007. Она содержит вкладки и кнопки для выполнения определенных действий. Лента позволяет работать с таблицами, формами, запросами и отчетами базы данных (рис. 2).

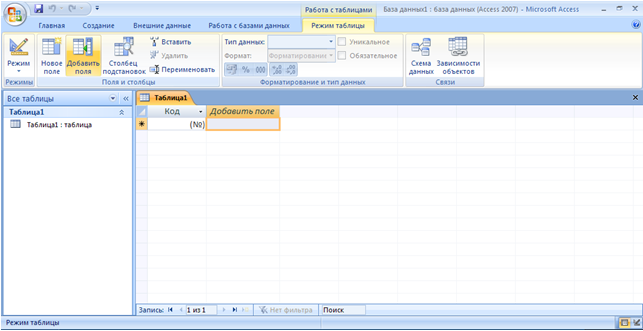


 Рис. 2. Созданная база данных с новой таблицей

#### Создание новой таблицы с помощью Конструктора

Для создания новой таблицы на вкладке Создание Ленты выбрать пиктограмму Конструктор таблиц. На экран будет выведено окно таблицы в режиме Конструктора, в котором можно задать имена, типы и свойства полей для вновь создаваемой таблицы.

Каждая строка в столбце Тип данных является полем со списком, элементами которого являются типы данных Access. Тип поля определяется характером вводимых в него данных (рис. 3).

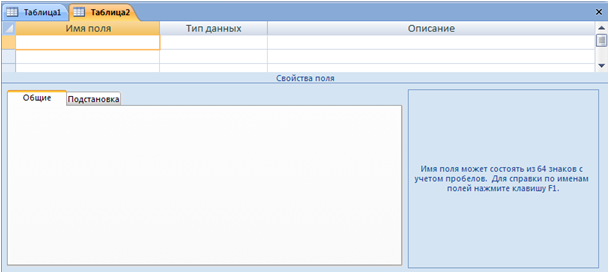


Рис. 3. Создание таблицы в режиме конструктора

Среди типов данных Access есть специальный тип – Счетчик. В поле этого типа Access автоматически нумерует строки таблицы в возрастающей последовательности. Редактировать значения такого поля нельзя.

Набор свойств поля зависит от выбранного типа данных. Для определения свойств поля используется бланк Свойства поля в нижней части окна конструктора таблиц.

#### Создание запросов

Для создания запроса выбрать на ленте вкладку Создание и нажать Мастер запросов. Появится окно для выбора способа построения запроса (рис. 4).

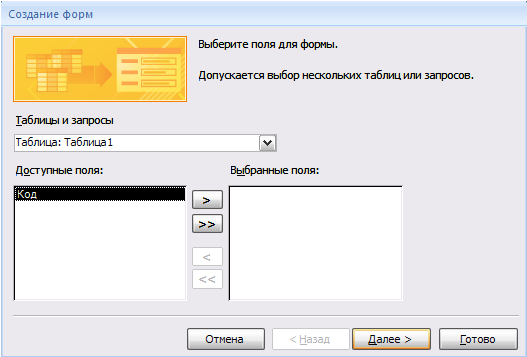


 Рис. 4. Мастер построения запросов

Существуют простые запросы и перекрестные запросы.

**Простой запрос** создает простой запрос из определенных полей.

**Перекрестный запрос** создает запрос, данные в котором имеют компактный формат, подобный формату сводных таблиц в Excel. С помощью перекрестного запроса можно более наглядно представить данные итоговых запросов, предусматривающих группировку по нескольким признакам (по двум, в частности).

В этом случае значение полей по первому признаку группировки могут стать заголовками строк, а по второму - заголовками столбцов.

Запросы имеют три режима отображения:

* **режим конструктора** – пример представлен выше. Этот режим рекомендуется для создания запросов. Запрос в этом режиме существует только в оперативной памяти;
* **режим SQL** – содержит команды на языке SQL (Structured Query Language – структурированный язык запросов) и указания, в каких таблицах и какие данные нужны пользователю.

Пример запись SQL-запроса, в котором производится выбор поля1и поля2 из таблицы с заданным именем для записей, если поле2 равно 0.

SELECT имя таблицы.[поле1], имя таблицы.[поле2]

FROM имя таблицы

WHERE(имя таблицы.[поле2]=0);

В таком виде запрос сохраняется в файле БД.

* **режим таблицы** – в этом режиме отображаются данные, отобранные с помощью запроса. На экране монитора данные, отвечающие условиям запроса представлены в форме таблицы.

#### Вычисляемые поля

Можно задать вычисления над любыми полями таблицы и сделать вычисляемое значение новым полем в запросе.

Для этого в строке Поле бланка **QBE** (бланк запроса) вводится формула для вычисления, причем имена полей, которые участвуют в вычислениях заключаются в квадратные скобки.

***Например:***

=[ Оклад]\*0.15.

При создании выражений для вычисляемых полей можно использовать Построитель выражений. Для этого нужно щелкнуть по пустому полю в бланке запроса, а затем по кнопке панели инструментов Построить, откроется окно Построитель выражений.

Все имена объектов, из которых строится выражение для вычисления, заключены в квадратные скобки, причем перед именем поля может стоять восклицательный знак (!) разделяющий имя поля и имя таблицы.

Выражение создается в верхней части окна. Можно самим ввести выражение, но проще использовать различные кнопки, расположенные под областью ввода.

#### Составление отчетов

Анализ данных в MS Access, может быть выполнен с помощью отчетов.

Основным предназначением отчетов, является представление данных для их просмотра как в электронной, так и в печатной форме. Возможность использовать отчеты для анализа данных обусловлена тем, что в них можно не только включать необходимые данные других объектов БД (таблиц, запросов и форм), но и использовать для их обработки формулы и выражения.

Существует два режима отображения отчетов. В режиме предварительного просмотра отчет отображается так, как он будет выглядеть при печати. Режим конструктора предоставляет пользователю доступ к макету отчета. При этом можно придать отчету необходимые свойства, а также изменить состав и свойства объектов отчета. В этом режиме можно создавать отчет. Однако обычному пользователю целесообразно для этого использовать мастер создания отчетов.

#### Сортировка записей

Сортировку записей MS Access может осуществлять по одному признаку, который выбирается пользователем путем установки курсора в нужный столбец таблицы, или нескольким признакам. В последнем случае выделяются поля, содержащие признаки, по которым должна осуществляться сортировка. Однако при этом следует учитывать, что сортировка производится только по признакам, записанным в смежных столбцах. Она осуществляется поочередно в каждом столбце, слева направо. Это означает, что для такой сортировки необходимо сначала изменить макет таблицы таким образом, чтобы соответствующие столбцы располагались рядом. При этом слева должны располагаться признаки, значения которых принимает большее количество записей.

Для фильтрации данных в таблицах БД можно использовать два типа фильтров: фильтр **«по выделенному»** или **расширенный фильтр**. С этой целью используется команда Записи/Фильтр... (для того чтобы эта команда стала доступной пользователю, необходимо открыть таблицу). Затем пользователь выбирает тип фильтра.

При применении фильтра «по выделенному» программа оставляет доступными для просмотра только записи, содержащие признак, совпадающий с тем, который выбрал пользователь. Такой фильтр можно установить, выделив в таблице часть поля, одну или несколько смежных ячеек, содержащих данные, которые должны быть в соответствующих полях результирующего набора. MS Access отобразит записи, совпадающие с выделенным образцом.

Особую разновидность фильтра «по выделенному» представляет собой результат выполнения команды Записи/ Фильтр/Исключить выделенное. В этом случае отбираются записи, не содержащие выделенных данных.

В случае использования расширенного фильтра СУБД открывает диалоговое окно с макетом фильтра.

В верхней части макета размещается окно с перечнем полей таблицы, а в нижней – бланк для записи условий фильтрации.

С дополнительным материалом по теме можно ознакомиться в литературе [1, 3, 4, 7].

#### Упражнение 1.

Создать базу данных с названием "Студенты-Экзамены".

***Решение.***

В диалоговом окне Приступая к работе с Microsoft Office выбрать Новая база данных, задать имя файла базы данных и выбрать место расположения. Будет создана новая база данных вместе с новой таблицей (рис. 5).

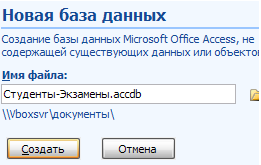


Рис. 5. Создание новой базы данных «Студенты-Экзамены»

#### Упражнение 2.

Создать новую таблицу с названием «Студенты», включающую поля:

    1) ФИО;

    2) номер зачетки;

    3) дата рождения;

    4) группа;

    5) адрес;

    6) стипендия;

    7) телефон.

Поле Номер зачетки установить ключевым полем.

***Решение.***

    1. Закрыть появившуюся автоматически Таблицу1 (нажать правой кнопкой мыши на вкладке «Таблица1» под лентой).

    2. Выбрать на ленте вкладку Создание, в разделе Таблицы выбрать Конструктор таблиц.

    3. Задать необходимые имена полей согласно заданию.

    4. Определить и установить соответствующий тип каждого поля (текстовый, числовой, дата/время, денежный) (рис. 6).

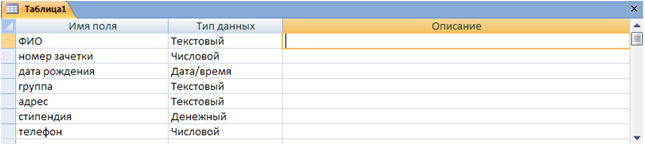


Рис. 6. Заполнение структуры таблицы

    5. Для поля Группа в нижней части окна в разделе Свойства поля задать значение Размер поля 10.

    6. Задать ключевое поле Номер зачетки (Выделить поле Номер\_зачетки и нажать кнопку Ключевое поле или задать с помощью контекстного меню) (рис. 7).

https://sites.google.com/a/ssga.ru/ssga4school/_/rsrc/1336710332786/informatika/practics/lab-8/128.png

Рис. 7. Создание ключевого поля

    7. Сохранить таблицу с именем Студенты.

#### Упражнение 3.

В режиме Таблица ввести 2 записи в таблицу Студенты.

***Решение.***

1. Перейти в режим Таблицы с помощью ленты, выбрать вкладку Конструктор, кнопку Режим.
2. Ввести 2 записи (рис. 8).

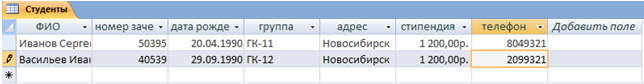


Рис. 8. Ввод двух записей в таблицу

Ввод данных в таблицу можно выполнить с помощью форм.

Access предлагает следующие способы создания форм:

* **Конструктор форм** – позволяет разрабатывать собственные экранные формы с заданными свойствами для просмотра, ввода и редактирования данных.
* **Мастер форм** – позволяет достаточно быстро создать форму на основе выбранных для нее данных.
* Автоформа: в столбец, ленточная, табличная.
* **Диаграмма** – позволяет создавать форму, данные в которой представлены в виде диаграммы.

По сравнению с простыми автоформами, формы, созданные с помощью Мастера более разнообразны по стилю оформления, могут содержать выбранные поля, в т. ч. и из нескольких связанных таблиц.

Для запуска Мастера форм нужно на ленте во вкладке Создание выбрать раскрывающийся список Другие формы – Мастер форм.

На 1 шаге Мастера форм необходимо определить поля будущей формы. После указания имени таблицы/запроса в списке Доступные поля появляется перечень всех полей данной таблицы.

Необходимо из этого перечня перенести все необходимые поля в список Выбранные поля.

На 2 шаге предлагается задать внешний вид формы: в один столбец, ленточный, табличный, выровненный.

На 3 шаге выбирается стиль формы из списка вариантов стилей.

На 4 шаге открывается последнее окно Мастера форм, где нужно ввести имя создаваемой формы (по умолчанию ей дается имя базовой таблицы/запроса) и вариант дальнейшей работы (открытие формы для просмотра, изменение макета формы).

Создавать и редактировать формы любой степени сложности позволяет только Конструктор форм. Также в режиме Конструктора можно отредактировать формы, созданные Мастером, или автоформы.

Для создания формы в режиме Конструктора перейти в ленте на вкладку Создание и выбрать Конструктор форм. На экране откроется окно Конструктора форм.

В окне конструктора форм элементы, расположенные в разделах заголовка и примечания формы, отображаются только в заголовке и примечании формы. Элементы, расположенные в области данных, отображаются для каждой записи базовой таблицы/запроса.

#### Упражнение 4.

С помощью мастера форм создать форму для заполнения таблицы Студенты и ввести 2 записи в форму.

***Решение.***

    1. Выбрать на ленте вкладку Создание, в разделе Формы выбрать Другие формы – Мастер форм.

    2. Следовать указаниям мастера форм (выбрать поля из таблицы Студенты (рис. 9).

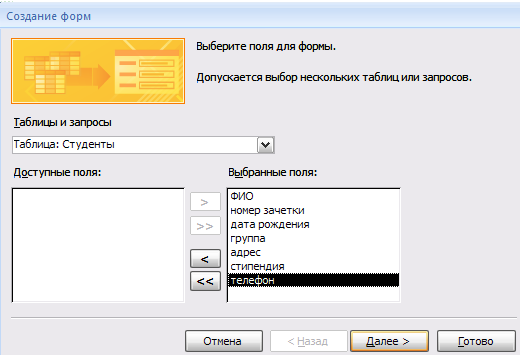


Рис. 9. Мастер форм

    3. Перейти с помощью формы до 3 записи с помощью указателей-стрелок или нажать на кнопку Новая (пустая) запись внизу окна формы.

    4. Ввести 2 новые записи (рис. 10).

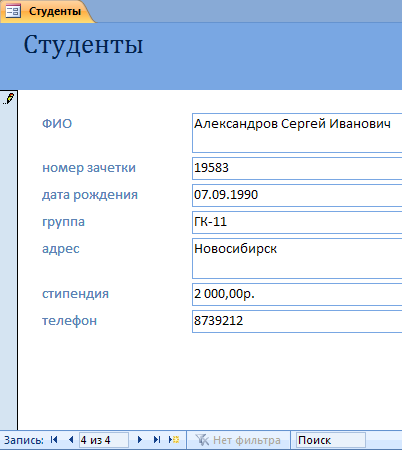


Рис. 10. Созданная форма с 4 записями

#### Упражнение 5.

Создать в режиме конструктора новую таблицу с названием «Экзамены» с полями: Номер зачетки, Экзамен1, Экзамен2, Экзамен3. Ключевое поле не создавать, поля Номер зачетки, Экзамен1, Экзамен2 и Экзамен3 задать числовыми.

#### Упражнение 6.

Установить тип поля Номер зачетки в таблице с помощью мастера подстановок, используя данные из таблицы Студенты.

Мастер подстановок позволяет формировать для нужного поля список значений, который может содержать данные другой таблицы или запроса, либо состоять из фиксированного набора значений. В обоих случаях Мастер подстановок облегчает ввод данных, так как поле Номер зачетки является общим для обеих таблиц.

Комбинированный список для поля Номер зачетки формируется на основе данных связанной таблицы Студенты.

***Решение.***

    1. Войти в режим конструктора таблицы Экзамены.

    2. Перейти в колонку Тип данных для поля Номер зачетки.

    3. Из списка доступных типов полей выбрать элемент Мастер подстановок (рис. 11).

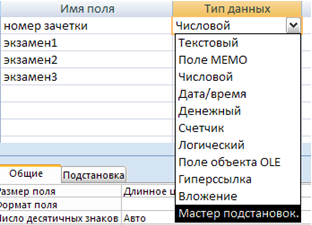


 Рис. 11. Выбор Мастера подстановок из списка Тип данных

Первое диалоговое окно Мастера подстановок предлагает выбрать источник формирования списка: на основе данных таблицы/запроса или фиксированного набора значений. В данном случае нужно выбрать первый вариант (рис. 12).

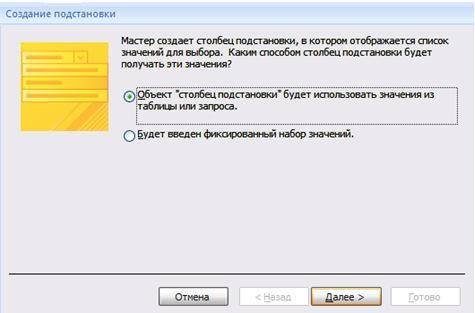


Рис. 12. Окно «Создание подстановки». Шаг 1

В следующем окне из приведенного списка таблиц/запросов следует выбрать таблицу/запрос, являющуюся источником данных для списка. В нашем примере такой таблицей является таблица Студенты, так как она служит источником данных для списка номеров зачеток (рис. 13).

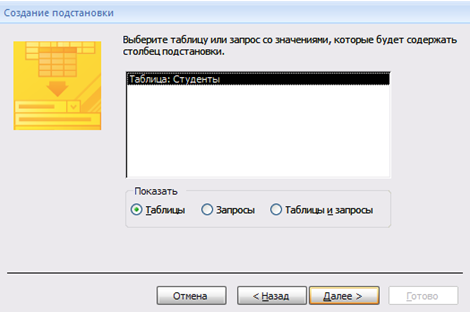


 Рис. 13. Окно «Создание подстановки». Шаг 2

В третьем окне Мастера подстановок из списка Доступные поля нужно выбрать поля, значения которых используются в списке. В данном случае можно выбрать поле ФИО, которое сделает список более информативным (рис. 14).

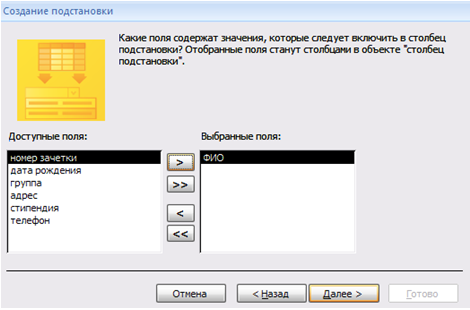


 Рис. 14. Окно «Создание подстановки». Шаг 3

Выбрать сортировку списка «ФИО» по возрастанию (рис. 15).

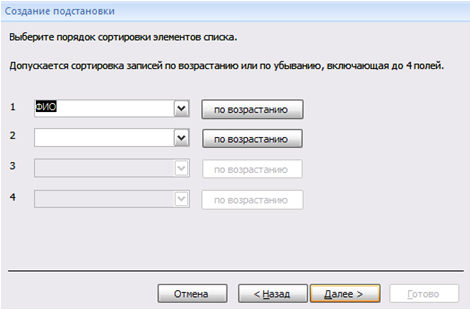


 Рис. 15. Окно «Создание подстановки». Шаг 4

Задать ширину столбцов, которые содержат столбец подстановки.

Нажать кнопку Готово для завершения процесса проектирования комбинированного списка.

    4. Сохранить таблицу Экзамены.

    5. В режиме Таблица ввести 2 записи в таблицу Экзамены.

#### Упражнение 7.

С помощью мастера форм создать форму для заполнения таблицы Экзамены и ввести 2 записи в форму.

***Решение.***

1. Выбрать на ленте вкладку Создание, в разделе Формы выбрать Другие формы – Мастер форм.
2. Следовать указаниям мастера форм (выбрать поля из таблицы Экзамены).
3. Перейти с помощью формы до 3 записи с помощью указателей-стрелок или нажать на кнопку Новая (пустая) запись внизу окна формы.
4. Ввести 2 новые записи.

#### Упражнение 8.

Установить связь один-ко-многим между таблицами.

***Решение.***

Для установления (изменения) связей между таблицами необходимо закрыть все открытые таблицы, формы, отчеты и запросы. На Ленте выбрать вкладку Работа с базами данных. Нажать кнопку Схема данных. Появляется Схема данных, включающая 2 таблицы и связи между ними. Связь между таблицами Студенты и Экзамены мы создали при установке мастера подстановок (рис. 16).

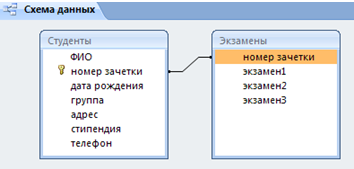


 Рис. 16. Схема данных

Перед созданием новой связи необходимо удалить старую. Для этого на линии между таблицами щелкнуть правой кнопкой мыши, выбрать Удалить. После этого нажать правой кнопкой мыши по названию каждой из таблиц и нажать Скрыть. Закрыть Схему данных, сохранив изменения.

Вновь открыть Схему данных. Для добавления в схему данных таблиц нажать кнопку Отобразить таблицу на Ленте или с помощью контекстного меню выбрать Добавить таблицу.

В окне Добавление таблицы нужно выделить имена таблиц, добавляемых в схему данных, и нажать на кнопку Добавить. После этого данное окно закрыть. В окне Схема данных появятся имена всех указанных таблиц вместе со списками полей (рис. 17).

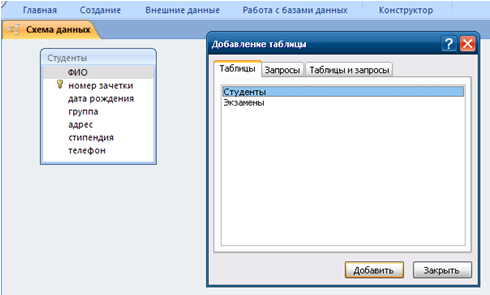


Рис. 17. Окно «Добавление таблицы» в Схеме данных

Добавить таблицу Экзамены и закрыть окно «Добавление таблицы».

Для связи нужных полей (в нашем случае Номер зачетки) нужно выделить ключевое поле Номер зачетки ключевой таблицы Студенты, нажать левую кнопку мыши, перетащить ее курсором на аналогичное поле в связываемой таблице Экзамены, после чего кнопку мыши отпустить. В результате появится диалоговое окно Связи. В этом окне Access заполнит первую строку именем поля, по которому связывались таблицы. Чтобы в связанных таблицах не нарушалась целостность данных, нужно щелкнуть по флажку Обеспечение целостности данных. После этого Access сделает невозможным запись в не ключевую таблицу такого значения общего поля, которого нет в ключевой таблице. После установления целостности данных Access включает две дополнительные опции: Каскадное обновление связанных полей и Каскадное удаление связанных полей (рис. 18).

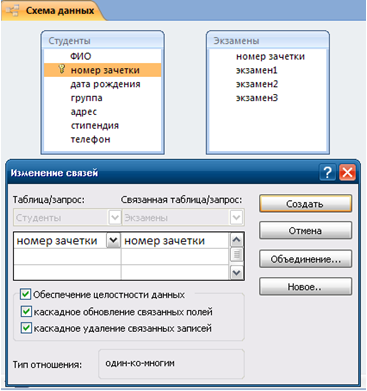


 Рис. 18. Окно «Изменение связей»

Если выбрать первую опцию, то при изменении какого-либо значения ключевого поля в ключевой таблице Access автоматически обновит значения этого поля для соответствующих записей во всех связанных таблицах. Например, если у одного из студентов изменился номер зачетки в таблице Студенты, то он автоматически должен измениться и в таблице Экзамены.

Выбор второй опции при удалении одной из записей в ключевой таблице приведет к удалению тех записей в таблице со стороны "много", которые имеют такое же значение ключа. Например, если из таблицы Студенты удалить запись об одном из студентов, то записи о результатах сданных им экзаменов будут удалены автоматически. Включим опции Каскадное обновление связанных полей и Каскадное удаление связанных записей.

Для завершения процесса создания связей, нужно щелкнуть по кнопке Создать. Access нарисует линию между таблицами в окне Схема данных, указывающую на наличие связи Один-ко-многим между ними. На конце линии у таблице со стороны "один" будет стоять цифра 1, а на другом конце, у таблицы со стороны "много" – символ бесконечности ∞ . После закрытия этого окна все установленные связи будут сохранены (рис. 19).

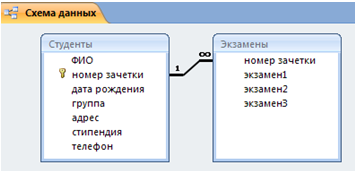


 Рис. 19. Созданная связь «один-ко-многим» между таблицами

#### Упражнение 9.

Создать форму для заполнения сразу обеих таблиц с помощью мастера форм и ввести поля: из первой таблицы Студенты: ФИО, Номер\_зачетки, Дата\_рождения, Группа, Адрес, Телефон, Стипендия; из второй таблицы Экзамены: Экзамен1, Экзамен2, Экзамен.

Ввести дополнительно 3 записи с помощью созданной формы.

***Решение.***

1. Выбрать на ленте вкладку Создание, в разделе Формы выбрать Другие формы – Мастер форм.
2. Следуя указаниям мастера форм выбрать необходимые поля из таблиц Студенты и Экзамены.
3. На втором шаге выбрать вид представления данных «Подчиненные формы».
4. Далее выбрать внешний вид подчиненной формы «Табличный».
5. Стиль выбрать по желанию.
6. На следующем шаге имена форм оставить установленными по умолчанию.
7. Открыть созданную форму, перейти с помощью формы до 5 записи с помощью указателей-стрелок или нажать на кнопку Новая (пустая) запись внизу окна формы.
8. Ввести 3 новые записи.

#### Упражнение 10.

Создать запрос в режиме конструктора с сортировкой по возрастанию по дате рождения и группе и вывести на экран только хорошистов (тех, у кого все оценки за экзамены не ниже 4), с указанием поля ФИО и номера зачетки студентов.

***Решение.***

    1. Перейти на вкладку Создание и нажать Конструктор запросов. Появится окно для построения запроса (рис. 20).

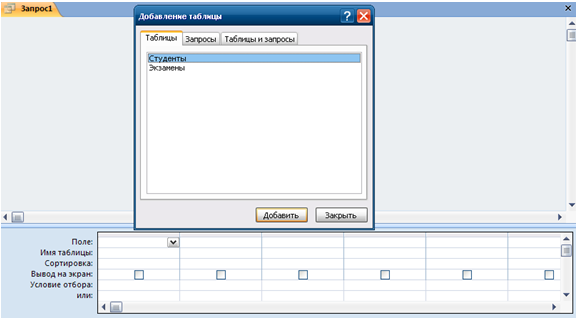


 Рис. 20. Окно «Добавление таблицы» в конструкторе запросов

    2. В окне Добавление таблицы выбрать нужные таблицы.

    3. В Поле указать нужные поля из таблиц, перечисленных в задании (ФИО, номер зачетки, дата рождения, группа, Экзамен1, Экзамен2, Экзамен3).

    4. Установить сортировку по возрастанию для Даты\_рождения и Группы.

    5. В Условие отбора в столбце Экзамен1, Экзамен2, Экзамен3 установить >3, что означает оценки которые больше 3. Второй вариант: установить 4 or 5.

    6. Установка в одной строке Условия отбора означает, логическую операцию И Экзамен1 >3, И Экзамен2>3, И Экзамен3 >3. То есть все три условия должны выполняться одновременно (рис. 21).

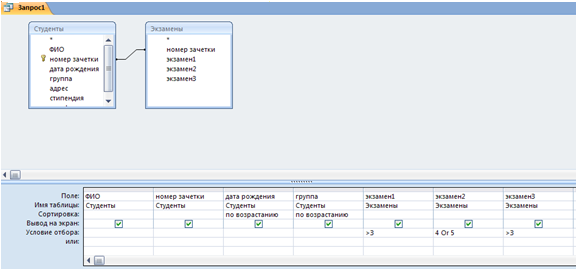


 Рис. 21. Созданный запрос

    7. Перейти в режим таблицы или нажать кнопку Выполнить для просмотра результата запроса (рис. 22).

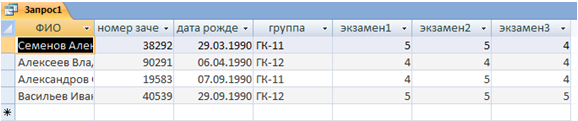


Рис. 22. Результат выполнения запроса

#### Упражнение 11.

Создать запросы в режиме конструктора с использованием союзов И, ИЛИ:

* вывести студентов, ФИО которых начинается на букву П и оценка за Экзамен1 «Не 3»;
* вывести студентов, родившихся между 01.01.1990 и 01.10.1992 или у кого стипендия не меньше 1000 руб.

***Решение.***

Для вывода ФИО, которые начинаются на определенную букву задать условие: Like “П\*”.

Для обозначения отрицания НЕ используется оператор Not (рис. 23).

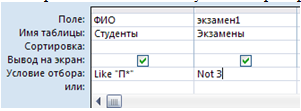


 Рис. 23. Созданный запрос

Для обозначения МЕЖДУ используется оператор Between. Например, для задания даты рождения в определенном интервале: может быть задан шаблон Between #дата1# and #дата2#.

При необходимости использования союза ИЛИ в запросе, условия размещаются в следующих строках (или), расположенных ниже строки Условие отбора (рис. 24).

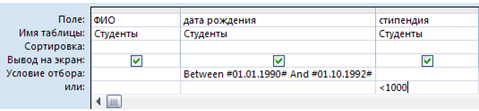


 Рис. 24. Созданный запрос

#### Упражнение 12.

По созданному запросу1 составить отчет в режиме мастер отчетов. В режиме конструктора заменить название отчета на Список студентов.

***Решение.***

Перейти на вкладку Создание и выбрать Мастер отчетов. Для отчета использовать Запрос1 и следовать указаниям мастера отчетов. Для изменения имени отчета выбрать Конструктор отчетов и в поле названия отчета сделать изменения (рис. 25).



 Рис. 25. Созданный отчет

#### Упражнение 13.

Добавить в таблицу Студенты новое поле Пол. В форму Студенты добавить группу переключателей с названием «Пол». Ввести значения с помощью формы.

***Решение.***

    1. Добавить в таблицу Студенты новое поле Пол после поля ФИО в режиме Конструктор. Тип данных поля – текстовый. В описании поля (третий столбец в режиме конструктора) введите «1-мужской, 2-женский». Описание будет выводиться в левой части строки состояния (расположена в нижней части окна).

    2. Таблицу Студенты сохранить и закрыть.

    3. Открыть форму Студенты в режиме конструктора.

    4. Увеличить свободное пространство формы (белая область с сеткой) путем перемещения нижней строки Примечание формы.

    5. При выборе режима Конструктора на Ленте включается вкладка Инструменты конструктора форм, которые позволяют вставлять объекты в форму.

    6. На Ленте во вкладке Конструктор в группе Элементы управления нажать кнопку Группа переключателей.

    7. Щелкнуть на свободном пространстве формы. Ввести в окне Мастера по созданию Группы значения: мужской; женский (рис. 26).

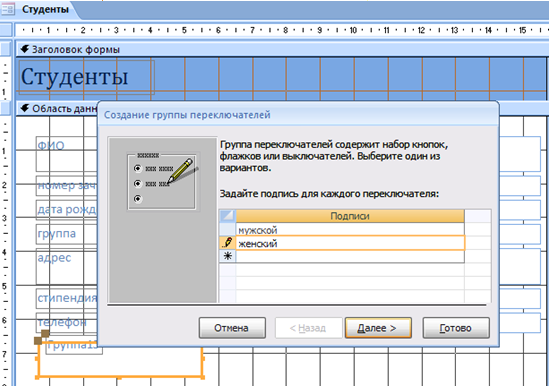


 Рис. 26. Создание группы переключателей. Шаг 1

    8. В следующем диалоговом окне не задавать переключатель, используемый по умолчанию.

    9. В третьем диалоговом окне оставить значения, установленные автоматически.

   10. В четвертом диалоговом окне установить переключатель в пункт «Сохранить значение в поле» и выбрать из списка необходимое поле Пол.

    11. На следующем шаге оставить все настройки без изменений.

    12. На последнем шаге задать подпись для группы переключателей – Пол. Нажать кнопку Готово (рис. 27).

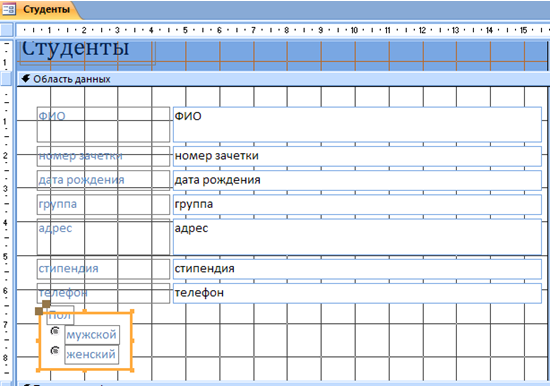


 Рис. 27. Размещенная группа переключателей на форме

#### Упражнение 14.

Задать значение поля Пол для каждой записи таблицы Студенты.

***Решение.***

1. Перейти в режим Формы (Двойной щелчок на имени Формы) и пролистать все записи и для каждой установить переключатель в нужное положение.
2. Закрыть форму и открыть таблицу и просмотреть поле Пол.

#### Упражнение 15.

С помощью фильтра вывести из таблицы Студенты на экран студентов одной группы.

***Решение.***

1. Открыть таблицу Студенты в режиме таблицы.
2. Установить курсор в поле Группа.
3. На вкладке Главная нажать кнопку Фильтр.
4. Установить галочку на номере одной нужной группы. Нажать ОК. Произойдет автоматическое включение фильтра. Данные будут отфильтрованы по установленному параметру.
5. Кнопка Применить фильтр позволяет включать и выключать установленный фильтр.
6. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.

### Методические рекомендации (указания) к выполнению самостоятельной работы №4

1. Тема самостоятельной работы: Создание процедур обработки событий с применением языка VBA.
2. Количество часов: 6ч.
3. Задания для самостоятельной работы: выполнение заданий по вариантам
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 1.2 Применение языка VBA в разработке приложений», создание условий для формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

1. Рекомендации преподавателя:

Visual Basic for Applications (VBA, Visual Basic для приложений) — немного упрощённая реализация языка программирования Visual Basic, встроенная в линейку продуктов Microsoft Office (включая версии для Mac OS), а также во многие другие программные пакеты, такие как AutoCAD, SolidWorks, CorelDRAW, WordPerfect. VBA покрывает и расширяет функциональность ранее использовавшихся специализированных макро-языков, таких как WordBasic.

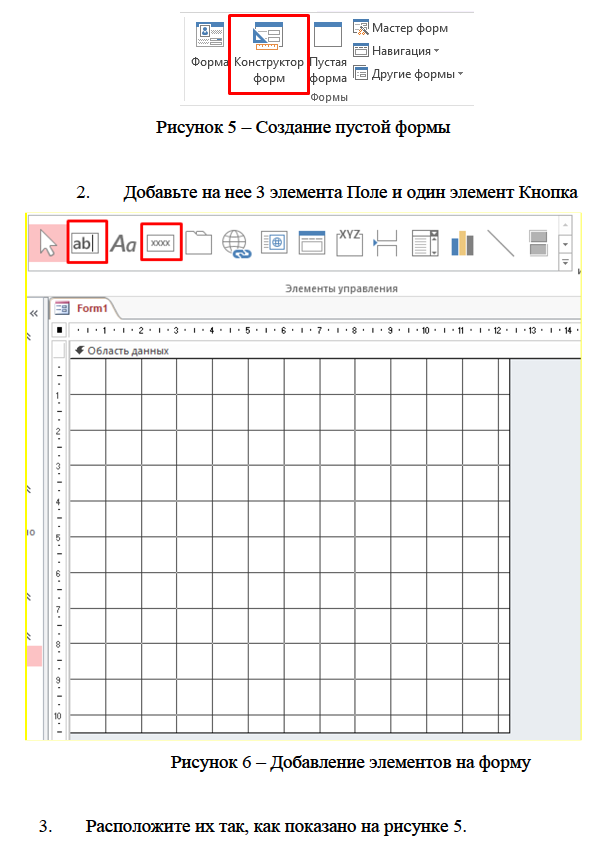
VBA является интерпретируемым языком. Как и следует из его названия, VBA близок к Visual Basic. VBA, будучи языком, построенным на COM, позволяет использовать все доступные в операционной системе COM объекты и компоненты ActiveX. По сути, возможно создание приложения на основе Microsoft Word VBA, использующего только средства Corel Draw.

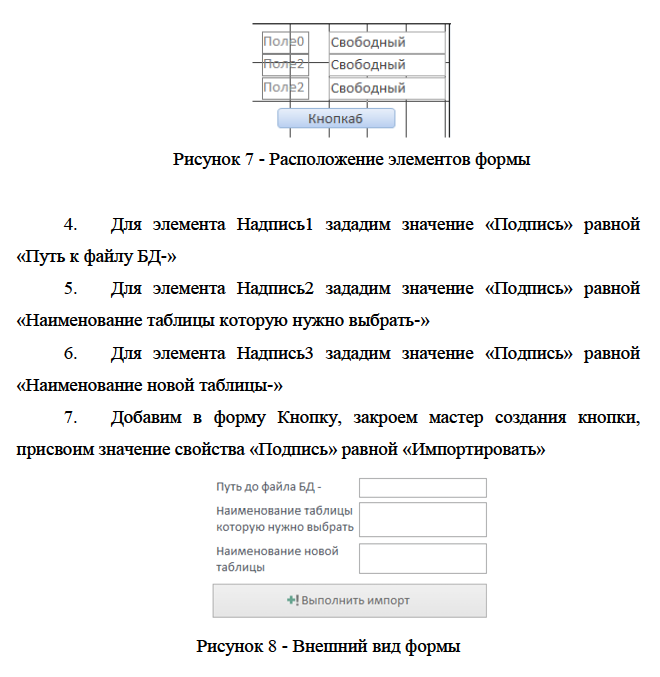
Написание программ на VBA позволяет: сделать базу данных "динамичной"; уменьшить размер базы данных (программирование позволит не создавать дополнительные объекты); обращаться к архивным данным в другие файлы или другие базы данных (т.е. подставлять в форму или в отчет такие данные), решать такие задачи, которые стандартными средствами MS Access решить невозможно.

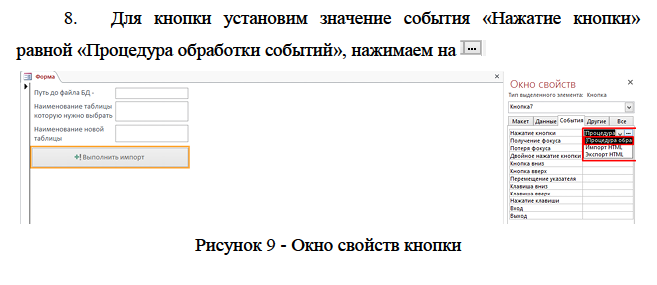
Для выполнения всех заданий потребуется готовая база данных «Зарплата».

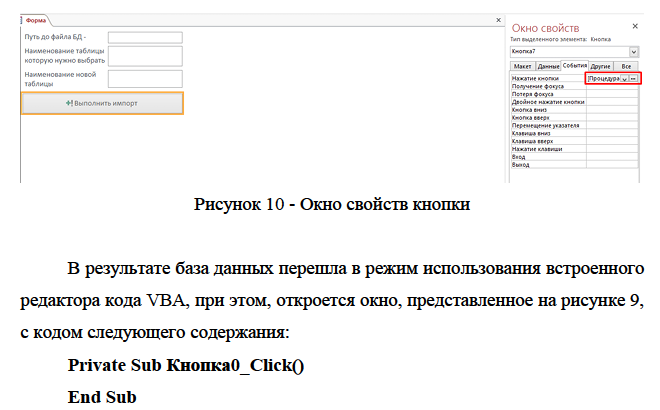
Создание формы импорта данных из другой таблицы, с использованием SQL-запроса и языка программирования VBA

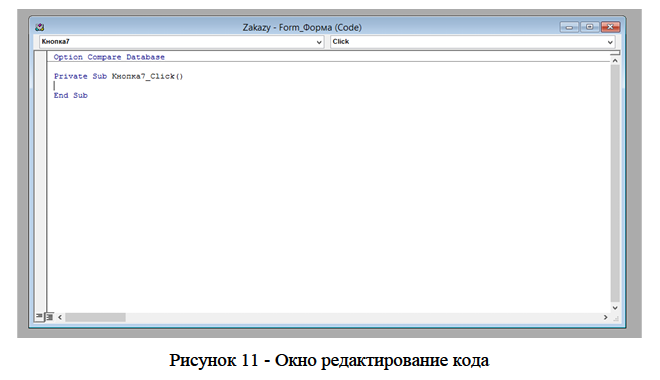
**Задание 1.** 1. В используемой базе данных создайте новую, пустую форму при помощи вкладки меню Создать-Конструктор форм.

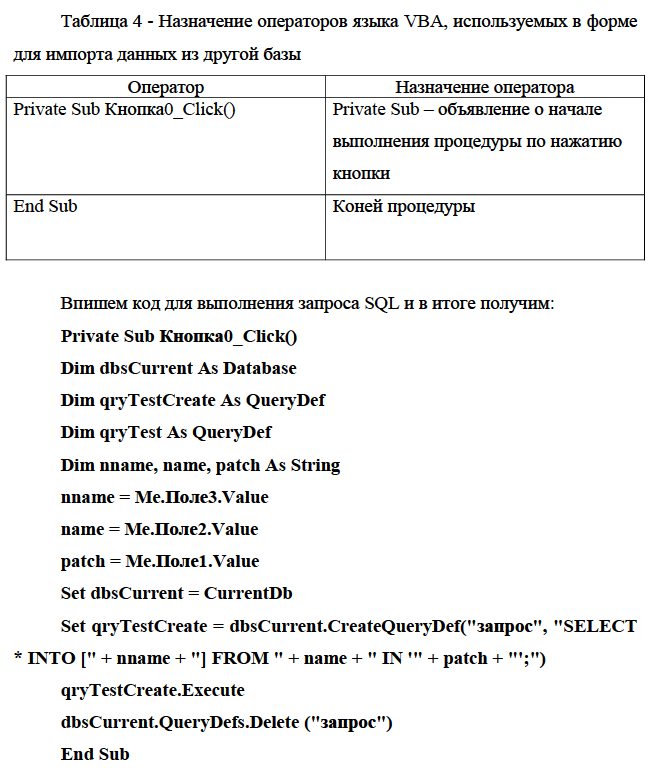


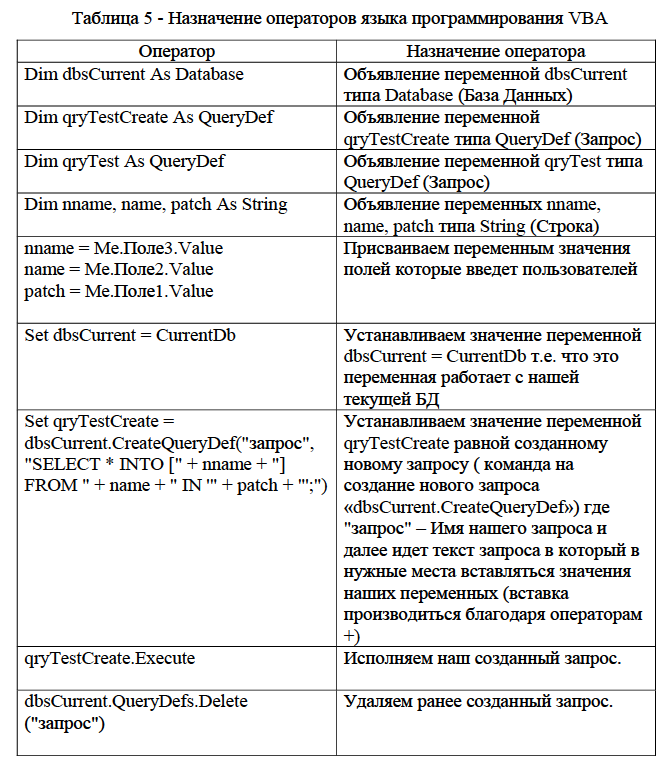
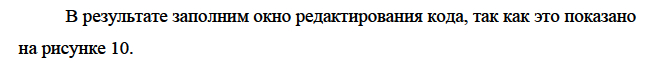


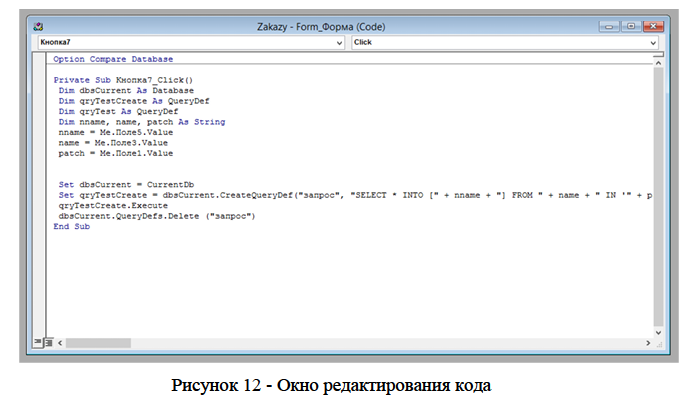


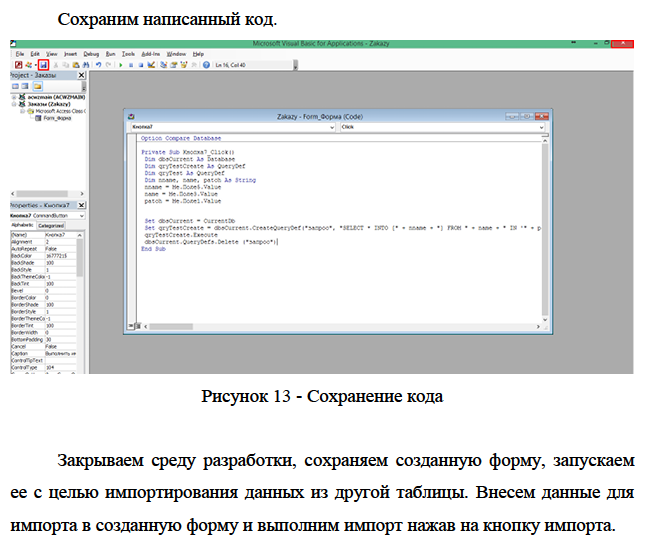


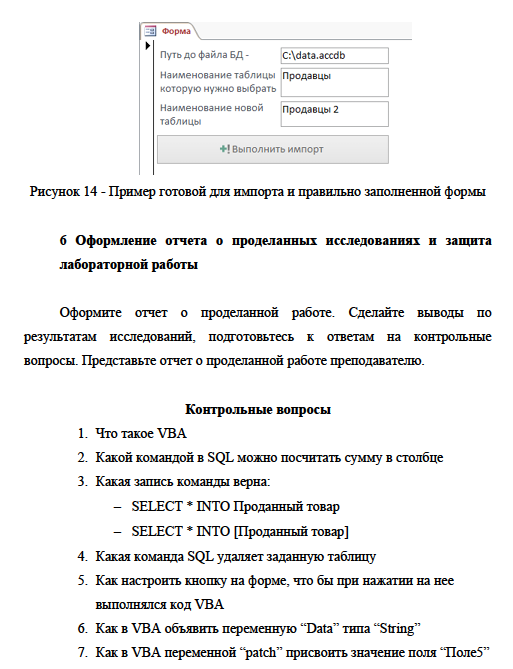








1. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.

### Методические рекомендации (указания) к выполнению самостоятельной работы №5

1. Тема самостоятельной работы: Использование алиасов, агрегатных функций с группировками.
2. Количество часов: 8ч.
3. Задания для самостоятельной работы: выполнение индивидуальных заданий
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 2.1 Организация запросов SQL», создание условий для формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

1. Рекомендации преподавателя:

В SQL добавлены дополнительные функции, которые позволяют вычислять обобщенные групповые значения. Для применения агрегатных функций предполагается предварительная операция группировки. В чем состоит суть операции группировки? При группировке все множество кортежей отношения разбивается на группы, в которых собираются кортежи, имеющие одинаковые значения атрибутов, которые заданы в списке группировки.

Например, сгруппируем отношение R1 по значению столбца Дисциплина. Мы получим 4 группы, для которых можем вычислить некоторые групповые значения, например количество кортежей в группе, максимальное или минимальное значение столбца Оценка.

Это делается с помощью агрегатных функций. Агрегатные функции вычисляют одиночное значение для всей группы таблицы. Список этих функций представлен в табл. 1.

|  |  |
| --- | --- |
|  | Таблица 1 |

### *Агрегатные функции*

|  |  |
| --- | --- |
| Функция | Результат |
| COUNT | Количество строк или непустых значений полей, которые выбрал запрос |
| SUM | Сумма всех выбранных значений данного поля |
| AVG | Среднеарифметическое значение всех выбранных значений данного поля |
| MIN | Наименьшее из всех выбранных значений данного поля |
| MAX | Наибольшее из всех выбранных значений данного поля |

### *Группировка отношения R1 по значению столбца «Дисциплина»*

|  |  |
| --- | --- |
|  | R1 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | ФИО | Дисциплина | Оценка |
| Группа 1 | Петров Ф. И. | Базы данных | 5 |
|  | Сидоров К. А. | Базы данных | 4 |
|  | Миронов А. В. | Базы данных | 2 |
|  | Степанова К. Е. | Базы данных | 2 |
|  | Крылова Т. С. | Базы данных | 5 |
|  | Владимиров В. А. | Базы данных | 5 |
| Группа 2 | Сидоров К. А. | Теория информации | 4 |
|  | Степанова К. Е. | Теория информации | 2 |
|  | Крылова Т. С. | Теория информации | 5 |
|  | Миронов А. В. | Теория информации | Null |
| Группа 3 | Трофимов П. А. | Сети и телекоммуникации | 4 |
|  | Иванова Е. А. | Сети и телекоммуникации | 5 |
|  | Уткина Н. В. | Сети и телекоммуникации | 5 |
| Группа 4 | Владимиров В. А. | Английский язык | 4 |
|  | Трофимов П. А. | Английский язык | 5 |
|  | Иванова Е. А. | Английский язык | 3 |
|  | Петров Ф. И. | Английский язык | 5 |

Агрегатные функции используются подобно именам полей в операторе SELECT, но с одним исключением: они берут имя поля как аргумент. С функциями SUM и AVG могут использоваться только числовые поля. С функциями COUNT, MAX и MIN могут использоваться как числовые, так и символьные поля. При использовании с символьными полями MAX и MIN будут транслировать их в эквивалент ASCII кода и обрабатывать в алфавитном порядке. Некоторые СУБД позволяют использовать вложенные агрегаты, но это является отклонением от стандарта ANSI со всеми вытекающими отсюда последствиями.

Например, можно вычислить количество студентов, сдававших экзамены по каждой дисциплине. Для этого надо выполнить запрос с группировкой по полю «Дисциплина» и вывести в качестве результата название дисциплины и количество строк в группе по данной дисциплине. Применение символа \* в качестве аргумента функции COUNT означает подсчет всех строк в группе.

SELECT R1. Дисциплина, COUNT(\*)

FROM R1

GROUP BY R1.Дисциплина

Результат:

|  |  |
| --- | --- |
| Дисциплина | COUNT(\*) |
| Базы данных | 6 |
| Теория информации | 4 |
| Сети и телекоммуникации | 3 |
| Английский язык | 4 |

Если же мы хотим сосчитать количество сдавших экзамен по какой-либо дисциплине, то нам необходимо исключить неопределенные значения из исходного отношения перед группировкой. В этом случае запрос будет выглядеть следующим образом:

SELECT R1.Дисциплина, COUNT (\*)

FROM R1

WHERE R1. Оценка IS NOT NULL

GROUP BY R1.Дисциплина

Результат:

|  |  |
| --- | --- |
| Дисциплина | COUNT(\*) |
| Базы данных | 6 |
| Теория информации | 3 |
| Сети и телекоммуникации | 3 |
| Английский язык | 4 |

В этом случае строка

|  |  |  |
| --- | --- | --- |
| Миронов А. В. | Теория информации | Null |

не попадет в набор кортежей перед группировкой, поэтому количество кортежей в группе для дисциплины «Теория информации» будет на 1 меньше.

Аналогичный результат можно получить, если записать запрос следующим способом:

SELECT R1. Дисциплина, COUNT(R1. Оценка)

FROM R1

GROUP BY R1. Дисциплина

Дело в том, что функция COUNT (ИМЯ АТРИБУТА) считает количество определенных значений в группе, в отличие от функции COUNT(\*), которая считает количество строк в группе. Действительно, в группе с дисциплиной «Теория информации» будет 4 строки, но только 3 определенных значения атрибута «Оценка».

Правила обработки значений NULL в агрегатных функциях

1. Если какие-либо значения в столбце равны NULL при вычислении результата функции они исключаются.
2. Если все значения в столбце равны NULL, то **Max Min Sum Avg**= NULL, count = 0 (ноль).
3. Если таблица пуста, count(\*) = 0.

Можно применять агрегатные функции также и без операции предварительной группировки, в этом случае все отношение рассматривается как одна группа и для этой группы можно вычислить одно значение на группу.

Правила интерпретации агрегатных функций

Агрегатные функции могут быть включены в список вывода и тогда они применяются ко всей таблице.

SELECT MAX (Оценка) from R1 даст максимальную оценку на сессии;

SELECT SUM (Оценка) from R1 даст сумму всех оценок на се c си;

SELECT AVG(Оценка) from R1 даст среднюю оценку по всей сессии.

Обратившись снова к базе данных «Сессия» (таблицы R1, R2, R3), найдем количество успешно сданных экзаменов:

SELECT COUNT(\*) As Сдано \_ экзаменов

FROM R1

WHERE Оценка > 2

Результат:

|  |
| --- |
| Сдано\_экзаменов |
| 13 |

 Это, конечно, отличается от выбора поля, поскольку всегда возвращается одиночное значение, независимо от того, сколько строк находится в таблице. Аргументом агрегатных функций могут быть отдельные столбцы таблиц. Но для того, чтобы вычислить, например, количество различных значений некоторого столбца в группе, необходимо применить ключевое слово DISTINCT совместно с именем столбца. Вычислим количество различных оценок, полученных по каждой дисциплине:

SELECT R1.Дисциплина, COUNT (DISTINCT R1.Оценка)

FROM R1

WHERE R1. Оценка IS NOT NULL

GROUP BY R1.Дисциплина

Результат:

|  |  |
| --- | --- |
| Дисциплина | COUNT(DISTINCT R1.Оценка) |
| Базы данных | 3 |
| Теория информации | 3 |
| Сети и телекоммуникации | 2 |
| Английский язык | 2 |

Тот же самый результат мы получим, исключив явное условие в части WHERE, в этом случае запрос будет иметь следующий вид:

SELECT R1. Дисциплина, COUNT(DISTINCT R1. Оценка)

FROM R1

GROUP BY R1. Дисциплина

Функция COUNT (DISTINCT R1.Оценка) считает только определенные различные значения.

В результат можно включить значение поля группировки и несколько агрегатных функций, а в условиях группировки можно использовать несколько полей. При этом группы образуются по набору заданных полей группировки. Операции с агрегатными функциями могут быть применены к объединению множества исходных таблиц. Например, поставим вопрос: определить для каждой группы и каждой дисциплины количество успешно сдавших экзамен и средний балл по дисциплине. Применим принцип дополнительного именования результирующих столбцов. Если мы хотим получить визуально более приемлемый результат, то можем написать после любого столбца его новое название, которое будет выведено в результирующем наборе данных. Новое название пишется через пробел и ключевое слово as. В некоторых СУБД допускается опускать данное ключевое слово и просто отделять имя столбца результата от выражения пробелом.

SELECT R2.Группа, R1.Дисциплина,COUNT(\*) as Всего,AV G (Оценка) as Средний\_балл

FROM R1,R2

WHERE R1. ФИО = R2. ФИО AND

R1. Оценка IS NOT NULL AND

R1.Оценка > 2

GROUP BY R2.Группа, R1.Дисциплина

Результат:

|  |  |  |  |
| --- | --- | --- | --- |
| Группа | Дисциплина | Всего | Средний\_балл |
| 4906 | Базы данных | 3 | 4.66 |
| 4906 | Теория информации | 2 | 4.5 |
| 4807 | Сети и телекоммуникации | 2 | 5 |

 Однако следует отметить, что данный результат мы получим в MS Access. А если мы выполним тот же запрос в MS SQL ServeR, то мы получим иной результат, потому что там при вычислении среднего просто отбрасывается дробная часть. Поэтому там результат будет:

|  |  |  |  |
| --- | --- | --- | --- |
| Группа | Дисциплина | Всего | Средний\_балл |
| 4906 | Базы данных | 3 | 4 |
| 4906 | Теория информации | 2 | 4 |
| 4807 | Сети и телекоммуникации | 2 | 5 |
| 4807 | Английский язык | 1 | 5 |
| 4906 | Английский язык | 1 | 5 |

 Для того чтобы и в этом случае был получен нужный результат, необходимо сделать предварительное преобразование типа данных столбца «Оценка», приведя его к действительному типу, тогда и результат вычисления среднего не будет целым числом. В этом случае запрос будет выглядеть следующим образом:

Select R2.Группа, R1.Дисциплина,Count(\*) as Всего, AVG(cast(Оценка as decimal(3,1))) as Средний\_балл

From R1,R2

where R1. ФИО = R2. ФИО and R1. оценка is not null

and R1. Оценка > 2

Group by R2. Группа, R1. Дисциплина

Здесь функция CAST() выполняет преобразование столбца «Оценка» к действительному типу данных. И теперь результат выполнения этого запроса будет такой же, как и в первом случае на Ms Access.

Мы не можем использовать агрегатные функции в предложении WHERE, потому что условия в этом разделе оцениваются в терминах одиночной строки, а агрегатные функции — в терминах групп строк.

Предложение GROUP BY позволяет определять подмножество значений в особом поле в терминах другого поля и применять функцию агрегата к подмножеству. Это дает возможность объединять поля и агрегатные функции в едином предложении SELECT. Агрегатные функции могут применяться как в выражении вывода результатов строки SELECT, так и в выражении условия обработки сформированных групп HAVING. В этом случае каждая агрегатная функция вычисляется для каждой выделенной группы. Значения, полученные при вычислении агрегатных функций, могут быть использованы для вывода соответствующих результатов или для условия отбора групп.

Построим запрос, который выводит группы, в которых по одной дисциплине на экзаменах получено больше одной двойки:

SELECT R2. Группа

FROM R1,R2

WHERE R1. ФИО = R2. ФИО AND

R1.Оценка = 2

GROUP BY R2.Группа, R1.Дисциплина

HAVING count(\*)> 1

Результат:

|  |
| --- |
| Группа |
| 4906 |

В дальнейшем в качестве примера будем работать не с БД «Сессия», а с БД «Банк», состоящей из одной таблицы F, в которой хранится отношение F, содержащее информацию о счетах в филиалах некоторого банка:

F = <N, ФИО, Филиал, ДатаОткрытия, ДатаЗакрытия, Остаток>;

Q = < Филиал, Город >;

поскольку на этой базе можно ярче проиллюстрировать работу с агрегатными функциями и группировкой.

Например, предположим, что мы хотим найти суммарный остаток на счетах в филиалах. Можно сделать раздельный запрос для каждого из них, выбрав SUM (Остаток) из таблицы для каждого филиала, но операция группировки GROUP BY, позволит поместить их все в одну команду:

SELECT Филиал, SUM(Остаток)

FROM F

GROUP BY Филиал

GROUP BY применяет агрегатные функции независимо для каждой группы, определяемой с помощью значения поля Филиал. Группа состоит из строк с одинаковым значением поля Филиал, и функция SUM применяется отдельно для каждой такой группы, т. е. суммарный остаток на счетах подсчитывается отдельно для каждого филиала. Значение поля, к которому применяется GROUP BY, имеет по определению только одно значение на группу вывода, как и результат работы агрегатной функции. Поэтому мы можем совместить в одном запросе расчет агрегатной функции для группы и вывод поля группировки. В общем случае, как мы уже видели, группировку можно использовать для нескольких полей. При использовании операции группировки мы можем накладывать дополнительные условия на группы, которые попадут в результирующее отношение. Условия отбора групп могут быть заданы с помощью раздела HAVING оператора SELECT.

Предположим, что мы хотели бы увидеть только те филиалы, суммарные значения остатков на счетах которых превышают $5 000. При этом мы хотели бы увидеть и суммарные остатки для выбранных филиалов. Чтобы вывести в результат филиалы, суммарные остатки в которых свыше $5 000, необходимо использовать предложение HAVING. Предложение HAVING определяет критерии, используемые, чтобы удалять определенные группы из вывода, точно так же, как предложение WHERE делает это для индивидуальных строк.

Правильной командой будет следующая:

SELECT Филиал, SUM(Остаток)

FROM F

GROUP BY Филиал

HAVING SUM (Остаток) > 5 000;

Аргументы в предложении HAVING подчиняются тем же самым правилам, что и в предложении SELECT, где используется GROUP BY. Они должны иметь одно значение на группу вывода.

Следующая команда будет запрещена:

SELECT Филиал,SUM(Остаток)

FROM F

GROUP BY Филиал

HAVING ДатаОткрытия = 27/12/2004;

Поле ДатаОткрытия не может быть использовано в предложении HAVING, потому что оно может иметь больше чем одно значение на группу вывода. Чтобы избежать такой ситуации, предложение HAVING должно ссылаться только на агрегаты и поля, выбранные GROUP BY. Имеется правильный способ сделать вышеупомянутый запрос:

SELECT Филиал,SUM(Остаток)

FROM F

WHERE ДатаОткрытия = ’27/12/2004’

GROUP BY Филиал

Смысл данного запроса следующий: найти сумму остатков по каждому филиалу счетов, открытых 27 декабря 2004 года.

Как говорилось ранее, HAVING может использовать только аргументы, которые имеют одно значение на группу вывода. Практически ссылки на агрегатные функции — наиболее общие, но и поля, выбранные с помощью GROUP BY, также допустимы. Например, мы хотим увидеть суммарные остатки на счетах филиалов в Санкт-Петербурге, Пскове и Урюпинске:

SELECT Филиал, SUM(Остаток)

FROM F,Q

WHERE F. Филиал = Q. Филиал

GROUP BY Филиал

HAVING Филиал IN (‘Санкт-Петербург’, ‘Псков’, ‘Урюпинск’);

Поэтому в арифметических выражениях предикатов, входящих в условие выборки раздела HAVING, прямо можно использовать только спецификации столбцов, указанных в качестве столбцов группирования в разделе GROUP BY. Остальные столбцы можно специфицировать только внутри спецификаций агрегатных функций COUNT, SUM, AVG, MIN и MAX, вычисляющих в данном случае некоторое агрегатное значение для всей группы строк.

Результатом выполнения раздела HAVING является сгруппированная таблица, содержащая только те группы строк, для которых результат вычисления условия отбора в части HAVING есть TRUE. В частности, если раздел HAVING присутствует в запросе, не содержащем GROUP BY, то результатом его выполнения будет либо пустая таблица, либо результат выполнения предыдущих разделов табличного выражения, рассматриваемый как одна группа без столбцов группирования. Рассмотрим пример. Допустим, мы хотим вывести общую сумму остатков по всем филиалам, но только в том случае, если она более $100 000. В этом случае наш запрос не будет содержать операции группировки, но будет содержать раздел HAVING и будет выглядеть следующим образом:

SELECT SUM(Остаток)

FROM F

HAVING SUM(Остаток) > 100 000;

Если суммарный остаток более чем $100 000, то мы его увидим в результирующем отношении, в противном случае мы получим пустое отношение.

При использовании агрегатных функций необходимо помнить, что в результирующем наборе могут присутствовать только значения полей группировки и возможно значения агрегатных функций. Не допустимо группировать по одним значениям, а выводить другие значения. Это будет синтаксической ошибкой.

Например, такой запрос всегда будет ошибочным:

Select A

From T

Group By B

1. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.

### Методические рекомендации (указания) к выполнению самостоятельной работы №6

1. Тема самостоятельной работы: Формирование запросов SQL, создание форм и отчетов.
2. Количество часов: 8ч.
3. Задания для самостоятельной работы: выполнение индивидуальных заданий
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 2.1 Организация запросов SQL», создание условий для формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

1. Рекомендации преподавателя:

**Задание 1:**

Выполнить запрос 3.12 с использованием различных предикатов предложения SELECT инструкции SQL.

Описание предикатов SELECT [ALL | DISTINCT | DISTINCTROW | [TOP n [PERCENT]]] в приложении 1 и справке MSAccess.

Минимальный синтаксис инструкции SELECT выглядит следующим образом.

**SELECT** *поля* **FROM** *таблица*

Назначить необходимые свойства запроса можно как в Свойствах запроса (Рис.60), так и при помощи предикатов инструкции SELECT(Рис.61).

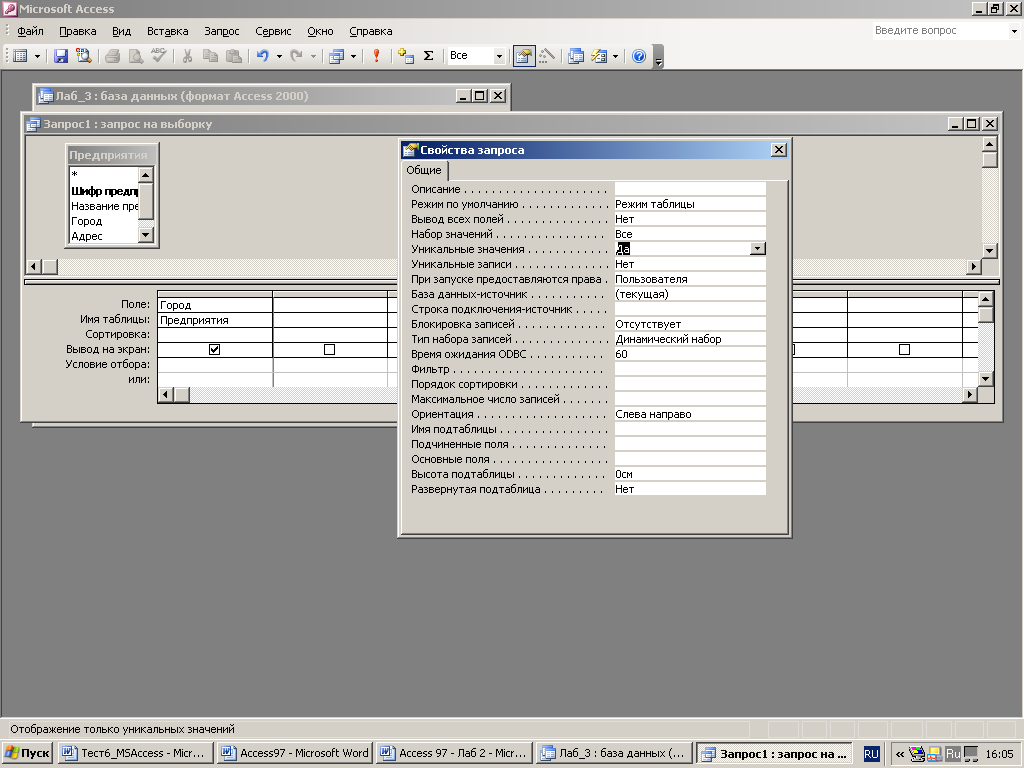
**

Рис. Перечень городов, в которых находятся предприятия (без повторений), установленный как свойство запроса «Уникальные значения»

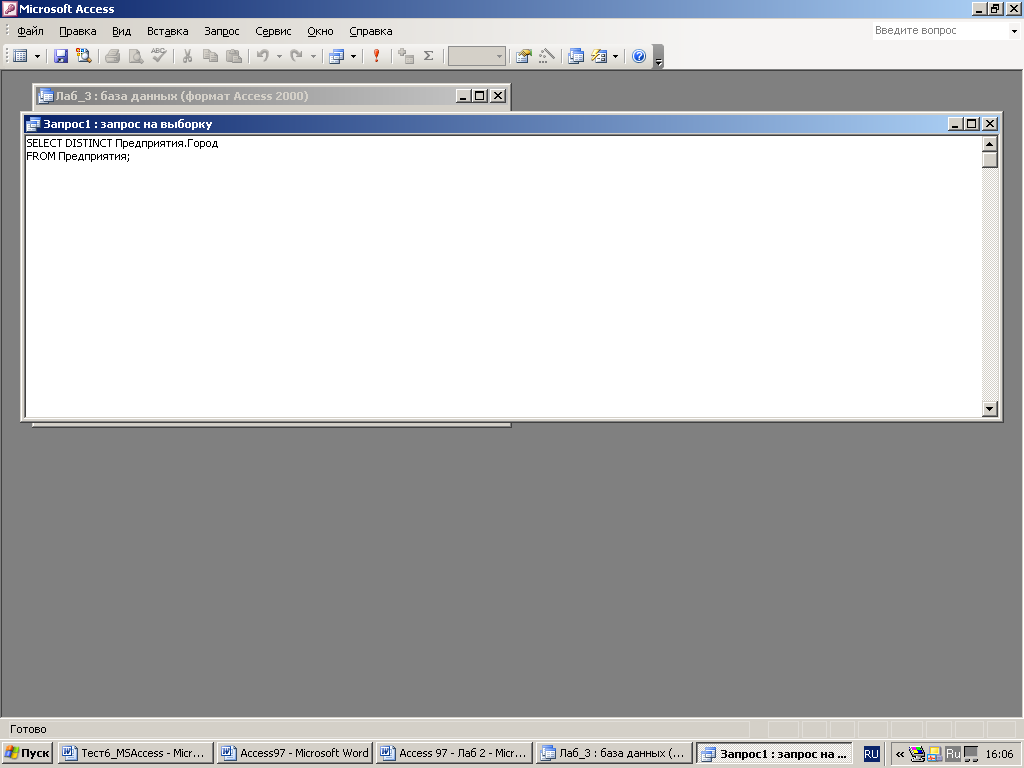
**

Рис.Перечень городов, в которых находятся предприятия (без повторений), заданный предикатом DISTINCT конструкции SELECT.

Как первый так и второй способ задания уникальных значений приводит к одному и тому же результату (Рис.2).

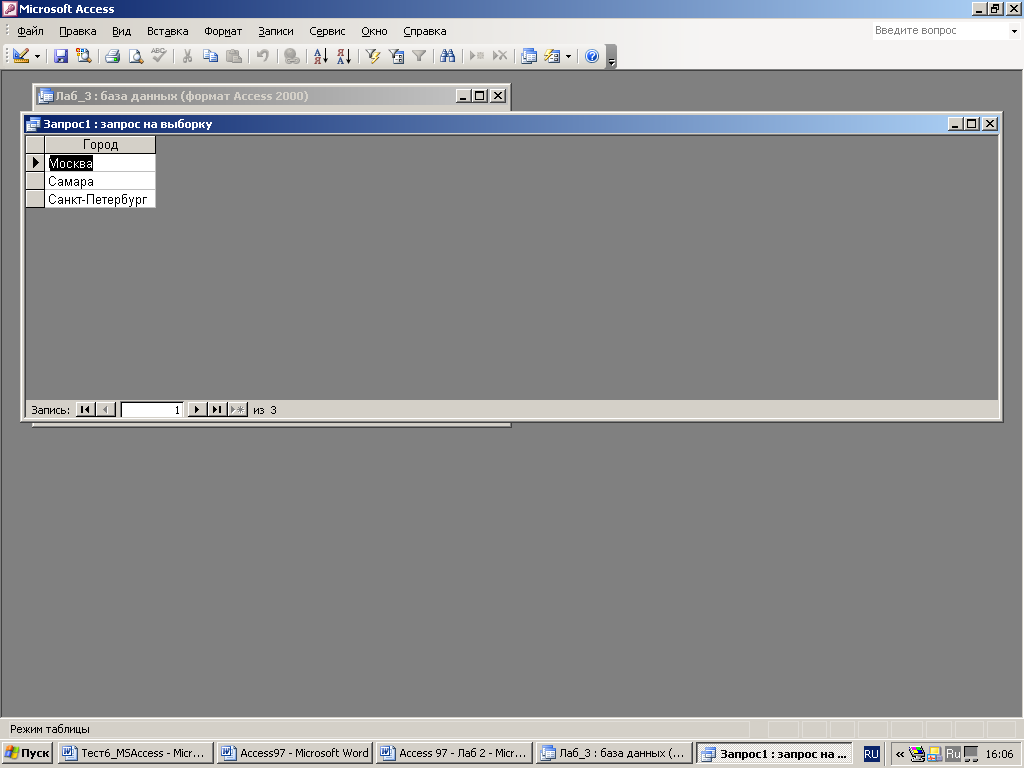
**

Рис.Перечень городов, в которых находятся предприятия (без повторений)

**Задание 2:**

Для запроса выполнить различные типы объединений и обосновать разницу в результатах. (INNER JOIN, LEFT JOIN, RIGHT JOIN).

**Задание :**

Для запроса опробовать действия различных предикатов предложения SELECT инструкции SQL и обосновать разницу в результате.

**Задание 4:**

Вывести адрес и телефон предприятия с запросом у пользователя шифра этого предприятия.

Последовательность выполнения этого задания будет следующая:

1. Формируем запрос на выборку, включив в него 3 поля, одно из которых будет полем для запроса параметра, а два других будут выводиться на экран (Рис. 4).

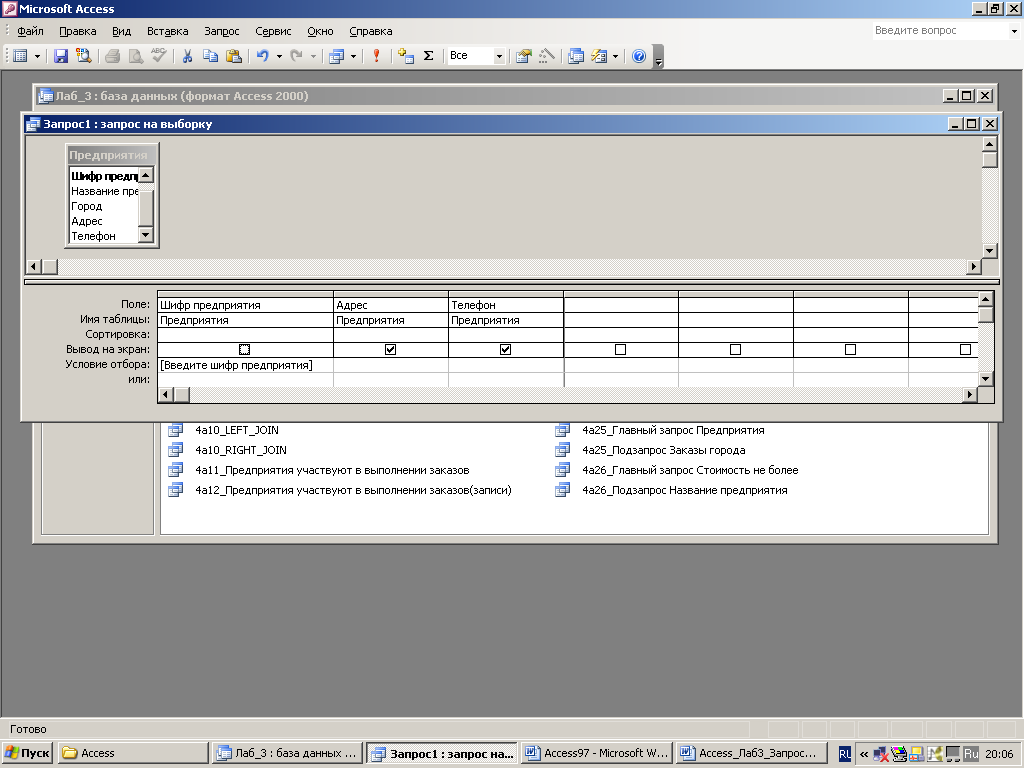
**

Рис.Формирование запроса с параметрами (в режиме Конструктора).

1. В «Условие отбора» вводим текст, который будет выводиться в окне запроса с параметром (при двойном клике по запросу) (Рис. 5)

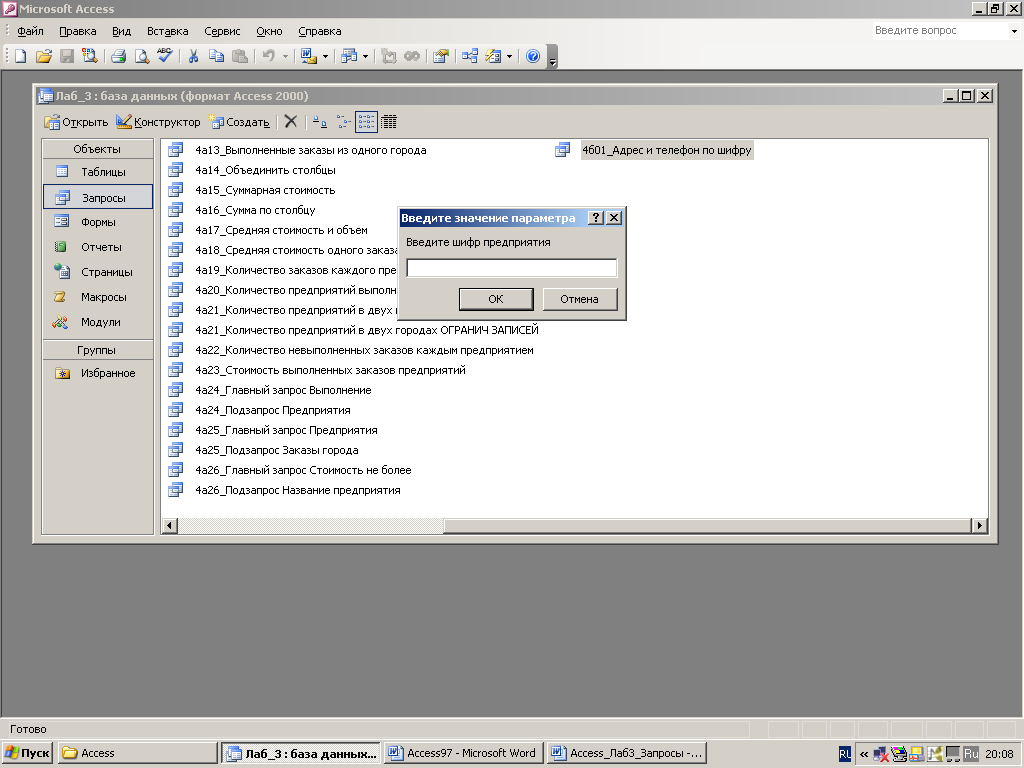
**

Рис.Работа запроса с параметром

1. В результате ввода параметра выводится таблица с указанным в запросе числом полей (Рис.6)

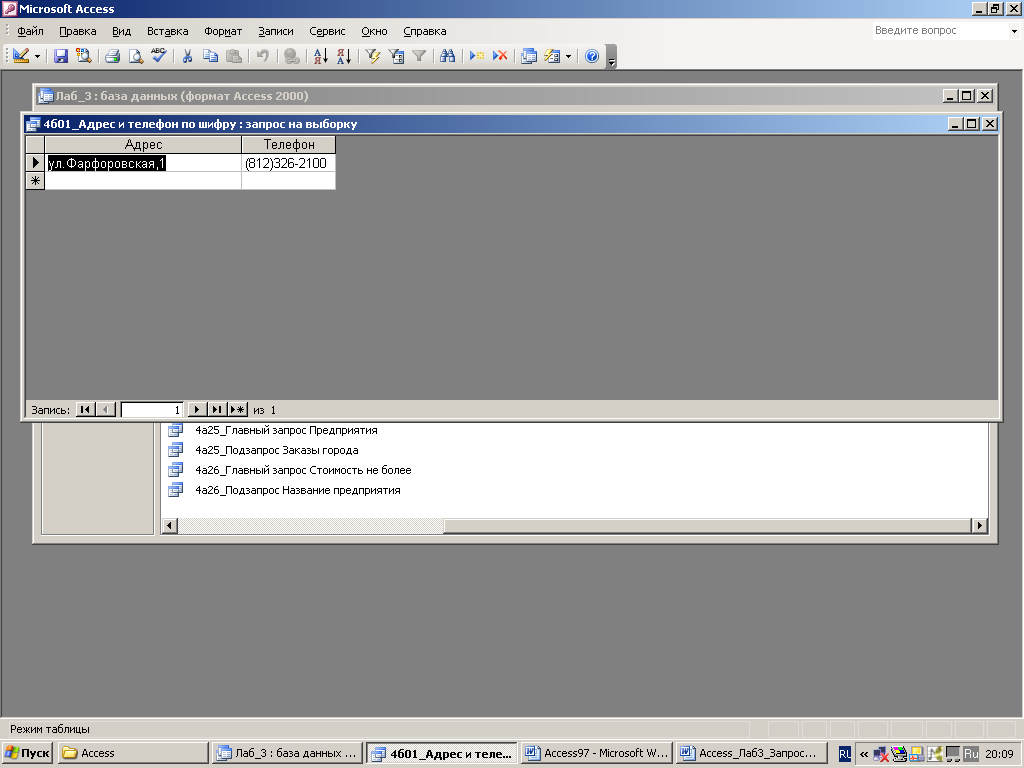
**

Рис.Результат работы запроса с параметром.

**Задание 5:**

Вывести информацию о заказах, в шифрах которых имеются буквы, задаваемые пользователем.

**Задание 6:**

Вывести шифры и объемы заказов, у которых диапазон даты отгрузки задается пользователем.

**Задание 7:**

Вывести информацию о заказах, выполняемых определенным предприятием (шифр предприятия задается пользователем) и дата выполнения которых позже даты, указываемой пользователем.

1. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.

### Методические рекомендации (указания) к выполнению самостоятельной работы №7

1. Тема самостоятельной работы: Создание и заполнение таблиц базы данных.
2. Количество часов: 8ч.
3. Задания для самостоятельной работы: создание и заполнение таблиц базы данных
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 3.2 Курсовое проектирование», создание условий для формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

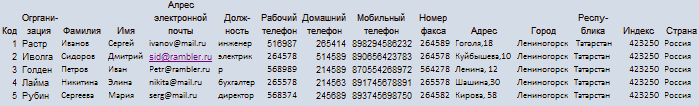
1. Рекомендации преподавателя:

**Задание 1. Создание пустой базы данных с помощью шаблонов таблиц.**

***Порядок работы***

1. Запустите программу СУБД Microsoft Access. Для этого выполните: *Пуск – Все программы – Microsoft office – Microsoft office Access 201).*
2. Перед Вами откроется окно следующего вида *(Рисунок 1)*:
3. Выберите команду *Новая база данных*. Затем введите *имя файла –База работников* и нажмите кнопку *Создать*. Перед Вами откроется окно следующего вида *(Рисунок 2)*.
4. Выберите команду *Создание*– *Части приложения. Шаблоны – Контакты (Рисунок 3).*
5. В левой панели появляется таблица *Контакты.*Щелкните дважды мышью по имени таблице. Перед вами откроется вся таблица *Контакты* с заголовками *(Рисунок 3)*.
6. Переименуйте поля *ИД, Область, край, Страна или регион* на следующие новые имена полей соответственно: *Код, Республика, Страна.*
7. Все поля после поля *Страна* удалите с помощью контекстного меню, выполнив команду *Удалить поле.*
8. Заполните ее следующими данными (Таблица 1).

Таблица 1.



1. У Вас должна получится таблица как на рисунке *(Рисунок 4)*. Сохраните таблицу (https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8281.gif) под именем *Работник.*
2. В данной таблице отсортируйте столбец “*Организация*” по алфавиту (*Главная* – https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8282.gif).

**Задание 2. Создание пустой базы данных с помощью конструктора таблиц.**

***Порядок работы***

1. Создадим таблицу под именем “*Студент*” с помощью конструктора таблиц.

Для этого выполните команду: *Создание – конструктор таблиц*.

Перед Вами откроется окно *(Рисунок 5)*:

1. Заполните *Имя поля* следующими данными (заголовками столбцов): *КодСтудент, Фамилия, Имя, Отчество, Адрес, Номер телефона, Специализация* соответственно типы данных для полей: *КодСтудент –*СЧЕТЧИК, *Фамилия, Имя, Отчество, Должность, Адрес, Специализация* – ТЕКСТОВЫЙ, *Номер телефона –*ЧИСЛОВОЙ.

У Вас должно получиться как на рисунке *(Рисунок 6):*

1. Далее Нажмите сохранить (https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8281.gif) и назовите таблицу “*Студент*”. Он автоматически запросит создать ключевое поле, нажмите кнопку ДА (поле *КодСтудент* будет *Ключевое поле* https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8283.gif).
2. Затем двойным щелчком левой кнопкой мыши откройте слева на таблицу *Студент.*Перед Вами откроется таблица *Студент* для заполнения *Рисунок 7*).
3. Заполните эту таблицу следующими данными (*Таблица 2*) и сохраните ее.

Таблица 2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **КодСтудент** | **Фамилия** | **Имя** | **Отчество** | **Адрес** | **Номер телефона** | **Специализация** |
| 1 | Иванов | Сергей | Александрович | г. Новороссийск | 457896 | технолог |
| 2 | Петров | Сергей | Петрович | г. Москва | 7458962 | технолог |
| 3 | Гаврелеева | Ольга | Ивановна | г. Москва | 3698521 | бухгалтер |
| 4 | Соколова | Инна | Олеговна | г. Новороссийск | 852967 | бухгалтер |
| 5 | Мухина | Олеся | Петровна | г. Москва | 8625471 | технолог |
| 6 | Апареева | Анна | Романовна | г. Люберцы | 748596 | технолог |
| 7 | Глинкина | Дина | Евгеньевна | г. Люберцы | 919597 | технолог |
| 8 | Сорина | Ольга | Сергеевна | г. Москва | 9191954 | бухгалтер |

1. Результаты покажите преподавателю.

**Создание базы данных, состоящей из двух таблиц**

*Цель работы:*

1. *Проектирование структуры базы данных.* Создать вариант базы данных с двумя таблицами: “Список”” и “ Группы”.
2. *Конструирование пустых таблиц базы данных.*
3. *Создание схемы базы данных.* В данном случае таблицы связаны связью один-ко-многим. Это значит, что в таблице “Группы” каждое значение связанного поля может встречаться только один раз, а в таблице “Список” – несколько раз (несколько человек могут быть из одной группы). Связи следует устанавливать при пустых таблицах.
4. *Ввод данных в таблицы.* Создать форму для ввода данных.
5. ***Порядок работы.***

**Задание 1. Создание базы данных – *Учебная база*.**

1. Запустите программу СУБД Microsoft Access. Для этого выполните: *Пуск – Все программы – Microsoft office – Microsoft office Access 2010.*
2. Перед Вами откроется окно следующего вида (*Рисунок 8*):
3. Выберите команду *Новая база данных*. Затем введите *имя файла –Учебная база* и нажмите кнопку *Создать*. Перед Вами откроется окно следующего вида (*Рисунок 9*):

**Задание 2. Создание таблицы “Группы”.**

1. Создайте таблицу под именем “Группы” с помощью конструктора таблиц. Для этого выполните команду: *Создание – конструктор таблиц*.
2. Заполните *Имя поля* следующими данными (заголовками столбцов): *Учебная группа, Преподаватель и* соответственно Тип данных для них: *Учебная группа –*ЧИСЛОВОЙ, *Преподаватель*– ТЕКСТОВЫЙ. Сделайте поле “Учебная группа” ключевым, установив курсор на имя поля и щелкнув по кнопке **Ключевое поле https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8284.gif.**
3. Сохраните таблицу под именем “Группы”, щелкнув по кнопке **Сохранить** https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8285.gif.
4. Закройте таблицу.

**Задание 3. Создание таблицы “Список”.**

1. Создайте таблицу под именем “Список” с помощью конструктора таблиц. Для этого выполните команду: *Создание – конструктор таблиц*.
2. Заполните *Имя поля* следующими данными (заголовками столбцов): Код, Фамилия, Имя, Отчество, Год рождения, Класс, Учебная группа. Выберите из ниспадающего списка числовой тип данных для поля “Код”.Сделайте поле “Код” ключевым, установив курсор на имя поля и щелкнув по кнопке **Ключевое поле https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8284.gif.**Тип данных полей “Фамилия”, “Имя”, “Отчество” – текстовый, полей “Год рождения”, “Школа”, “Класс” – числовой. Установите тип данных поля “Учебная группа” числовой. Общие свойства поля не меняйте. Выберите вкладку **Подстановка**, тип элемента управления – **Поле со списком**, источник строк – **Группы**. Получите значения элементов, как показано на рисунке *(Рисунок 10)*.
3. Сохраните таблицу под именем “Список”, щелкнув по кнопке **Сохранить** https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8285.gif.
4. Закройте таблицу.

**Задание 4. Создание схемы данных.**

1. В ленточном меню выберите вкладку **Работа с базами данных,**щелкните по кнопке **Схема данных** Image8286.gif (1033 bytes)**.**Появится окно **Схема данных.**
2. В появившемся окне **Добавление таблицы** выделите таблицу “Группы” и щелкните по кнопке **Добавить,**выделите таблицу “Список” и щелкните по кнопке **Добавить.**В окне**Схема данных** появится условный вид этих таблиц. Щелкните по кнопке **Закрыть** окна **Добавление таблицы.**
3. Увеличьте окно таблицы “Список” так, чтобы были видны все поля.
4. Установите курсор мыши на имя поля “Учебная группа” в таблице “Группы” и, не отпуская кнопку мыши, перетащите курсор мыши на поле “Учебная группа” таблицы “Список”. Отпустите кнопку мыши. Появится диалоговое окно **Изменение связей**, представленное на рисунке *(Рисунок 11)*.
5. Установите флажок **Обеспечение целостности данных.** Это невозможно будет сделать, если типы полей “Учебная группа” заданы не одинаково.
6. Установите флажок **каскадное обновление связанных полей**. Это приведет к тому, что при изменении номера группы в таблице “Группы” автоматически изменится соответствующий номер в таблице “Список”.
7. Установите флажок **каскадное удаление связей**. Это приведет к тому, что при удалении записи с номером группы из таблицы “Группы”, будут удалены все записи из таблицы “Список”, в которых находились соответствующие номера групп.
8. Щелкните по кнопке **Создать**. Появится связь “один-ко-многим”. Схема данных представлена на рисунке *(Рисунок 12)*.
9. Закройте схему данных, щелкнув по кнопке https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8287.gif в верхнем правом углу окна и ответив утвердительно на вопрос о сохранении схемы данных.

**Задание 5. Заполнение таблицы “Группы” значениями.**

1. Откройте таблицу “Группы” в режиме **Таблицы**.
2. Заполните ее записями из таблицы 1.

Таблица 1

|  |  |
| --- | --- |
| **Учебная группа** | **Преподаватель** |
| 101 | Смирнова З.В. |
| 102 | Вильданова Р.Ш. |
| 103 | Зайцева С.А. |
| 104 | Зазулина И.В. |
| 105 | Друк Н.Н. |

**Задание 6. Создание формы для ввода данных.**

1. В окне базы данных выделите имя таблицы **Список**и выполните команду *Создание – Форма.*
2. Появится пустая форма ввода, представленная на рисунке *(Рисунок 13).*

Заполните форму данными, приведенными в таблице 2.

Таблица 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Код | Фамилия | Имя | Отчество | Год рождения | Школа | Класс | Учебная группа |
| 1 | Чернова | Наталья | Алексеевна | 2001 | 5 | 9 | 101 |
| 2 | Куликов | Клим | Алексеевич | 1999 | 2 | 10 | 103 |
| 3 | Архипов | Виктор | Александрович | 2000 | 5 | 9 | 101 |
| 4 | Баранкова | Александра | Николаевна | 1998 | 5 | 11 | 104 |
| 5 | Новоселов | Алексей | Иванович | 2000 | 3 | 9 | 105 |

1. Сохраните введенные данные. Имя формы – **Список.** Закройте форму.
2. Перейдите в окно **Таблицы.** Откройте таблицу **Список**. Убедитесь, что в таблице появились новые записи.
3. Щелкнув по кнопке **Сохранить,**сохраните текущую таблицу.
4. Закройте таблицу.

**Задание 7. Проверка каскадного обновления связанных полей.**

1. Откройте таблицу “Группы”.
2. Исправьте номера учебных групп на 201, 202, 203, 204, 205. Сохраните таблицу.
3. Откройте таблицу “Список”. Убедитесь, что значения поля “Учебная группа” изменились. Закройте таблицу “Список”.

**Задание 8. Проверка каскадного удаления связанных полей.**

1. Откройте таблицу “Группы”.
2. Удалите первую запись (Выделите первую строку, правой кнопкой мыши вызовите контекстное меню и выберите команду *Удалить запись.*
3. Ответьте утвердительно на запрос об удалении. Закройте таблицу “Группы”.
4. Откройте таблицу “Список”. Убедитесь, что исчезли записи с номером группы 201.

**Предъявите преподавателю:**таблицу **Групп**ы, таблицу **Списо**к на экране, форму **Список**.

**Задание 9. Завершение работы с программой Access.**

1. Выполните команду *Файл – Выход.*
2. Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Ответьте утвердительно.
3. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.

### Методические рекомендации (указания) к выполнению самостоятельной работы №8

1. Тема самостоятельной работы: Создание запросов.
2. Количество часов: 8ч.
3. Задания для самостоятельной работы: создание запросов
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 3.2 Курсовое проектирование», создание условий для формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

1. Рекомендации преподавателя:

**Создание запросов к готовой базе данных.**

**Задание 1. Открытие базы данных, изготовленной на прошлом занятии.**

1. В папке **Для уроков** в своей личной папке откройте файл **База работников.**
2. На экране появится окно с основными элементами базы данных. В базе данных должны быть две таблицы: **Работник** и **Студент**.

**Задание 2. Создание запроса на выборку.**

1. Выполните команду **Создание – Конструктор запросов.**
2. В появившемся диалоговом окне **Добавление таблицы**выберите из списка имя таблицы **Студент**, щелкните по кнопке **Добавить***(Рисунок 14)*.
3. Закончите выбор, щелкнув по кнопке **Закрыть.** Появится возможность выбора полей из таблицы “Студент”. Для этого достаточно дважды щелкнуть по именам полей или перетащить мышью названия полей в клетку запроса.
4. Создайте телефонную книгу для всех студентов, фамилии которых начинаются на букву **С.**Для этого в поле***Условие отбора*** напишите условие **Like “С\*”***(Рисунок 15):*.
5. Сохраните запрос, щелкнув по кнопке **Сохранить**. Введите имя запроса **Телефонная книга**и щелкните по кнопке**ОК.**
6. Щелкните по кнопке **Выполнить**для представления запроса. Закройте запрос.
7. Убедитесь в правильности полученного запроса, щелкнув по имени запроса **Телефонная книга** слева в окне **Все объекты Access.**Закройте таблицу.
8. Создайте запрос на выборку тех студентов, которые приехали из Москвы или Люберцы.
9. Для этого выполите команду **Создание – Конструктор запросов.**
10. В появившемся диалоговом окне **Добавление таблицы**выберите из списка имя таблицы **Студент**, щелкните по кнопке **Добавить***(Рисунок 14).*
11. Закончите выбор, щелкнув по кнопке **Закрыть.** Появится возможность выбора полей из таблицы “Студент”. Для этого достаточно дважды щелкнуть по именам полей или перетащить мышью названия полей в клетку запроса.
12. В поле***Условие отбора*** напишите условия для поля **Адрес** так, как показано на рисунке *(Рисунок 16):*
13. Сохраните запрос, щелкнув по кнопке **Сохранить**. Введите имя запроса **Адрес**и щелкните по кнопке**ОК.**
14. Щелкните по кнопке **Выполнить** для представления запроса. Закройте запрос.

**Самостоятельное задание.**

1. Составьте запрос на выборку тех студенток, имя которых – Ольга.
2. Составьте запрос на выборку работников организаций, названия которых начинаются на букву **Р,**используя таблицу**Работник.**
3. Составьте запрос на выборку всех студентов, которые обучаются по специальности технолога.
4. Составьте запрос на выборку работников организаций, которые работают по должности **инженер**или **бухгалтер**.
5. Результаты предъявите учителю.

**Задание 3. Завершение работы с программой Access.**

1. Выполните команду *Файл – Выход.*
2. Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Ответьте утвердительно.

**Создание и использование запросов.**

**Задание 1.**

1. Откройте базу данных “Учебная база” из своей личной папки, которая должна находиться в папке *Для уроков* на *Рабочем столе*.
2. В базе данных должны быть две связанные таблицы: **Группы** и **Список**.

**Задание 2. Коррекция данных в таблице “Группы”.**

1. Откройте таблицу **Группы**, выделив ее и щелкнув по кнопке ***Открыть.***
2. Добавьте недостающие записи. Исправьте существующие. В предыдущей работе с “Учебной базой” Вы удаляли первую запись. Необходимо ее восстановить. Вы исправляли номера учебных групп на 201, 202, 203, 204, 205. Верните первоначальные значения группам. Добавьте еще одну запись в таблицу **Группы**: Учебная группа – **106**, Преподаватель – Ф.И.О. вашего классного руководителя, например, как на рисунке *(Рисунок 17)*:
3. Закройте таблицу Группы, предварительно сохранив ее.

**Задание 3. Коррекция данных в таблице “Список”.**

1. Откройте таблицу **Список**, выделив ее и щелкнув по кнопке ***Открыть.***
2. Восстановите недостающие две записи. Добавьте еще три новые записи, содержащие информацию о ваших одноклассниках и о Вас, например *(Рисунок 18):*.
3. Перейдите в режим ***Конструктор***. Добавьте еще два поля – **Номер телефона**и**Адрес**, тип оставьте ***Текстовый***. Сохраните изменения, щелкнув по кнопке https://urok.1sept.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/645807/Image8281.gif ***Сохранить***.
4. Перейдите в режим Таблицы и заполните соответствующими данными поля **Номер** **телефона**и **Адрес**.
5. Закройте таблицу **Список**, предварительно сохранив ее.

**Задание 4. Создание запроса на выборку.**

1. Выполните команду **Создание – Конструктор запросов.**
2. В появившемся диалоговом окне **Добавление таблицы**выберите из списка имя таблицы **Список**, щелкните по кнопке **Добавить**.
3. Закончите выбор, щелкнув по кнопке **Закрыть.** Появится возможность выбора полей из таблицы “**Список**”. Для этого достаточно дважды щелкнуть по именам полей или перетащить мышью названия полей в клетку запроса.
4. Создайте телефонную книгу для всех учащихся, имена которых начинаются на букву А. Для этого в поле***Условие отбора*** напишите условие **Like “А\*”***(Рисунок 19):*
5. Сохраните запрос, щелкнув по кнопке **Сохранить**. Введите имя запроса **Номера телефонов**и щелкните по кнопке**ОК.**
6. Щелкните по кнопке **Выполнить**для представления запроса. Закройте запрос.
7. Убедитесь в правильности полученного запроса, щелкнув по имени запроса **Номера телефонов** слева в окне **Все объекты Access.**Закройте таблицу.
8. Создайте запрос на выборку учащихся 9-х классов, которых обучает преподаватель Смирнова З.В.
9. Для этого выполите команду **Создание – Конструктор запросов.**
10. В появившемся диалоговом окне **Добавление таблицы**выберите из списка имена таблиц **Список**и**Группы**, щелкая по кнопке **Добавить***(Рисунок 20).*
11. Закончите выбор, щелкнув по кнопке **Закрыть.** Появится возможность выбора полей из таблиц **Список** и **Группы**. Для этого достаточно дважды щелкнуть по именам полей или перетащить мышью названия полей в клетку запроса.
12. В поле***Условие отбора*** напишите условия для поля **Преподаватель**из таблицы**Группы**и поля**Класс**таблицы **Список** так, как показано на рисунке *(Рисунок 20):*
13. Сохраните запрос, щелкнув по кнопке **Сохранить**. Введите имя запроса **Смирнова**и щелкните по кнопке**ОК.**
14. Щелкните по кнопке для представления запроса. Закройте запрос.

**Задание 4. Создание запроса с использованием логических операций в условии отбора.**

1. Создайте запрос на выборку тех преподавателей, которые обучают учащихся школ в 102 или 103 группе из школы №2 и №3.
2. Для этого выполите команду **Создание – Конструктор запросов.**
3. В появившемся диалоговом окне **Добавление таблицы**выберите из списка имена таблиц **Список**и**Группы**, щелкая по кнопке **Добавить.**
4. Закончите выбор, щелкнув по кнопке **Закрыть.** Появится возможность выбора полей из таблиц **Список** и **Группы**. Для этого достаточно дважды щелкнуть по именам полей или перетащить мышью названия полей в клетку запроса.

В поле***Условие отбора*** напишите условия для полей **Школа**и **Учебная группа** таблицы **Список** так, как показано на рисунке *(Рисунок 21):*

1. Сохраните запрос, щелкнув по кнопке **Сохранить**. Введите имя запроса **Преподаватель**и щелкните по кнопке**ОК.**
2. Щелкните по кнопке для представления запроса. Закройте запрос.

**Самостоятельное задание.**

1. Составьте запрос на выборку фамилий только тех учеников, которые родились в 1999 и 2000 годах.
2. Составьте запрос на выборку учащихся школы №5, которые учатся в 9 и 11 классах.
3. Составьте запрос на выборку всех фамилий от **Куликова** до конца алфавита.

**Задание 5. Создание запроса на выборку с использованием построителя.**

1. Выделите запрос **Номера телефонов** и откройте его в режиме **Конструктор** с помощью контекстного меню (правой кнопкой мыши).
2. Смените поле **Номер телефона** на поле **Адрес**.
3. Сохраните запрос под именем **Адрес**, выполнив команду **Файл – Сохранить объект как.**
4. Установите курсор в строку **Условие отбора** под столбцом **Фамилия**.
5. Щелкните по кнопке **Построитель**. Появится окно, в котором можно строить сложные запросы *(Рисунок 22)*.
6. В столбце **Элементы выражений** щелкните по кнопке **Операторы**, затем в столбце **Категории выражений** выберите **Логические**, в столбце **Значения выражений** щелкните по кнопке **Not**. Это слово появится в верхнем окне. Фамилию **Архипов** в кавычках введите вручную.
7. Щелкните по кнопке **ОК**. В строке ***Условие отбора*** появится данное выражение.
8. Щелкните по кнопке для представления запроса.
9. Закройте запрос, сохранив его под именем **Не\_Архипов**.

**Предъявите преподавателю** на экране запросы: Номера телефонов, Смирнова, Преподаватель, Год рождения, Школа 5, Куликов, Адрес, Не\_Архипов.

**Задание 6**. **Завершение работы с программой Access.**

1. Выполните команду *Файл – Выход.*
2. Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Ответьте утвердительно.
3. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.

### Методические рекомендации (указания) к выполнению самостоятельной работы №9

1. Тема самостоятельной работы: Создание форм.
2. Количество часов: 8ч.
3. Задания для самостоятельной работы: создание форм
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 3.2 Курсовое проектирование», создание условий для формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

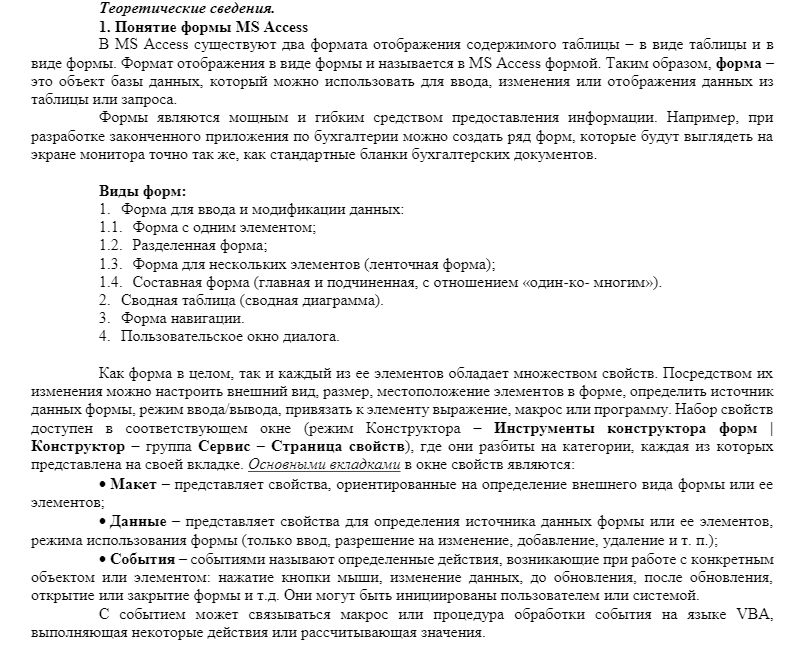
ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

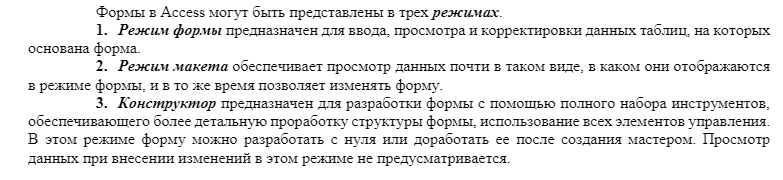
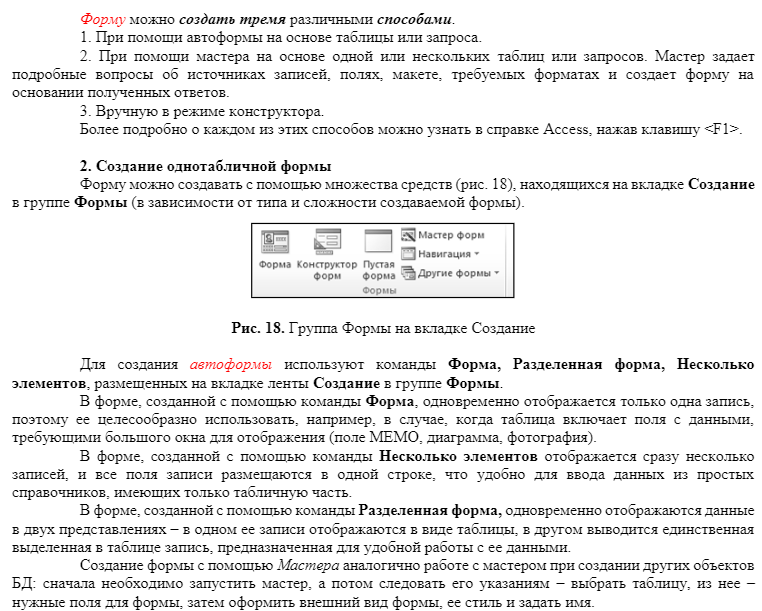
ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

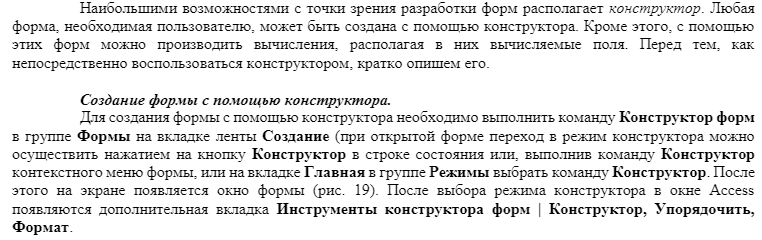
ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

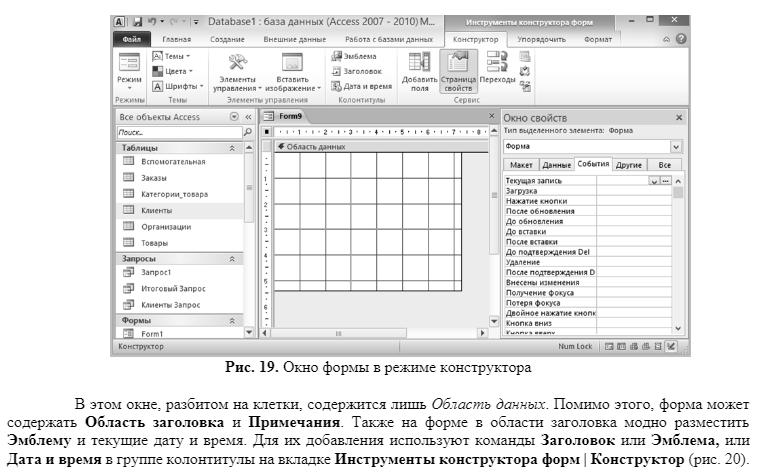
ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

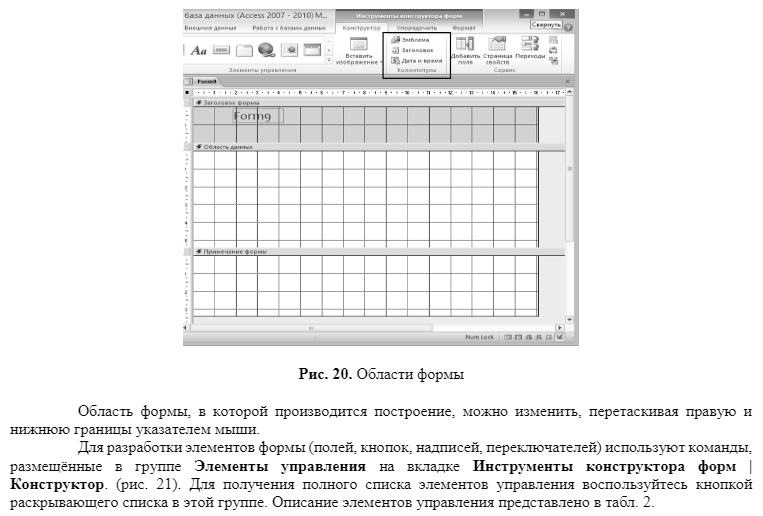
1. Рекомендации преподавателя:

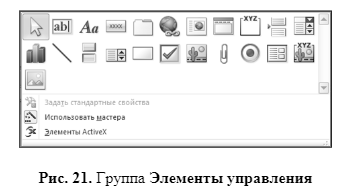


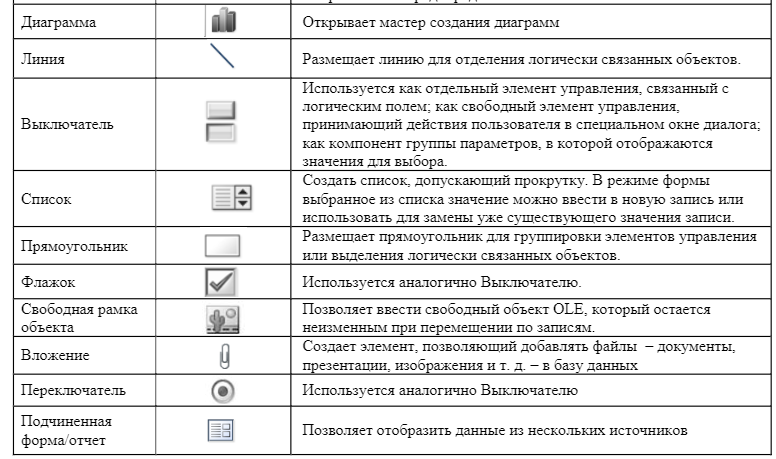


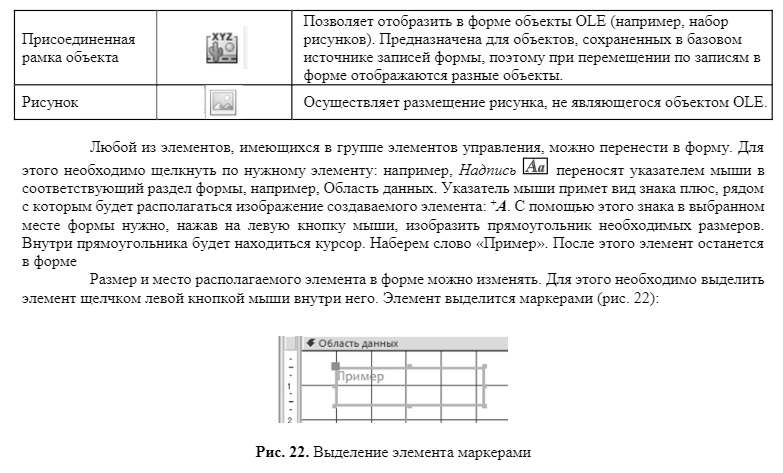


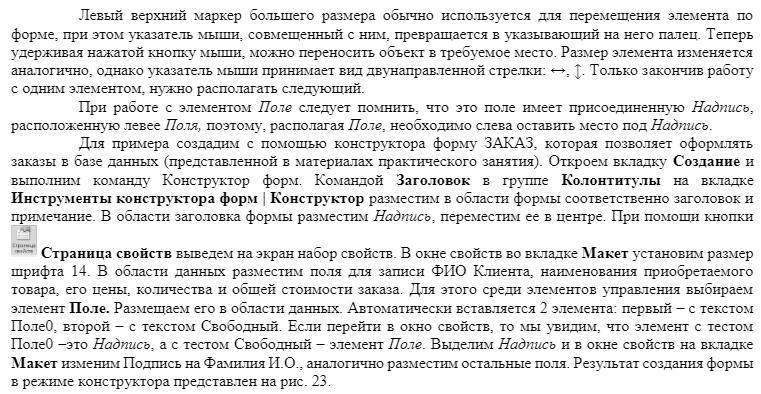


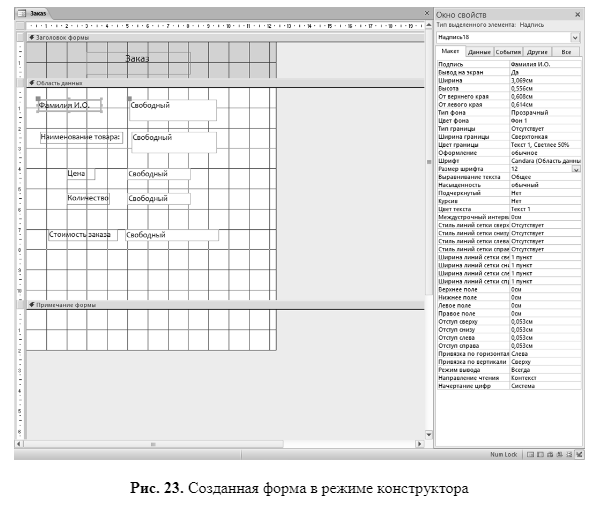


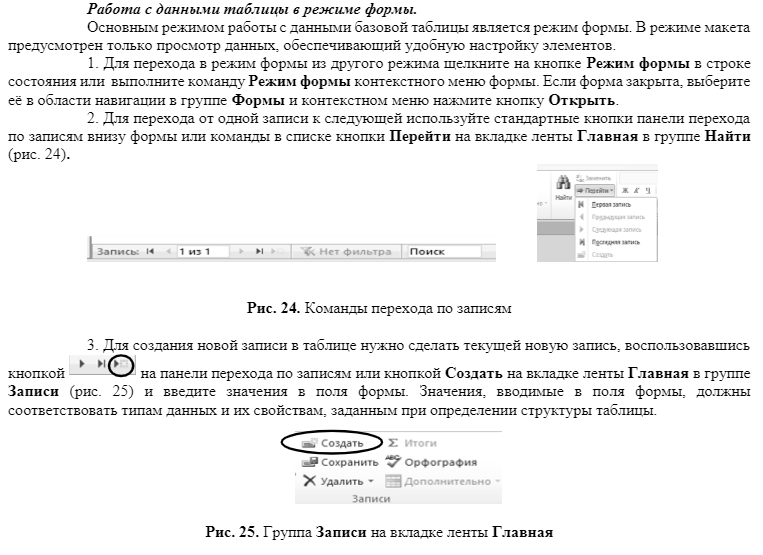


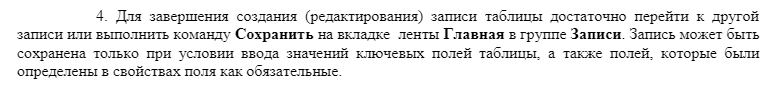


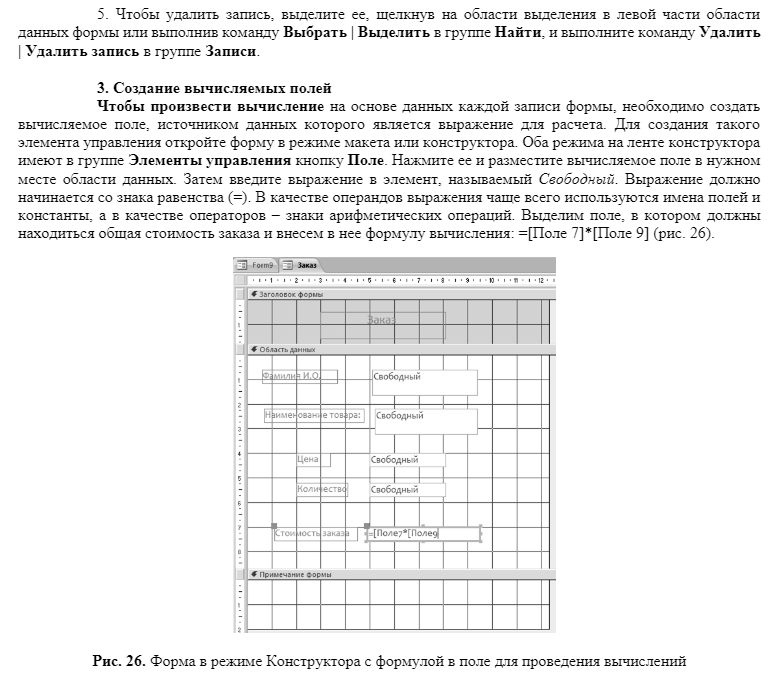


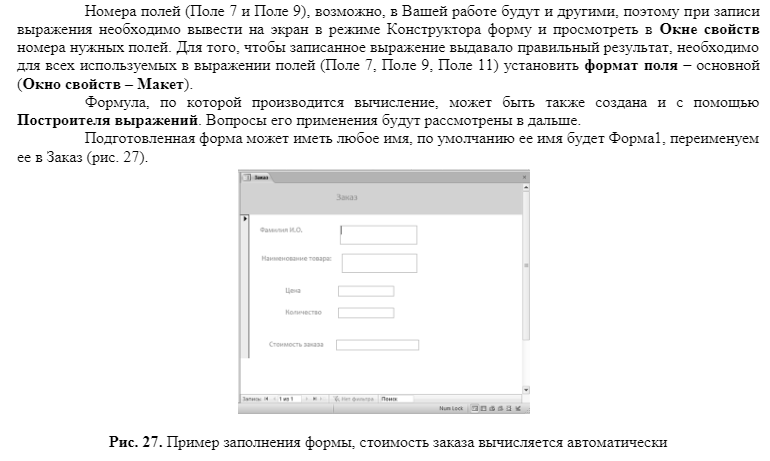


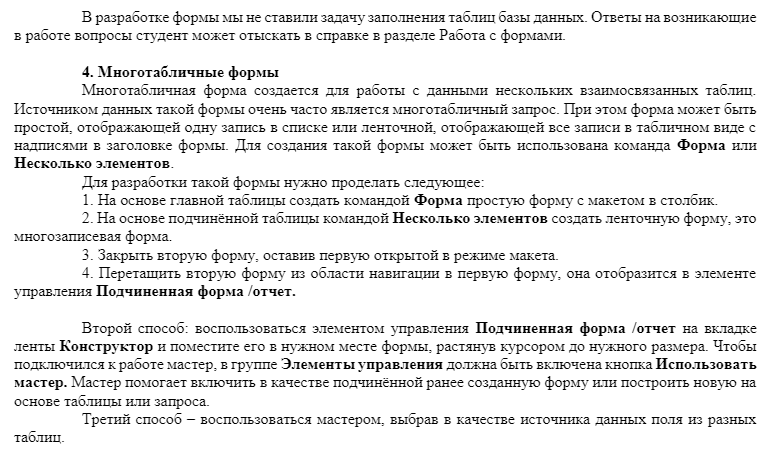




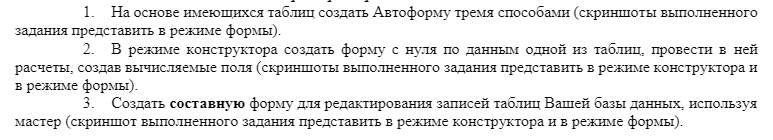








Задания для самостоятельной работы:



1. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.

### Методические рекомендации (указания) к выполнению самостоятельной работы №10

1. Тема самостоятельной работы: Создание макросов для автоматизации работы с БД.
2. Количество часов: 7ч.
3. Задания для самостоятельной работы: создание макросов для автоматизации работы с БД
4. Цель самостоятельной работы: формирование умений и знаний по теме

«Тема 3.2 Курсовое проектирование», создание условий для формирования компетенций:

ОК 2 - организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество,

ОК 4 - осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития,

ОК 5 - владеть информационной культурой, анализировать и оценивать информацию с использованием информационно- коммуникационных технологий,

ОК 8- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации,

ОК 9 - ориентироваться в условиях частой смены технологий в профессиональной деятельности.

1. Рекомендации преподавателя:

Макрос – это средство для автоматизации задач и добавления функциональных возможностей в формы, отчеты и элементы управления.

В приложении Aceess макросы можно рассматривать как упрощенный язык программирования, на котором программа записывается в виде списка макрокоманд для выполнения. При создании макроса каждая макрокоманда выбирается из списка после чего к ней добавляется необходимая информация.

Макрос создается с помощью конструктора макросов. Чтобы отобразить построитель макросов на вкладке Создание в группе Макросы и код щелкните Макрос.

**1.Создание макроса пользовательского интерфейса**

* Microsoft Access 2010 макросы, связанные с объектами пользовательского интерфейса (такими как кнопки, текстовые поля, формы и отчеты), называются макросами пользовательского интерфейса. Этим они отличаются от макросов данных, которые связываются с таблицами. С помощью макросов пользовательского интерфейса можно автоматизировать последовательности действий, таких как открытие другого объекта, применение фильтра, запуск операции экспорта и многие другие задачи. В этой статье описан измененный конструктор макросов и продемонстрировано выполнение основных задач, связанных с созданием макросов пользовательского интерфейса.

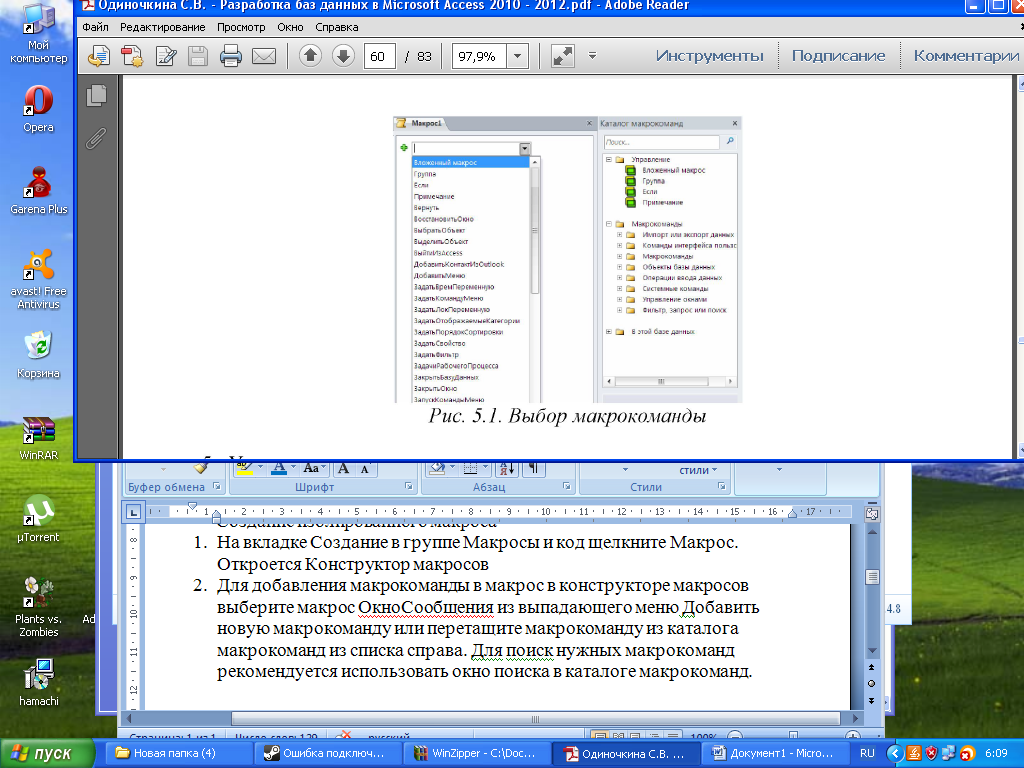
Макросы могут содержаться в объектах макроса (иногда их называют изолированными макросами) либо могут быть внедрены в свойства событий форм, отчетов или элементов управления. Внедренные макросы становятся частью объекта или элемента управления. Объекты макроса отображаются в области навигации в группе Макросы; внедренные макросы не отображаются. Каждый макрос состоит из одной или нескольких макрокоманд. В зависимости от текущего контекста некоторые макрокоманды могут быть недоступны. В частности, при работе с веб-базой данных недоступны некоторые макрокоманды, несовместимые с функцией публикации в службах Access.

**2. Изолированные макросы**

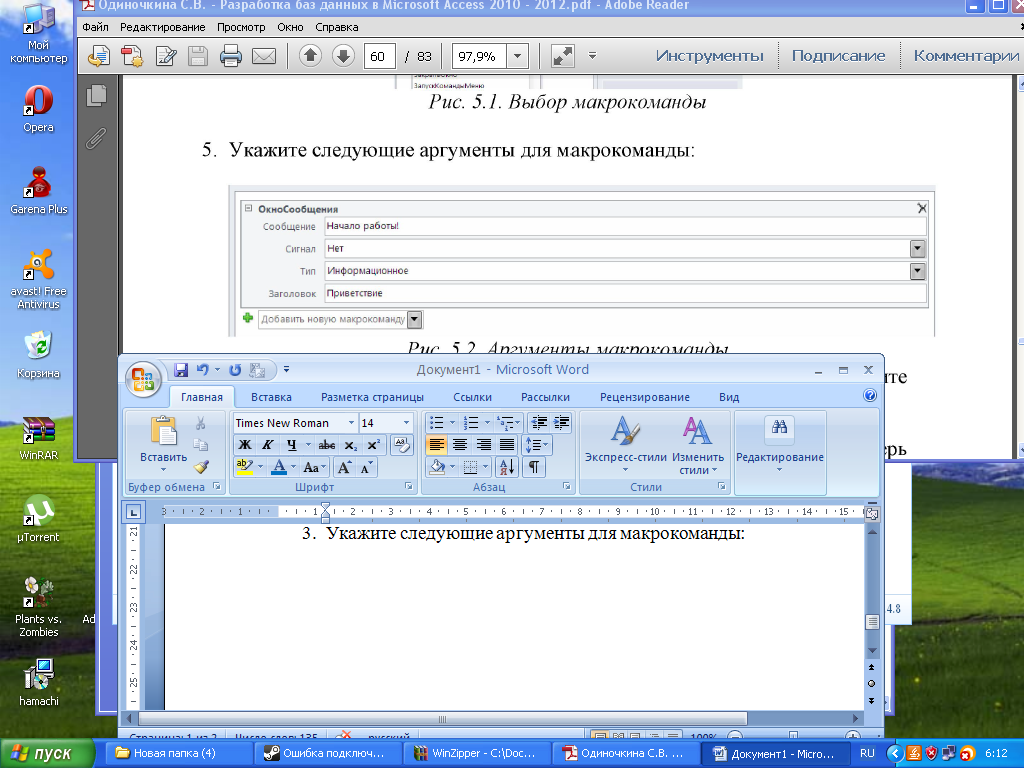
Изолированные макросы удобно использовать повторно в разных частях приложения. Вызывая тот или иной макрос из других макросов, можно избежать дублирования программного кода в нескольких местах.

**Создание изолированного макроса:**

1. На вкладке Создание в группе Макросы и код щелкните Макрос. Откроется Конструктор макросов
2. Для добавления макрокоманды в макрос в конструкторе макросов выберите макрос ОкноСообщения из выпадающего меню Добавить новую макрокоманду или перетащите макрокоманду из каталога макрокоманд из списка справа. Для поиск нужных макрокоманд рекомендуется использовать окно поиска в каталоге макрокоманд.



1. Укажите следующие аргументы для макрокоманды:



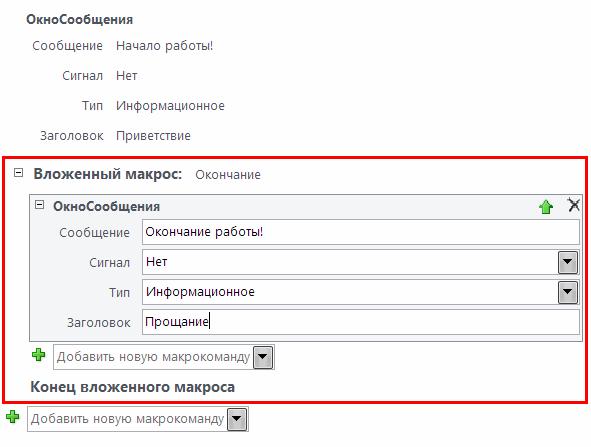
1. Выполните макрос с помощью Кнопки выполнить. Подтвердите сохранение макроса, присвоив ему имя «Сообщение».

**3.Вложенные макросы**

Вложенные макросы разрешают использование в макросе именованных групп макрокоманд, которые могут быть вызваны только их непосредственным запуском.

**Создание вложенных макросов**

1. Откройте макрос «***Сообщения***» в **режиме конструктора** **(Design)**.
2. Добавьте **Вложенный макрос** из каталога макрокоманд.
3. Присвойте имя вложенному макросу «Окончание»
4. Добавьте в окно вложенного макроса макрокоманду **ОкноСообщения** со следующими аргументами:



*Рис. Вложенный макрос*

1. Сохраните изменения и закройте окно макроса.
2. Выполните макрос из окна базы данных. Будет выполнен только первый макрос из созданных.

**Связь группы макросов с формой (работа с формой «*Товары*»).**

1. Откройте форму «***Товары***» в **режиме конструктора** **(Design)**.
2. Откройте **Окно свойств формы** **(Property Sheet)** (**F4** или кнопка **Страница свойств (Property Sheet)** в группе **Сервис (Tools)**)
3. Выберите вкладку **Свойств формы** - **События** **(Events**).
4. Для свойства **Открытие** **(On open)** – выберите из списка макрос «***Сообщения***».
5. Для свойства **Закрытие** **(on Close)** – выберите макрос «***Сообщения.Окончание***».
6. Сохраните и закройте форму . Затем откройте форму «***Товары***» (выполнится основной макрос «***Сообщения***», а потом закройте (выполнится вложенный макрос «***Сообщения.Окончание***»).

**4. Создание внедренного макроса**

* описанной ниже процедуре создается макрос, который внедряется в свойство события объекта. Такой макрос не отображается в области навигации, однако его можно вызывать из событий, таких как **Загрузка** или **Нажатие кнопки**.

Поскольку макрос становится частью объекта формы или отчета, внедренные макросы рекомендуется создавать для автоматизации задач, которые специфичны для определенной формы или отчета.

* 1. В области навигации щелкните правой кнопкой мыши форму или отчет, которые будут содержать макрос, и выберите пункт **Режим макета**.
  2. Если окно свойств не открыто, нажмите клавишу F4.

1. Выберите элемент управления или раздел, содержащий свойства события, в который нужно встроить макрос. В верхней части окна свойств из выпадающего списка в разделе **Тип выбора** можно выбрать элемент управления или раздел, а также форму или отчет целиком.
2. В области задач "Страница свойств" откройте вкладку **Событие**. Щелкните поле свойства для события, которое должно запускать макрос. Например, если требуется запускать макрос при нажатии кнопки, щелкните поле **Нажатие кнопки**.

**Примечания:**

Если поле свойства содержит слова **[Внедренный макрос]**, это означает, что для данного события уже создан макрос. Если нужно отредактировать его, выполните остальные действия процедуры.

Если поле свойства содержит слова **[Процедура обработки событий]**, это означает, что для данного события уже создана процедура Visual Basic для приложений. Прежде чем встраивать в это событие макрос, необходимо удалить процедуру. Чтобы сделать это, удалите слова **[Процедура обработки событий]**,однако сначала следует просмотреть соответствующую процедуру и убедиться, что ее удаление не нарушит функциональность базы данных. Иногда функциональность процедуры VBA можно воссоздать с помощью внедренного макроса.

1. Нажмите кнопку **Построение** …
2. В появившемся диалоговом окне **Построитель** установите флажок **Конструктор макросов** и нажмите кнопку **ОК**.

В Access запустится конструктор макросов. Перейдите к следующему разделу, в котором описано добавление команд в макрос.

**5. Добавление команд в макрос**

Команды представляют собой отдельные действия, составляющие макрос, и имя каждой из них соответствует выполняемому ей действию (например, **НайтиЗапись** или **ЗакрытьБазуДанных**).

**Действие 1. Выбор или поиск макрокоманды**

Первым действием при добавлении команды является ее поиск в раскрывающемся списке **Добавить новую макрокоманду** или в каталоге макрокоманд.

Примечания:

* По умолчанию в раскрывающемся списке **Добавить новую макрокоманду** и каталоге макрокоманд выводятся только те команды,которыеможно выполнить в недоверенных базах данных. Чтобы увидеть полный список команд, выполните указанные ниже действия.
* На вкладке **Конструктор** в группе **Показать или скрыть** выберите пункт **Показать все действия**.
* Если каталог макрокоманд не отображается, на вкладке **Макет** в группе **Показать/скрыть** нажмите кнопку **Каталог макрокоманд**.
* Чтобы найти макрокоманду, воспользуйтесь одним из указанных ниже способов.
* Щелкните стрелку в раскрывающемся списке **Добавить новую макрокоманду** и прокрутите список до нужной команды.В верхней части списка выводятся элементы программного потока, а затем в алфавитном порядке — макрокоманды.
* Перейдите к нужной команде в области каталога макрокоманд. Команды группируются по категориям. Чтобы просмотреть команды в той или иной категории, разверните ее. При выборе команды в нижней части каталога макрокоманд появляется ее краткое описание.
* Найдите нужную команду в каталоге макрокоманд, введя запрос в поле поиска в верхней части соответствующей области. По мере ввода список действий фильтруется, при этом отображаются все макросы, содержащие введенный текст. В Access поиск текста выполняется как по именам макросов, так и по их описаниям.

**Действие 2. Добавление макрокоманды в макрос**

Найдя нужную макрокоманду, добавьте ее в макрос одним из указанных ниже способов.

Выберите команду в списке **Добавить новую макрокоманду** или просто начните вводить ее имя в поле. Access добавит команду в то место, где отображается список **Добавить новую макрокоманду**.

Перетащите команду из каталога макрокоманд на область макроса. При этом появится полоса вставки, указывающая, в какое именно место будет вставлена команда, как только будет отпущена кнопка мыши.

Дважды щелкните команду в каталоге макрокоманд.

Если в области макроса выделена команда, Access добавит новую макрокоманду под выделенной.

Если в области макроса выделен блок **Группа**, **Если**, **Иначе если**, **Иначе** или **Вложенный макрос**, Accessдобавит новую макрокоманду всоответствующий блок.

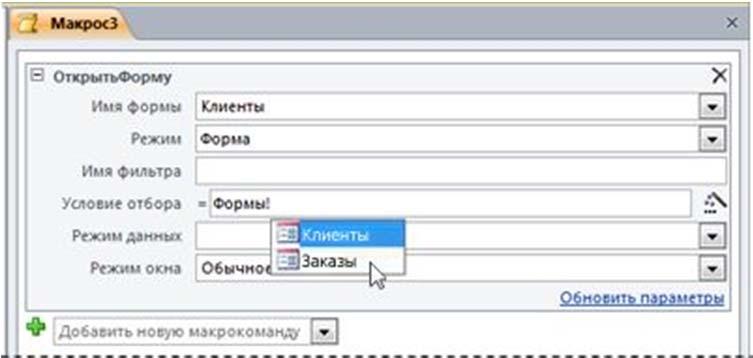
Если в области макроса не выделена ни одна команда или блок, Access добавит новую макрокоманду в конец макроса.

**Примечания:**

* Все ранее созданные макросы выводятся в узле **В этой базе данных** в каталоге макрокоманд.
* При перетаскивании изолированного макроса (указанного в разделе **Макросы**) в область макроса создается команда **ЗапускМакроса**, кото-рая запускает макрос, в который была перетащена команда. После этого из раскрывающегося списка можно запускать вложенные макросы (если они есть).
* Чтобы просто скопировать команды из изолированного макроса в текущий (вместо создания команды **ЗапускМакроса**), щелкните его в ката-логе макрокоманд и выберите команду **Добавить копию макроса**.
* При перетаскивании внедренного макроса (указанного в списке объекта формы или отчета) в область макроса действия из него копируются в текущий макрос.
* Можно создать макрокоманду, перетащив объект базы данных из области навигации в область макроса. При перетаскивании таблицы, запроса, формы, отчета или модуля в область макроса приложение Access добавляет макрокоманду, открывающую таблицу, запрос, форму или отчет. При перетаскивании макроса добавляется макрокоманда, запускающая макрос.

**Действие 3. Заполнение аргументов**

Большинству макросов необходим как минимум один аргумент. Чтобы просмотреть описания аргументов, выберите макрокоманду и наведите указатель на нужный аргумент. Для многих аргументов значение можно выбрать в раскрывающемся списке. Если аргумент требует ввода выражения, функция IntelliSense поможет ввести его, предлагая по мере ввода допустимые значения (см. рисунок ниже).



Увидев нужное значение, добавьте его в выражение. Для этого дважды щелкните значение или выделите его с помощью клавиш со стрелками, а затем нажмите клавишу TAB или ENTER.

**6. Перемещение макрокоманды**

Макрокоманды выполняются в порядке сверху вниз по тексту макроса. Чтобы переместить команду в макросе вверх или вниз, воспользуйтесь одним из указанных ниже способов.

Перетащите команду в нужное место вверх или вниз.

Выделите команду и нажмите клавиши CTRL + СТРЕЛКА ВВЕРХ или CTRL + СТРЕЛКА ВНИЗ.

Выделите команду, а затем щелкните стрелку **Вверх** или **Вниз** в правой

части области макроса.

**7.Удаление макрокоманды**

Чтобы удалить макрокоманду, выполните указанные ниже действия.

Выделите команду и нажмите клавишу DELETE. Кроме того, можно нажать кнопку **Удалить** (X) в правой части области макроса.

**Примечания:**

При удалении блока макрокоманд (например, блока **Если** или **Группа**) удаляются все действия в этом блоке.

Команды **Вверх**, **Вниз** и **Удалить** также доступны в контекстном меню макрокоманды.

**8.Запуск макроса**

Запустить макрос можно любым из указанных ниже способов.

Дважды щелкнуть макрос в области навигации.

Вызвать макрос с помощью макрокоманды **ЗапускМакроса** или **При**

**Ошибке**.

Указать имя макроса в свойстве "Событие" любого объекта. В результате макрос будет выполнен при возникновении этого события.

**9.Обмен данными**

MS Access позволяет осуществлять обмен данными с другими СУБД и с другими базами данных MS Access, с программами работы с электронными таблицами (Excel и Lotus), а также импортировать и экспортировать данные из текстовых файлов. Кроме того, с помощью этих средств можно копировать объекты из одной базы данных MS Access в другую.

**10.Экспорт данных**

Экспорт данных позволяет использовать информацию, сохраненную в Access-базе данных при работе с другой программой:

1. В области навигации выберите из списка таблицу, которую предполагается экспортировать.
2. Откройте контекстное меню и выберите команду Экспорт.
3. Выберите приложение в которое будет экспортирована таблица.
4. Нажмите кнопку **Экспорт.**
5. В открывшемся окне диалога укажите необходимые опции.
6. Нажмите **ОК.** Таблица будет сохранена в новом файле указанного типа.

**11. Импорт данных**

MS Access может считывать данные, представленные в другом формате,

* сохранять их в новой таблице данных:
  1. Перейдите в область навигации и вызовите контекстное меню.
  2. Выберите команду **Импорт** и приложение из которого будут импортироваться данные.
  3. В открывшемся окне диалога укажите необходимые опции.

**ЗАДАНИЕ:**

1. Откройте ранее созданную БД **Продажи**.
2. Создайте изолированный макрос по п.2
3. Создайте вложенный макрос по п.3
4. Создайте еще одну БД.
5. Используя импорт данных, поместите в нее таблицы **Товары** **1** и **Склады**,форму **Товары 1**,созданную в результате выполнения предыдущей практической работы.
6. Создайте макрос, позволяющий при каждом новом открытии этой БД открывать форму **Товары 1**.
7. Откройте БД **Продажи.**
8. Используя экспорт данных, поместите таблицы **Фирмы** и **Продажи** в созданную в этой практической работе БД и перейдите в нее.
9. Создайте запрос на отображение данных о названиях товаров, их марке, ценах и данных о складах, где хранятся эти товары.
10. На основе этого запроса создайте простую форму.

**Контрольные вопросы:**

1. Макрос – это …
2. С помощью чего создаются макросы?
3. Какие макросы называются макросами пользовательского интерфейса?
4. Какие действия можно автоматизировать с помощью макросов пользовательского интерфейса?
5. Какие макросы называются изолированными?
6. Какие макросы называются внедренными?
7. В чем разница между внедренными и изолированными макросами?
8. Для чего используют вложенные макросы?
9. Какие действия нужно выполнить для создания внедренного макроса?
10. Из каких действий состоит добавление команд в макрос?
11. С чем MS Access позволяет осуществлять обмен данными?
12. Какие действия нужно выполнить для экспорта данных?
13. Какие действия нужно выполнить для импорта данных?
14. Литература для самостоятельной работы:

[Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Федорова. – М.: Издательский центр «Академия», 2017. – 224 с.