



Белгород 2022 г.

## Содержание

	Стр.
1. Пояснительная записка	3
1.1. Краткая характеристика дисциплины, ее цели и задачи. Место лабораторных работ в курсе дисциплины	3
1.2. Организация и порядок проведения лабораторных работ	3
1.3. Общие указания по выполнению лабораторных работ	3
1.4. Критерии оценки результатов выполнения лабораторных работ	4
2. Тематическое планирование лабораторных работ	6
3. Содержание лабораторных работ	7
Лабораторная работа № 1. Проектирование структуры базы данных.	7
Лабораторная работа № 2. Открытие, редактирование и пополнение табличного файла. Модификация структуры табличного файла.	14
Лабораторная работа № 3. Индексирование и сортировка таблиц	20
Лабораторная работа № 4. Редактирование проекта и макета таблиц. Поиск и сортировки данных в СУБД ACCESS. Принципы создания запросов выборки	26
Лабораторная работа № 5. Установление взаимосвязей между таблицами	31
Лабораторная работа № 6. Создание и нормализация таблиц в базе данных	34
Лабораторная работа № 7. Создание форм	38
Лабораторная работа № 8. Создание и печать отчетов	41
Лабораторная работа № 9-10. Создание макросов. Основы программирования СУБД ACCESS	49
Лабораторная работа № 11. Создание базы данных и ее объектов средствами языка SQL	56
Лабораторная работа № 12-13. Выполнение простых запросов к БД в SQL	59
Лабораторная работа № 14. Выборка данных из БД. Модификация содержимого БД.	64
Лабораторная работа № 15. Применение встроенных механизмов защиты информации в системах управления базами данных.	69
4. Информационное обеспечение обучения	74

## 1. Пояснительная записка

### 1.1. Краткая характеристика дисциплины, ее цели и задачи. Место лабораторных работ в курсе дисциплины

Дисциплина ОП.08 Основы проектирования баз данных является частью рабочей основной образовательной программы в соответствии с ФГОС по специальности СПО 09.02.07 Информационные системы и программирование, квалификация - Разработчик веб и мультимедийных приложений

Дисциплина изучается в IV семестре. В целом рабочей программой предусмотрено 30 часов на выполнение лабораторных работ, что составляет 47 % от обязательной аудиторной нагрузки, которая составляет 66 часов, при этом максимальная нагрузка составляет 68 часов, из них 2 часа приходится на самостоятельную работу обучающихся.

Цель настоящих методических рекомендаций: оказание помощи обучающимся в выполнении лабораторных работ по дисциплине ОП.08 Основы проектирования баз данных, качественное выполнение которых поможет обучающимся освоить обязательный минимум содержания дисциплины и подготовиться к промежуточной аттестации в форме дифференцированного зачета.

### 1.2. Организация и порядок проведения лабораторных работ

Лабораторные работы проводятся после изучения теоретического материала. Введение лабораторных работ в учебный процесс служит связующим звеном между теорией и практикой. Они необходимы для закрепления теоретических знаний, а также для получения практических навыков и умений. При проведении лабораторных работ задания, выполняются студентом самостоятельно, с применением знаний и умений, усвоенных на предыдущих занятиях, а также с использованием необходимых пояснений, полученных от преподавателя. Обучающиеся должны иметь методические рекомендации по выполнению лабораторных работ, конспекты лекций, измерительные и чертежные инструменты, средство для вычислений.

### 1.3. Общие указания по выполнению лабораторных работ

Курс лабораторных работ по дисциплине ОП.08 Основы проектирования баз данных предусматривает проведение 14 работ, посвященных изучению:

- Основы теории баз данных.
- Модели данных.
- Особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании.
- Основы реляционной алгебры.
- Принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных.
- Средства проектирования структур баз данных.
- Язык запросов SQL.

При подготовке к проведению лабораторной работы необходимо:

- ознакомиться с лабораторным оборудованием;
- ознакомиться с порядком выполнения работы.

После выполнения лабораторной работы обучающийся к следующему занятию оформляет отчет, который должен содержать:

- название лабораторной работы, ее цель;
- краткие, общие сведения об изучаемом лабораторном оборудовании;
- необходимый графический материал, указанный преподавателем при выполнении лабораторной работы (принципиальная схема лабораторной установки, графики);
- данные, полученные непосредственно из проводимых опытов;
- результаты обработки данных опытов с необходимыми пояснениями;

- графический материал, отображающий полученные в ходе опытов значения измеряемых величин;
- оценку результатов испытаний.

При работе в лаборатории необходимо руководствоваться инструкциями по технике безопасности, учитывающими все специфические особенности лаборатории, такие как наличие высокого напряжения, легкодоступных для прикосновения токоведущих частей электрооборудования.

В лаборатории нельзя находиться в отсутствие преподавателя или лица, ответственного за технику безопасности.

При нахождении в лаборатории следует находиться в рабочей зоне, указанной преподавателем. С самого начала необходимо убедиться в том, что испытательный стенд находится в полностью обесточенном (отключенном) состоянии.

Перед выполнением лабораторной работы необходимо получить вводные инструкции преподавателя и внимательно ознакомиться с описанием лабораторного стенда и оборудованием.

**Внимание! Включать лабораторные установки и выполнять какие-либо действия с приборами допускается ТОЛЬКО с разрешения преподавателя!**

При обнаружении признаков неисправности, таких как: появление искрения, дыма, специфического запаха, аномальных показаний измерительных приборов, следует немедленно отключить все источники электроэнергии и сообщить о случившемся преподавателю.

При возникновении реальной опасности травматизма для одного или нескольких присутствующих, участники испытания должны произвести срочное отключение лаборатории от всех источников электроэнергии выключением вводного автомата. Лаборатории должны иметь средства пожаротушения и оказания первой медицинской помощи. На первом занятии изучаются правила техники безопасности и проводится вводный инструктаж с последующей проверкой его усвоения, о чем свидетельствует запись в журнале по технике безопасности кабинета/лаборатории, подписываемый преподавателем, проводившем инструктаж, и всеми обучающимися.

#### **1.4. Критерии оценки результатов выполнения лабораторных работ**

Критериями оценки результатов работы обучающихся являются:

- уровень усвоения обучающимся учебного материала;
- умение обучающегося использовать теоретические знания при выполнении практических задач;
- сформированность общеучебных и профессиональных компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 11.1. Осуществлять сбор, обработку и анализ информации для проектирования баз данных.

ПК 11.2. Проектировать базу данных на основе анализа предметной области.

ПК 11.3. Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области.

ПК 11.4. Реализовывать базу данных в конкретной системе управления базами данных.

ПК 11.5. Администрировать базы данных.

ПК 11.6. Защищать информацию в базе данных с использованием технологии защиты информации.

- обоснованность и четкость изложения материала;
- уровень оформления работы.
- анализ результатов.

#### Критерии оценивания лабораторной работы

Оценка	Критерии оценивания
5	Работа выполнена в полном объеме с соблюдением необходимой последовательности проведения, содержит результаты и выводы, все записи, таблицы, рисунки, чертежи, графики выполнены аккуратно. Обучающийся владеет теоретическим материалом, формулирует собственные, самостоятельные, обоснованные, представляет полные и развернутые ответы на дополнительные вопросы.
4	Работа выполнена в полном объеме с соблюдением необходимой последовательности проведения, содержит результаты и выводы, все записи, таблицы, рисунки, чертежи, графики выполнены аккуратно. Обучающийся владеет теоретическим материалом, допуская незначительные ошибки на дополнительные вопросы.
3	Работа выполнена в полном объеме, содержит результаты и выводы, все записи, таблицы, рисунки, чертежи, графики выполнены аккуратно. Обучающийся владеет теоретическим материалом на минимально допустимом уровне, допуская ошибки на дополнительные вопросы.
2	Работа выполнена не полностью. Студент практически не владеет теоретическим материалом, допускает ошибки при ответе на дополнительные вопросы.

## 2. Тематическое планирование лабораторных работ

	Наименование тем	Вид и название работы студента	Количество часов на выполнение работы
<b>Тема 4</b>	<b>Проектирование структур баз данных</b>		<b>20</b>
		Лабораторная работа №1. Проектирование структуры базы данных.	2
		Лабораторная работа №2. Открытие, редактирование и пополнение табличного файла. Модификация структуры табличного файла.	2
		Лабораторная работа №3. Индексирование и сортировка таблиц	2
		Лабораторная работа №4. Редактирование проекта и макета таблиц. Поиск и сортировки данных в СУБД ACCESS. Принципы создания запросов выборки	2
		Лабораторная работа №5. Установление взаимосвязей между таблицами	2
		Лабораторная работа №6. Создание и нормализация таблиц в базе данных	2
		Лабораторная работа №7. Создание форм	2
		Лабораторная работа №8. Создание и печать отчетов	2
		Лабораторная работа №9-10. Создание макросов. Основы программирования СУБД ACCESS	4
<b>Тема 5</b>	<b>Организация запросов SQL</b>		<b>10</b>
		Лабораторная работа №11. Создание базы данных и ее объектов средствами языка SQL	2
		Лабораторная работа №12-13. Выполнение простых запросов к БД в SQL	4
		Лабораторная работа №14. Выборка данных из БД. Модификация содержимого БД.	2

		Лабораторная работа №15. Применение встроенных механизмов защиты информации в системах управления базами данных.	2
		<b>Итого:</b>	<b>30</b>

### 3.Содержание лабораторных работ

#### Лабораторная работа №1

#### **Проектирование структуры базы данных.**

##### **Цель работы:**

- 1) изучить этапы проектирования реляционной БД;
- 2) изучить и практически освоить процесс создания новой БД средствами СУБД MS Access, включая разработку макета таблиц в режиме Конструктора и построение схемы БД.

##### **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.

##### **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.

##### **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:

<b>Таблица Клиенты</b>	<b>Таблица Автомобили</b>	<b>Таблица Заказы</b>
1. Код клиента (ключ)	1. Код модели (ключ)	1. Код заказа (ключ)
2. Фамилия	2. Модель	2. Код клиента
3. Имя	3. Мощность двигателя	3. Код Модели
4. Отчество	4. Цвет	4. Дата заказа
5. Адрес	5. Количество дверей	5. Скидка, %
6. Телефон	6. Заводская цена	6. Оплачено
	7. Издержки (транспортные, предпродажные)	
	8. Специальная модель	
	9. Дополнительно е 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-многочленные 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 в рабочей папке файл БД, присвоив имя toyota. Заполнить свойства БД.

4.2. Выбрать в окне БД вкладку Таблицы.

4.3. 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

Имя поля	Тип данных	Описание	Свойства поля
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	Мемо	Дополнительные аксессуары	Значение по умолчанию: Радио/плеер, 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 подстановок позволяет создать для указанных полей список, из которого выбирается значение.

4.4. Перейти в режим Таблицы, сохранив созданный макет таблицы под именем Автомобиля.

4.5. Добавить в таблицу Автомобиля 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/плеер, раздвижная крыша, лаковое покрытие “Металлик”	Радио/плеер, 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

4.6. 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)
<b>Код клиента</b>	Счетчик	Ключевое поле, уникальный номер клиента в БД	Индексированное поле: Да/Совпадения не допускаются Ключевое поле задается в меню Правка/Ключевое поле
<b>Фамилия</b>	Текст	Фамилия	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
<b>Имя</b>	Текст	Имя	Размер поля: 20, Индексированное поле: Да/Совпадения допускаются
<b>Отчество</b>	Текст	Отчество	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
<b>Индекс</b>	Числовой	Почтовый индекс	Размер поля: Длинное целое, Индексированное поле: Да/Совпадения

			допускаются
<b>Страна</b>	Текст	Название страны	Размер поля: 20, Индексированное поле: Да/Совпадения допускаются
<b>Населенный пункт</b>	Текст	Название населенного пункта	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
<b>Почтовый адрес</b>	Текст	Почтовый адрес	Размер поля: 50, Индексированное поле: Нет
<b>Телефон</b>	Текст	Контактный телефон	Размер поля: 20, Индексированное поле: Нет

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

4.8. 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)
<b>Код заказа</b>	Счетчик	Ключевое поле, уникальный номер заказа	Индексированное поле: Да/Совпадения не допускаются Ключевое поле задается в меню Правка/Ключевое поле
<b>Код модели</b>	Числовой, <b>*Мастер подстановок</b>	Внешний ключ, для связи с таблицей Автомобили	Размер поля: Длинное целое Индексированное поле: Да, допускаются совпадения
<b>Код клиента</b>	Числовой, <b>*Мастер подстановок</b>	Внешний ключ, для связи с таблицей Клиенты	Размер поля: Длинное целое Индексированное поле: Да, допускаются совпадения
<b>Дата заказа</b>	Дата/время	Дата формирования заказа ДД.ММ.ГГ	Формат: Краткий формат даты Индексированное поле: Да/Совпадения допускаются
<b>Скидка</b>	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).

4.9. Добавить 5 записей в таблицу Заказы.

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

5.1. Выполнить команду Схема данных из меню Сервис. В диалогом окне Добавление таблицы последовательно добавить все три таблицы. Закрывать 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.

5.4. Ответьте на контрольные вопросы, оформите отчет и сдайте его вместе с БД преподавателю.

## Лабораторная работа № 2

### Открытие, редактирование и пополнение табличного файла. Модификация структуры табличного файла.

#### Цель работы:

- добавлять данные в таблицу и редактировать имеющиеся;
- научиться разбивать данные на несколько таблиц;
- менять структуру таблиц;
- обеспечивать целостность данных.

#### Введение

В процессе ввода данных очень часто возникает необходимость защитить оператора от ошибки и разграничить доступ к важной информации. Для этого данные разных категорий разделяют по разным таблицам, кроме этого, как правило, такое разделение позволяет более компактно хранить информацию. Данные в главной таблице индексируют и, используя эти индексы, связывают с подчиненными таблицами. В данной лабораторной работе мы будем приводить, созданную ранее базу данных, к такому виду.

#### Ход работы

1. Откройте базу данных Европа
2. Проверьте содержимое таблицы и при необходимости внесите соответствующие коррективы. Для этого откройте таблицу Страны Европы в режиме таблица.

Албания	Площадь, 30 518 кв. км
Площадь, 28 748 кв. км	Столица Брюссель
Столица , Тирана	Число жителей 9 865 000
Число жителей , 3 149 000	Основная религия Христианство
Основная религия , Атеизм	(католики)
Денежная единица, Лек	Денежная единица Франк
Гос. строй Республика	Гос. строй Монархия
Андорра	Болгария
Площадь, 468 кв. км	Площадь, 110 994 кв. км
Столица Андорра-ла-Вьеха	Столица София
Число жителей 51 400	Число жителей 8 978 000
Основная религия Христианство	Основная религия Христианство
(католики)	(православные)
Денежная единица Франк	Денежная единица Лев
Гос. строй Княжество	Гос. строй Республика
Бельгия	Дания

Площадь, 43 092 кв. км  
Столица Копенгаген  
Число жителей 5 130 000  
Основная религия Христианство  
(протестанты)

Денежная единица Крона  
Гос. строй Монархия

Финляндия  
Площадь, 338 145 кв. км  
Столица Хельсинки  
Число жителей 4 952 000  
Основная религия Христианство  
(протестанты)

Денежная единица Марки  
Гос. строй Республика

Франция  
Площадь, 543 965 кв. км  
Столица Париж  
Число жителей 55 860 000  
Основная религия Христианство  
(католики)

Денежная единица Франк  
Гос. строй Республика

Греция  
Площадь, 131 957 кв. км  
Столица Афины  
Число жителей 10 055 000  
Основная религия Христианство  
(православные)

Денежная единица Драхма  
Гос. строй Республика

Ирландия  
Площадь, 70 285 кв. км  
Столица Дублин  
Число жителей 3 553 000  
Основная религия Христианство  
(католики)

Денежная единица Фунт  
Гос. строй Республика

Исландия

Площадь, 103 000 кв. км  
Столица Рейкьявик  
Число жителей 248 000  
Основная религия Христианство  
(протестанты)

Денежная единица Крона  
Гос. строй Республика

Италия  
Площадь, 301 277 кв. км  
Столица Рим  
Число жителей 57 401 000  
Основная религия Христианство  
(католики)

Денежная единица Лира  
Гос. строй Республика

Югославия  
Площадь, 255 804 кв. км  
Столица Белград  
Число жителей 23 591 000  
Основная религия Христианство  
(православные)

Денежная единица Динар  
Гос. строй Республика

Лихтенштейн  
Площадь, 160 кв. км  
Столица Вадуц  
Число жителей 27 840  
Основная религия Христианство  
(католики)

Денежная единица Франк  
Гос. строй Монархия

Люксембург  
Площадь, 2 586 кв. км  
Столица Люксембург  
Число жителей 372 000  
Основная религия Христианство  
(католики)

Денежная единица Франк  
Гос. строй Герцогство

Мальта

Площадь, 316 кв. км  
Столица Валлетта  
Число жителей 347 000  
Основная религия Христианство  
(католики)

Денежная единица Фунт  
Гос. строй Республика

Монако  
Площадь, 2 кв. км  
Столица Монако  
Число жителей 28 000  
Основная религия Христианство  
(католики)

Денежная единица Франк  
Гос. строй Княжество

Нидерланды  
Площадь, 41 863 кв. км  
Столица Амстердам  
Число жителей 14 741 000  
Основная религия Христианство  
(католики)

Денежная единица Флорин  
Гос. строй Монархия

Норвегия  
Площадь, 323 878 кв. км  
Столица Осло  
Число жителей 4 202 000  
Основная религия Христианство  
(протестанты)

Денежная единица Крона  
Гос. строй Монархия

Польша  
Площадь, 312 683 кв. км  
Столица Варшава  
Число жителей 37 864 000  
Основная религия Христианство  
(католики)

Денежная единица Злотый  
Гос. строй Республика

Португалия

Площадь, 92 389 кв. км  
Столица Лиссабон  
Число жителей 10 349 000  
Основная религия Христианство  
(католики)

Денежная единица Эскудо  
Гос. строй Республика

Румыния  
Площадь, 237 500 кв. км  
Столица Бухарест  
Число жителей 23 014 000  
Основная религия Христианство  
(православные)

Денежная единица Леи  
Гос. строй Республика

Сан-Марино  
Площадь, 61 кв. км  
Столица Сан-Марино  
Число жителей 22 830  
Основная религия Христианство  
(католики)

Денежная единица Лира  
Гос. строй Республика

Швейцария  
Площадь, 41 293 кв. км  
Столица Берн  
Число жителей 6 626 000  
Основная религия Христианство  
(католики)

Денежная единица Франк  
Гос. строй Республика

Испания  
Площадь, 504 783 кв. км  
Столица Мадрид  
Число жителей 38 996 000  
Основная религия Христианство  
(католики)

Денежная единица Песета  
Гос. строй Монархия

Великобритания

Площадь, 244 110 кв. км  
Столица Лондон  
Число жителей 57 006 000  
Основная религия Христианство  
(протестанты)  
Денежная единица Фунт  
Гос. строй Монархия

Швеция  
Площадь, 449 964 кв. км  
Столица Стокгольм  
Число жителей 8 415 000  
Основная религия Христианство  
(протестанты)  
Денежная единица Крона  
Гос. строй Монархия

Чехо-Словакия  
Площадь, 127 900 кв. км  
Столица Прага  
Число жителей 15 604 000  
Основная религия Христианство  
(католики)  
Денежная единица Крона  
Гос. строй Республика

Венгрия  
Площадь, 93 031 кв. км  
Столица Будапешт  
Число жителей 10 591 000  
Основная религия Христианство  
(католики)  
Денежная единица Форинт  
Гос. строй Республика

Германия  
Площадь, 357 042 кв. км  
Столица Бонн  
Число жителей 77 370 000  
Основная религия Христианство  
(протестанты)  
Денежная единица Марка  
Гос. строй Республика

Австрия

Площадь, 83 857 кв. км  
Столица Вена  
Число жителей 7 557 000  
Основная религия Христианство  
(католики)  
Денежная единица Шиллинг  
Гос. строй Республика

3. Настройте ширину столбцов таблицы Страны Европы по ширине данных.

4. Создайте таблицу Религия с полями:

Название поля	Тип данных
Код_религии	Счетчик
Религия	Текстовый

5. Поле Код\_религии сделайте ключевым

6. Создайте таблицу Строй с полями:

Название поля	Тип данных
Код_строя	Счетчик
Строй	Текстовый

7. Поле Код\_строя сделайте ключевым

8. Заполните эти таблицы, выбрав необходимые данные из исходной таблицы (таким образом, напротив каждой религии и каждого строя будет стоять его код)

9. Откройте таблицу Страны Европы и замените названия в полях Строй и Религия на соответствующие им коды в ранее созданных таблицах (для автоматизации попробуйте использовать команду замены из меню правка)

10. Откройте таблицу Страны Европы в режиме конструктора

11. Для полей Строй и Религия измените тип данных на числовой

12. Одновременно установив им размер поля Длинное целое (так как связываемые поля, как правило, должны иметь одинаковый тип данных)

Установка связи между двумя таблицами

1. Выполните команду Схема данных из меню Сервис

2. В диалоговом окне добавления таблиц добавьте в схему все три таблицы

3. На поле Код\_религии таблицы Религия нажмите левую клавишу мыши и удерживая ее перетащите на поле Религия таблицы Страны Европы

В появившемся диалоговом окне необходимо установить нужную связь: флажок напротив опции обеспечение целостности данных означает, что перед тем как занести данные в подчиненную таблицу, программа будет проверять их на соответствие главной. (Таблица Страны Европы является подчиненной для таблиц Религия и Строй). Флажок напротив опции каскадное обновление связанных полей означает, что изменения в главной таблице автоматически будут влиять на подчиненную. Флажок напротив опции каскадное удаление связанных полей означает, что поля удаленные в главной таблице будут удалены и в подчиненной.

4. Установите все эти флажки

5. Аналогичную операцию проделайте с таблицами Страны Европы и Строй (Если связь не устанавливается, еще раз проверьте типы данных связываемых полей)

6. Закройте схему с сохранением

7. Попробуйте в таблице Страны Европы в полях Строй и Религия поменять коды на несуществующие в главных таблицах. Получилось? Почему?

Контрольные вопросы

1. Чем поле отличается от записи?

2. Какие структуры ACCESS вы знаете?

3. Для чего служит структура "таблица"?
4. Для чего данным задаются различные типы?
5. Что можно настроить в свойствах поля таблицы?
6. Для чего данные разбивают на несколько таблиц?
7. Какое необходимое условие для создания связи с обеспечением целостности данных?
8. Каким образом при разбиении достигается компактность хранения информации?

## Индексирование и сортировка таблиц

**Цель:** Знакомство с технологией создания и использования простых и составных индексов и первичных ключей. Знакомство с использованием ключевых полей для связывания таблиц. Изучение типов связей.

### Задание 1. Создание индексов и ключей

1. Создать базу данных База№3 используя базу БАЗА\_ИСХ

База содержит таблицы:

Личности	Адреса	Предметы	Экзамены
ЛичнКод	ЛичнКод	КодПредм	КодПредмета
Фамилия	ПочтКод	Наименование	КодСтудента
Имя	Город	Семестр	Балл
Рожд	Улица	Часы	ДатаЭкз
Пол	Дом	Контроль	
Рост	Корпус		
Адрес	Квартира		
Телефон	Телефон		

Заполнить таблицы экспериментальным содержимым (5-7 записей) используя мастера подстановки

### 2. Создать первичные ключи в ранее сконструированных таблицах Личности и Адреса.

2.1. Открыть таблицу Личности в режиме Конструктора таблиц. Пользуясь контекстным меню поля ЛичнКод отметить его как ключевое поле.

2.1.1. Находясь в окне Личности: таблица, вызвать меню “Вид”-“Индексы” и просмотреть содержимое открывшегося окна.

2.2. Открыть таблицу Адреса в режиме Конструктора таблиц. Пользуясь окном Индексы (меню “Вид”-“Индексы”), создать первичный ключ для данной таблицы на основе поля ЛичнКод.

### 3. Создать составные индексы.

3.1. Открыть для таблицы Адреса, находящейся в режиме Конструктора таблиц, окно Индексы и просмотреть его содержимое.

3.2. Создать новый индекс Пол, предназначенный для ускорения процесса поиска и выполнения запросов по данному полю.

3.3. Создать новый составной индекс ГорУлиДом, включив в его состав поля Город, Улица, Дом.

3.4. Находясь в окне Индексы и переходя от строки PrimaryKey к ГорУлиДом, просмотреть расположенные ниже Свойства поля и дать объяснения увиденным свойствам индексов. Окно Индексы для таблицы Адреса будет выглядеть примерно так, как показано на рис.11.1.

3.4.1. Для получения сведений о свойствах индексов прочитать пояснения, размещенные в области Свойства индекса окна Индексы, а также раздел подсказки (F1), посвященный этому вопросу.

3.5. Создать составной индекс ФамИмяРожд, включающий поля Фамилия, Имя, Рожд.

3.6. Открыть для таблицы Личности, находящейся в режиме Конструктора таблиц, окно Индексы и просмотреть его содержимое.

3.6.1. Добавить в составной индекс ГорУлиДом поле Квартира.

3.6.2. Сделать составной индекс ГорУлиДом уникальным. Данный индекс необходим для обеспечения логической достоверности данных (предполагается, что не может быть двух студентов, проживающих в одной квартире).

3.6.3. Проверить работоспособность созданного индекса, введя два одинаковых адреса.

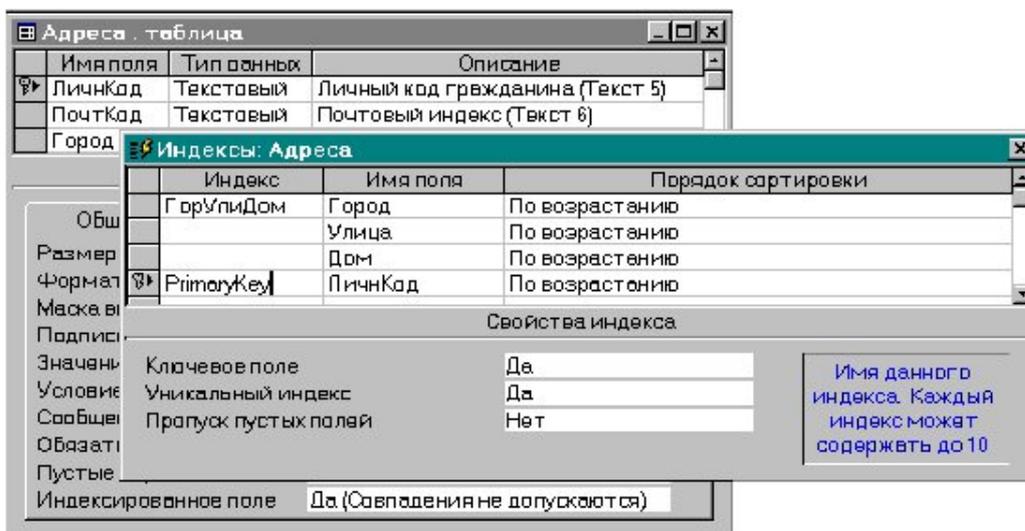


Рис.11.1

3.7. В таблице Предметы для последующего создания индексов сделать следующие коррективы:

3.7.1. Ввести новое числовое поле Семестр, имеющее минимальный размер, допускающее ввод только следующих значений [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] и заполнить его данными.

3.7.2. Создать составной индекс КодНаиСем, включающий поля КодПредм, Наименов, Семестр.

Заметьте, что КодПредм не будет первичным ключом таблицы, каким, например, в таблицах Личности и Адреса является поле ЛичнКод, так как в нескольких семестрах может читаться один и тот же предмет с одним кодом и разной продолжительностью.

3.7.3. Установить такие свойства индекса КодНаиСем, чтобы этот составной индекс стал ключом таблицы и не допускал повторяющихся значений ключа (комбинаций значений полей, составляющих этот ключ).

3.8. Перейти в режим таблицы и ввести в таблицу Предметы несколько повторяющихся значений, например: П-005 ИНФОРМАТИКА П-005 ИНФОРМАТИКА

3.9. Для таблицы Экзамены создать составной индекс КопКосДат и придать ему свойства ключа таблицы. В его состав должны войти поля КодПредм, КодСтуд, ДатаЭкз.

3.9.1. Для проверки правильности созданного ключа ввести в таблицу Экзамены данные о студенте, дважды сдававшем (пересдававшем) в один день один экзамен и получившем разные оценки. Убедитесь в невозможности ввести две различные по всему множеству полей записи с одинаковыми значениями составных ключей.

3.9.2. Ввести в таблицу Экзамены 5 корректных записей о студентах, упомянутых в таблице Личности.

4. Ввести в каждую из имеющихся таблиц по 5 новых записей с различными (допустимыми - 3 записи и недопустимыми - 2 записи) значениями уникальных составных ключей и убедиться в том, что правильно определенные ключи:

- не позволяют вводить логически неверные данные (например, о студенте, дважды сдававшем в один день один экзамен, что не допускается правилами сдачи экзаменов);

- позволяют ввести данные, о двух полных однофамильцах (тезках), родившихся в один день;

- не позволяет присвоить двум людям (хоть и имеющим одинаковые данные) один личный учетный код.

4.1. Проверить правильность работы всех таблиц базы данных, просмотреть их содержимое и сохранить все таблицы в базе данных База№3.

Примечание. Созданные в данной работе индексы, особенно составные, создавались с целью демонстрации возможности использования механизма индексации для обеспечения логической достоверности данных.

Основная цель использования индексов состоит в существенном ускорении поиска и выборки данных при использовании индексированных полей в запросах. В данном задании выборка не выполняется, поскольку на базе данных База№3 в силу

ее чрезвычайно малого объема невозможно заметить разницу в скорости выполнения выборки с использованием индексов и без их использования.

Приведенные варианты индексов служат демонстрационным целям, и в следующих заданиях в структуру ключевых полей и индексов будут внесены существенные изменения, создающие новую систему ключевых полей, предназначенную в первую очередь для организации связей между таблицами с целью их совместного использования.

## **Задание 2. Связывание таблиц**

1. Открыть базу данных База№3.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

1.2. Удалить все индексы, созданные ранее для всех таблиц базы данных База№3. Отменить все признаки первичных ключей для всех таблиц базы данных.

2. Создать новую систему первичных ключей для таблиц по следующему образцу (табл. 2)

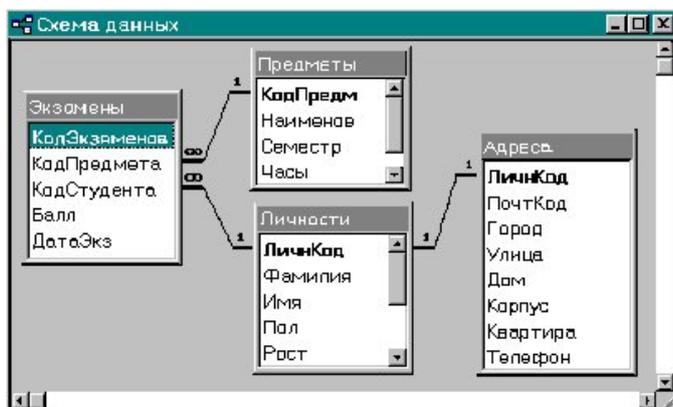
Таблица 2

Таблица	Первичный ключ
Личности	ЛичнКод
Адреса	ЛичнКод
Предметы	КодПредм
Экзамены	КодЭкзамена

2.1. В тех полях таблиц, которые становятся ключевыми, удалить повторяющиеся записи.

2.3. В таблице Экзамены для поля КодЭкзамена, которое будет уникально определять каждую запись таблицы и использоваться как первичный ключ, установить тип поля - Счетчик.

2.4. Открыть окно Схема данных и разместить таблицы так, как показано на рис.12.1 (сначала без связей).



3. Установить связи между таблицами Личности, Адреса, Предметы, Экзамены.

3.1. Внимательно прочитать все подсказки, комментарии и пояснения, касающиеся связывания таблиц и параметров объединения.

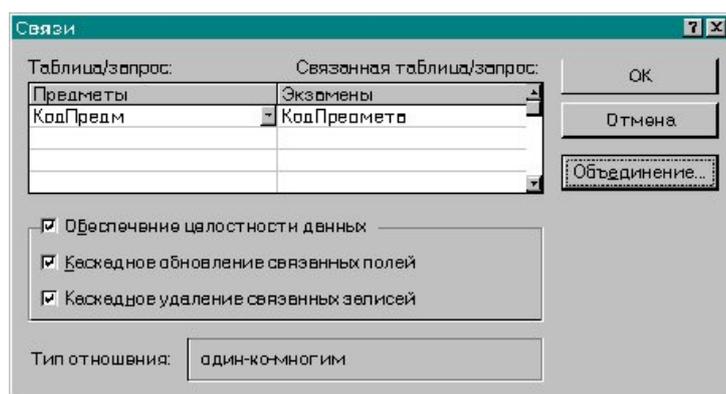
3.2. Выбрав таблицу Экзамены, с помощью мыши “перетащить” поле

КодПредмета на поле КодПредм таблицы Предметы, указав тем самым необходимость установления связи между таблицами по этим полям.

3.3. В открывшемся окне Связи определить тип и параметры связи между таблицами Экзамены и Предметы.

3.3.1. Между полями Предметы.КодПредм и Экзамены.КодПредм должна быть установлена связь типа Один-ко-Многим (рис.12.2).

3.3.2. Установить режим Обеспечения целостности данных: Каскадное обновление связанных полей и Каскадное удаление связанных записей. Для установки этих режимов необходимо активизировать соответствующие переключатели в группе Обеспечение целостности окна Связи.



3.3.3. Нажав кнопку Объединение, открыть окно Параметры объединения и выбрать такой режим объединения данных различных таблиц, при котором из связанных таблиц будут выбираться только те записи, для которых совпадают значения полей связи.

3.4. Установить связь между таблицами Личности и Экзамены через поля КодСтудента и ЛичнКод соответственно. Тип отношения между этими таблицами - Один-ко-Многим, где стороной Много выступает таблица Экзамены. Обеспечить целостность данных при обновлении и удалении записей. Установить объединение таблиц - по первому типу (когда из связанных таблиц выбираются только записи, для которых совпадают значения полей связи).

3.5. Установить связь типа Один-к-Одному между соответствующими полями таблиц Личности и Адреса с обеспечением целостности данных при обновлении и удалении записей.

3.6. Активизировать и просмотреть окно Схема данных. Его внешний вид должен соответствовать образцу, приведенному на рис.12.1.

4. Используя подтаблицы просмотреть связанные данные для таблицы Личности (меню “Вставка”  “Подтаблица”).

5. Сохранить все таблицы и базу данных База№3.

Контрольные вопросы

1. Назначение индексов в базе данных?
2. Как создать составной индекс?
3. Назначение и особенности задания связей между таблицами.

## Редактирование проекта и макета таблиц. Поиск и сортировки данных в СУБД ACCESS. Принципы создания запросов выборки

**Цель работы:** освоить приемы редактирования проекта БД и макета таблиц, поиска и сортировки данных в СУБД ACCESS Освоить принципы создания запросов выборки.

### Задачи работы

Создать запросы выборки и получить сведения о данных с использованием различных критериев.

Произвести изменения в проекте БД, макете одной из таблиц, поиск и сортировку записей в таблице.

### 1. Теоретические сведения.

#### Использование Мастера подстановок при вводе данных в таблицы

*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 и обеспечивает их достоверность.

Например, в базе данных **Студенты-Экзамены** поле *Номер зачетки* является общим для обеих таблиц. При заполнении этого поля в неключевой таблице **Экзамены** важно, чтобы его значения для каждого студента точно соответствовали значениям в ключевой таблице. Для этого значения поля *Номер зачетки* в таблице **Экзамены** лучше всего представить в виде списка. Это избавит от необходимости печатать данные при заполнении этой таблицы, предотвратит появление ошибок при наборе и сделает ввод данных более наглядным (рис.4).

Комбинированный список для поля *Номер зачетки* формируется на основе данных связанной таблицы **Студенты**. Для создания комбинированного списка нужно войти в режим конструктора таблицы **Экзамены**, перейти в колонку *Тип данных* для поля *Номер зачетки*. Из списка доступных типов полей выбрать элемент *Мастер подстановок*.

Первое диалоговое окно Мастера подстановок предлагает выбрать источник формирования списка: на основе данных таблицы\запроса или фиксированный набор значений. В данном случае нужно выбрать первый вариант.

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**, т.к. она послужит источником данных для списка номеров зачетов.

В третьем окне Мастера подстановок из списка *Доступные поля* нужно выбрать поля, значения которых фигурируют в списке. В данном случае можно выбрать поля *Фамилия* и *Имя*, которые сделают список более информативным. После этого можно нажать кнопку **Готово** для завершения процесса проектирования комбинированного списка. Сохранив проект таблицы, можно открыть таблицу **Экзамены** и просмотреть результат работы.

■ Экзамены : таблица					
Номер зачетк	Предмет	Итог	Оценка	Дата	
12522	История	Дифзачет	5	10.09.98	
12522	Боярино Андрейка	Дифзачет	3	06.09.98	
12578	Сидоров Петр	экзамен	3	09.09.98	
12579	Петров Иван	зачет	0	25.09.98	
12536	Каледин Сергей	зачет	0	24.09.98	
12578	химия	дифзачет	4	10.10.98	

**Рис.4 Использование комбинированного списка**

Другая возможность использования Мастера подстановок - это создание списков с фиксированными значениями. Он используется в тех случаях, когда набор значений в поле строго ограничен. Например, поле **Факультет** в таблице **Студенты** может иметь ограниченный набор значений (все факультеты, имеющиеся в данном ВУЗе). Никаких других значений в этом поле быть не может. Чтобы ввести единообразие в записях и избежать ошибок при вводе, это поле также можно представить в виде списка, формируемого Мастером подстановок. Процесс создания списка похож на описанный выше, но на первом шаге Мастера подстановок следует выбрать опцию **Фиксированный набор значений**. На следующем шаге нужно создать список этих значений (в данном примере записать в список все факультеты данного ВУЗа). После чего можно завершить процесс создания списка и сохранить изменения в структуре. Теперь при вводе данных в поле **Факультет** нет необходимости печатать значения, а достаточно просто выбрать из предлагаемого списка нужное.

## 2. Ход работы

Создать новую базу данных База№4.

### Задание 1.

Для создания структуры ключевой таблицы **Студенты** рекомендуется использовать режим конструктора.

**Внимание!** В бланке Свойства обязательно указать длину текстовых полей, формат числовых полей и дат. Поле Номер зачетки в таблице Студенты объявить ключевым и индексированным со значением Совпадения не допускаются.

Структура таблицы **Студенты** должна быть следующей:

Имя поля	Тип поля
Номер зачетки	Числовой
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Факультет	Текстовый
Курс	Числовой
Группа	Числовой
Дата рождения	Дата\Время

<b>Стипендия</b>	<i>Числовой</i>
------------------	-----------------

Заполнить таблицу содержимым (10-15 записей).

Структура таблицы **Экзамены** может быть следующей:

<b>Имя поля</b>	<b>Тип поля</b>
<b>Номер зачетки</b>	<i>Мастер подстановок..</i>
<b>Предмет</b>	<i>Текстовый</i>
<b>Оценка</b>	<i>Числовой</i>
<b>Дата сдачи</b>	<i>Дата\Время</i>

Обязательно определить нужные формат и длину полей в бланке **Свойства**.

Тип поля **Номер зачетки** определяется Мастером подстановок, используя для подстановки данные из таблицы **Студенты**. В качестве доступных при подстановке полей выбрать **Фамилию** и **Имя**.

Поле **Номер зачетки** в таблице **Экзамены** объявить индексированным со значением **Совпадения допускаются**.

**Внимание!** При сохранении структуры неключевой таблицы Access может предупредить об отсутствии ключевого поля и предложит создать это поле сейчас. В данном случае следует отказаться от этого.

После определения структур обеих таблиц вызвать окно **Схема данных** и добавить в схему данных обе таблицы (**Студенты** и **Экзамены**). Установить в окне схемы данных связь между таблицами по полю **Номер зачетки**. В окне **Связи** включить переключатель **Определение целостности данных**. После этого нужно указать тип связи: **Один-ко-многим** и включить опции **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**. После этого закрыть окно **Связи**.

В режиме таблицы ввести данные в таблицу **Экзамены**, используя созданный с помощью Мастера подстановок список в поле **Номер зачетки**.

Для проверки соблюдения целостности данных при работе с таблицами нужно:

\*\*\*изменить значение ключевого поля (**Номер зачетки**) для одной из записей в таблице **Студенты**. Перейти в таблицу **Экзамены** и проверить, изменилось ли в ней значение общего поля для соответствующих записей;

\*\*\*удалить одну из записей в таблице **Студенты**. Перейти в таблицу **Экзамены** и проверить, удалены ли в ней соответствующие записи.

**Внимание!** После редактирования таблицы **Студенты** нужно сначала сохранить в ней изменения, а затем переходить в неключевую таблицу.

## Задание 2.

В режиме Конструктора таблицы **Студенты** произвести в ее структуре следующие изменения:

\* добавить поля **Город, Адрес, Телефон**;

\* определить тип поля **Факультет** с помощью Мастера подстановок, взяв в качестве источника данных фиксированный набор значений (список всех факультетов ВУЗа).

\* для поля **Город** в свойстве **Значение по умолчанию** задать значение: *Уфа*.

\* для поля **Курс** ввести условие на значение:  $>0, \leq 5$  и задать соответствующее сообщение об ошибке.

Данные в поля **Город, Адрес, Телефон** вводить в режиме таблицы.

Отредактировать значения в поле **Факультет**, используя список значений, созданный Мастером подстановок.

Для нескольких записей использовать значение по умолчанию в поле **Город**.

В одной из записей попробовать внести в поле **Курс** значение, большее 5.

Вызвать окно для изменения схемы данных. Скрыть одну из таблиц, включенных в схему данных (например, **Экзамены**). Затем отобразить все прямые связи.

Изменить макет таблицы **Студенты**:

- зафиксировать столбцы **Фамилия** и **Номер зачетки**.
- поле **Город** поставить после поля **Отчество**;
- скрыть столбцы **Адрес, Телефон** и **Стипендия**;
- оставить для столбцов только вертикальную сетку;
- установить произвольно цвет фона для записей;
- изменить шрифт для записей таблицы на курсив.

Отсортировать таблицу **Студенты** по следующим признакам:

- возрастанию в поле **Фамилия**;
- убыванию в поле **Стипендия**;
- возрастанию в поле **Факультет** и убыванию в полях **Курс** и **Группа**.

Найти в таблице **Студенты** все записи, удовлетворяющие следующим условиям:

- студенты, чьи фамилии начинаются с определенной буквы;
- студенты, обучающиеся на одном курсе определенного факультета.

Найти записи для студентов определенного факультета и заменить для них название этого факультета. Например, **эконом.** на **экономический**.

### Задание 3.

Создать простой запрос - выбрать несколько произвольных полей из таблицы **Студенты**.

С помощью **Конструктора** создать запросы, удовлетворяющие условиям:

- единственное значение факультета;
- два различных факультета;
- фамилии студентов, начинающиеся с определенной буквы (использовать шаблоны);
- фамилии студентов, заканчивающиеся на “ов”;
- фамилии студентов одного факультета и одного курса;
- фамилии и имена студентов, проживающие в одном из городов или обучающиеся на одном из факультетов;
- фамилии студентов, у которых стипендия больше 400 рублей;
- фамилии студентов, занимающиеся не в 1-ой группе и стипендия которых в пределах от 200 до 500 р.

### Примечание:

В запрос должны быть включены поля **Фамилия, Имя, Отчество** и те поля, где вводятся критерии.

После того как был задан критерий для запроса, запрос нужно выполнить и сохранить под именем, подходящим по смыслу.

Для запросов с полем типа **Дата/время** добавить поле *Дата рождения* и выбрать записи, удовлетворяющие условиям:

- дата больше 1.1.80;
- дата в интервале значений и задан факультет;
- фамилии и имена студентов, родившихся в 80-х годах;
- вычислить возраст студентов;
- фамилии и имена студентов, родившихся в первой половине месяца;

Создать итоговый запрос:

- оставить в запросе поля *Факультет, Стипендия, Номер зачетки*, вычислить максимальное значение стипендии для каждого факультета и подсчитать количество студентов на каждом факультете (используя Count).

Запрос с вычисляемыми полями:

- включить в запрос вычисляемое поле, которое является результатом сцепления текстовых полей *Фамилия, Имя, Отчество*. Назвать поле *Ф. И. О. студента*.

- используя построитель выражений, подсчитать надбавку студентам, равную 15% от стипендии;

Создать запрос, в котором используются поля из двух ранее созданных и связанных таблиц, задав ему имя **Запрос для 2-х таблиц**

–убрать несколько полей таблицы **Студенты** и добавить поля *Предмет* и *Оценка* из таблицы **Успеваемость**;

–выбрать поле *Фамилия, предмет и Оценка*, вычислить минимальное значение по полю *Оценка*;

–сгруппировать по номеру зачетки и вычислить среднюю оценку для каждого студента.

### Контрольные вопросы

1. Как отсортировать записи по одному полю? по нескольким полям?
2. Как произвести поиск и замену данных в Access?
3. Как используется Мастер подстановок?
4. Что такое запрос ?
5. Что такое бланк QBE ?
6. Где записываются критерии условия выбора для запроса?
7. Перечислите основные операторы, используемые в запросе.
8. В чем различие между операторами OR и AND ?
9. Назначение итоговых запросов.
10. Назначение построителя выражений.
11. Как вычислить сумму значений заданного поля?
12. Как осуществить сортировку записей в запросе?

**Установление взаимосвязей между таблицами.**

**Цель работы:** Знакомство с основными понятиями присоединения и импорта данных. Создание учебной базы данных. Знакомство с основными приемами работы по установке связей между файлом БД и файлами электронных таблиц, текстовых файлов.

**Теоретическая часть.****Импорт и присоединение таблиц из текстовых файлов**

Access позволяет импортировать в базы данных табличные данные из обычных текстовых файлов. Эти файлы могут быть созданы либо одним из текстовых редакторов (например, приложением "Блокнот" в Windows), либо путем экспорта из другого приложения. Такая возможность особенно удобна в тех случаях, когда вам необходимы данные из приложения, которое не позволяет экспортировать данные из внутреннего представления в форматы DBF, WKS или XLS.

Существуют два основных формата текстовых файлов, в которых содержатся табличные данные:

- с полями фиксированной ширины — строки состоят из полей фиксированной ширины, пустые позиции при этом заполняются пробелами;
- с полями переменной ширины — со специальными символами — разделителями полей.

Для файлов с фиксированной шириной полей начало каждого поля определяется по его позиции относительно начала строки. Строки имеют одинаковую длину и разделяются символами новой строки (обычно это пара символов — возврат каретки и перевод строки).

Для файлов с разделителями полей существует несколько стандартных вариантов разделения полей. Они приведены в табл. 3.4.

**Таблица 1.1.** Форматы текстовых файлов, поддерживаемые Access

<b>Формат</b>	<b>Описание</b>
Записи в одну строку, поля разделены запятыми	Разделителем записей является символ новой строки. Некоторые приложения включают все значения полей в двойные кавычки, другие приложения ограничиваются заключением в кавычки лишь текстовых значений, чтобы отличить их от числовых
Записи в одну строку, поля разделены знаками табуляции	Разделителем записей является символ новой строки. Предполагается, что все значения полей текстовые, а сами поля разделены знаками табуляции. Этот формат поддерживается большей частью текстовых процессоров
Записи в одну строку, поля разделены пробелами	Access позволяет использовать пробел в качестве символа разделителя полей, однако использование такого разделителя очень неудобно для файлов, содержащих текстовые данные (в которые также может входить пробел)

Процесс импорта данных, хранящихся в текстовом файле, выполняется с помощью Мастера импорта текстов. Выберите команду **Файл, Внешние данные, Импорт** и следуйте указаниям Мастера. Если файл содержит разделители полей, то укажите символ разделителя на втором шаге Мастера импорта текстов или выберите

переключатель **фиксированная ширина полей**. Если в первой строке импортируемого файла содержатся заголовки полей, установите флажок **Первая строка содержит имена полей**

### *Ход выполнения работы.*

#### *Задание 1.*

Создайте следующие таблицы. Таблица Покупатели – текстовый файл **таблица1.txt** (в качестве разделителя можно выбрать знак табуляции), таблица Заказы – электронная таблица **Заказы**. Импортируйте созданные таблицы в базу данных **База5** создайте в ней таблицу **Продавцы**. Задайте ключевые поля (если они не заданы), установите связи, обеспечивающие целостность данных.

**Таблица 2.1 Продавцы**

Номер	Имя	Город	Комиссионные
1001	Дугинов	Липецк	0,12
1002	Ольшевский	Саратов	0,13
1004	Помякушина	Липецк	0,11
1007	Шикин	Бронницы	0,15
1003	Борескова	Новгород	0,1

**Таблица 2.2 Покупатели**

Номер	Имя	Город	Рейтинг	Номер_продавца
2001	Голубев	Липецк	100	1001
2002	Дмитриева	Архангельск	200	1003
2003	Кустов	Саратов	200	1002
2004	Кузьминов	Подольск	300	1002
2006	Кречко	Липецк	100	1001
2008	Фролов	Саратов	300	1007
2007	Каратыгин	Архангельск	100	1004

**Таблица 2.3 Заказы**

Номер	Стоимость	Дата_покупки	Номер_покупателя	Номер_продавца

3001	18,69	10/03/1996	2008	1007
3003	767,19	10/03/1996	2001	1001
3002	1900,10	10/03/1996	2007	1004
3005	5160,45	10/03/1996	2003	1002
3006	1098,16	10/03/1996	2008	1007
3009	1713,23	10/04/1996	2002	1003
3007	75,75	10/04/1996	2004	1002
3008	4723,00	10/05/1996	2006	1001
3010	1309,95	10/06/1996	2004	1002
3011	9891,88	10/06/1996	2006	1001

### ***Задание 2.***

Создайте еще три произвольных таблицы в Access, Excel и текстовом файле и присоедините их к существующей базе используя связь с таблицами.

#### **Контрольные вопросы:**

1. Особенности форматов текстовых файлов при присоединении.
2. Особенности присоединения и импорта таблиц баз данных Access.
3. Что будет отображено в таблицах после присоединения электронной таблица содержащей формулы.
4. Как происходит определение типов полей импортируемой таблицы.

Лабораторная работа №6  
Создание и нормализация таблиц базы данных

**Цель работы:**

- научиться определять структуру таблицы и устанавливать свойства ее полей в режиме конструктора таблиц;
- научиться определять простой и составной первичный ключ;
- научиться приводить таблицы к нормальной форме;
- уметь устанавливать межтабличные связи.

Необходимо создать базу данных о студентах, поступивших в ВУЗ, состоящую из двух таблиц: «Группа» и «Студент».

Дана исходная таблица 1. Привести ее к нормальной форме.

Таблица 1.

Группа	Кол-во студентов в группе	ФИО	Пол	Номер студента	Год рождения	Адрес	Проходной балл	Код специальности
11	25	Иванова	Ж	2	05.02.87		4,5	611
11	25	Сидоров	М	17	10.06.87		4,8	611
11	25	Петров	М	10	11.12.86		4,3	611
12	20	Мазуров	М	11	29.10.86		4,5	615
12	20	Бойко	Ж	1	06.01.87		5,0	615
12	20	Пашкова	Ж	9	15.12.86		4,3	615
13	20	Степанов	М	13	27.09.86		4,9	3514
13	20	Бородина	Ж	5	20.04.87		5,0	3514

Для создания структуры созданных таблиц необходимо определить основные характеристики полей таблиц, свойства полей, назначить первичный ключ. В таблицах 2 и 3 заполнить основные характеристики полей полученных таблиц. Заполненные таблицы 2 и 3 занести в отчет.

Таблица 2. Описание свойств полей таблицы «Группа»

Имя	Ключ.	Тип	Размер,	Число	Подпись поля	Условие на значение
-----	-------	-----	---------	-------	--------------	---------------------

Поля	поле	Данных	Байт	десятич. Знаков		Сообщение об ошибке
		Текстовый	5			
		Текстовый	6			
		Числовой	Байт	0		
		Числовой	Одинарное пл. точкой	с 2		

Таблица 3. Описание свойств полей таблицы «Студент»

Имя Поля	Ключ. Поле	Тип данных	Размер, Байт	Число десятич Знаков	Подпись поля	Условие на значение
						Сообщение об ошибке
		Текстовый	5			
		Текстовый	2			
		Текстовый	15			
		Подстановкой (м, ж)	1			
		Дата/Время	Краткий форм. Д.			
		Текстовый	25			
		Числовой	Одинарное с пл. точкой	2		

### I. Создание структуры базы данных

1. Запустите программу Microsoft Access.
2. В первом диалоговом окне Access установите переключатель «**Новая база данных**».
3. В следующем окне выберите папку и укажите в качестве имени файла новой базы данных **свою фамилию**. Щелкните на кнопке **Создать**.
4. В окне новой базы данных из списка объектов выберите объект **Таблицы**. Создайте структуру таблицы Группа в режиме Конструктора. При задании **Условия на значение** для полей пользуйтесь Построителем выражений.
5. Задайте ключевое поле.

6. Сохраните структуру первой таблицы под именем **Группа**.
7. Создайте структуру таблицы **Студент**. Условие задавайте Построителем выражений.
8. Задайте составной уникальный ключ, состоящий из двух полей: **НГ** (номер группы) и **НС** (номер студента в группе). Для выделения полей пользуйтесь клавишей CTRL.
9. Сохраните таблицу под именем **Студент**.

## **II. Ввод данных в таблицы**

10. В окне базы данных откройте таблицу **Группа** двойным щелчком на ее значке.
11. Введите записи.
12. Закройте таблицу **Группа**.
13. Откройте таблицу **Студент**.
14. Введите записи.
15. Закройте таблицу **Студент**.

## **III. Установка связей между таблицами**

16. Выполните команду **Сервис-Схема данных** или щелкните на кнопке **Схема данных**.
17. В окне **Схемы данных** расположите все таблицы.
18. Установите между таблицами **Группа** \* **Студент** связь по простому ключу **НГ**.
19. В открывшемся окне обратите внимание, что тип отношения установится *один-ко-многим*. Установите все флажки, обеспечивающие целостность во взаимосвязанных таблицах.
20. Установите все флажки, обеспечивающие целостность во взаимосвязанных таблицах.
21. Сохраните схему данных и закройте окно.
22. Откройте таблицы **Группа** и **Студент**. Проверьте, как влияют установленные параметры поддержания связной целостности:
  - в таблице **Группа** измените номер группы (что произошло в подчиненной таблице **Студент**? Ответ запишите в отчет);
  - в подчиненную таблицу **Студент** попытайтесь ввести запись с номером группы, не представленным в главной таблице **Группа**; Что произошло? Ответ запишите в отчет.

- в главной таблице **Группа**, удалите запись, которая вводилась последней (выделите ее и выполните команду **Правка-Удалить**); что произошло в подчиненной таблице **Студент**? Ответ запишите в отчет.

23. Закройте таблицы **Группа** и **Студент**.

#### **IV. Ввод и корректировка записей во взаимосвязанных таблицах**

24. Откройте таблицу **Группа**.

25. Создайте заново удаленную запись.

26. Используя значки «+» в строке записи, введите несколько записей в подчиненную таблицу **Студент**. (если «плюсы» не отображены, выполните команду **Вставка-Подтаблица**).

27. Покажите работу преподавателю. Закройте окно Базы данных и Access.

#### **Контрольные вопросы**

1. Что такое нормализация базы данных?
2. Какие нормальные формы вы знаете? Дайте им определения.
3. Что такое декомпозиция?

## Лабораторная работа №7

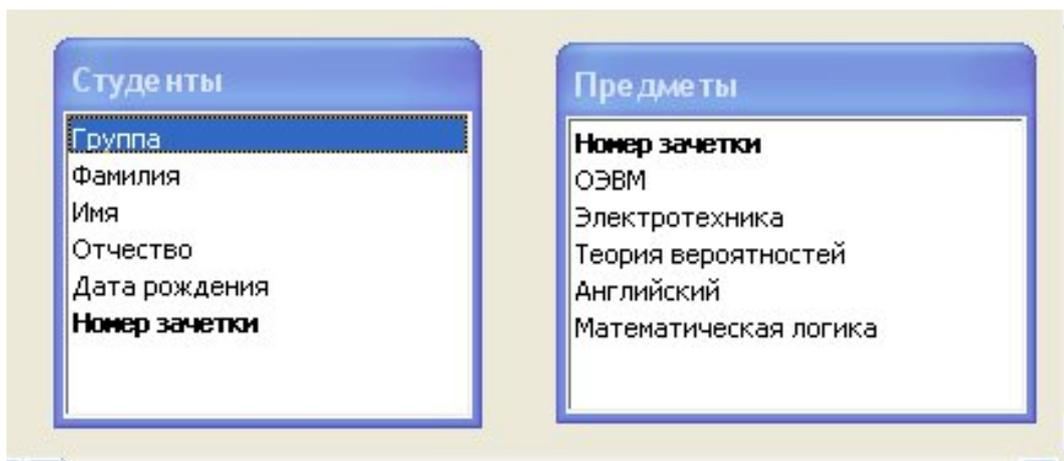
### Создание форм

**Цель работы:** в ходе выполнения лабораторной работы научиться создавать и редактировать экранные формы.

#### Ход работы:

**Задание 1.** Создайте базу данных **БАЗА10**. Разработайте структуру таблиц **Предметы** и **Студенты**. Свойства полей этих таблиц запишите в отчет (*Имя поля, Ключевое поле, Тип Данных, Размер поля, Число десятичных знаков, Подпись поля, Условие на значение, Сообщение об ошибке, Маска ввода и т.д.*).

Предусмотреть для некоторых полей маску ввода и ввод данных с помощью мастера подстановки (при необходимости создать дополнительные таблицы). Таблицы должны находиться в третьей нормальной форме. Создать схему данных. Схему данных зарисовать в отчет.



Заполнить таблицы экспериментальным содержанием (10-15 записей).

**Задание 2.** Создать на основе таблиц Microsoft Access экранную форму в одну строку, в один столбец, составную экранную форму на основе связанных таблиц. Создать простую форму с помощью конструктора форм, предусмотреть вычисляемые поля, маску ввода, ограничения на вводимые символы.

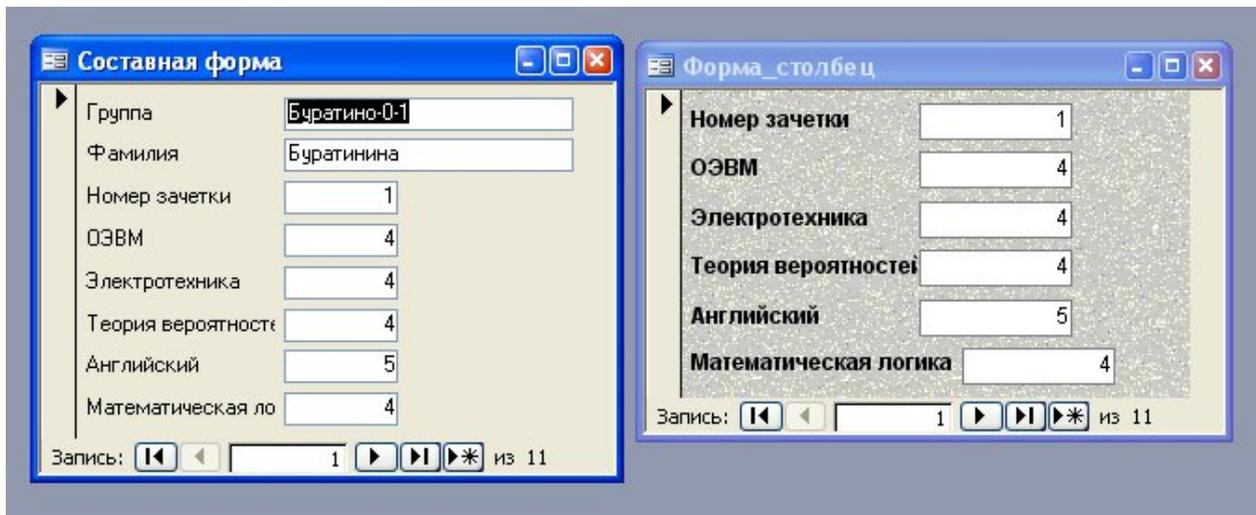


Рис 2. Пример форм, созданных при помощи мастера

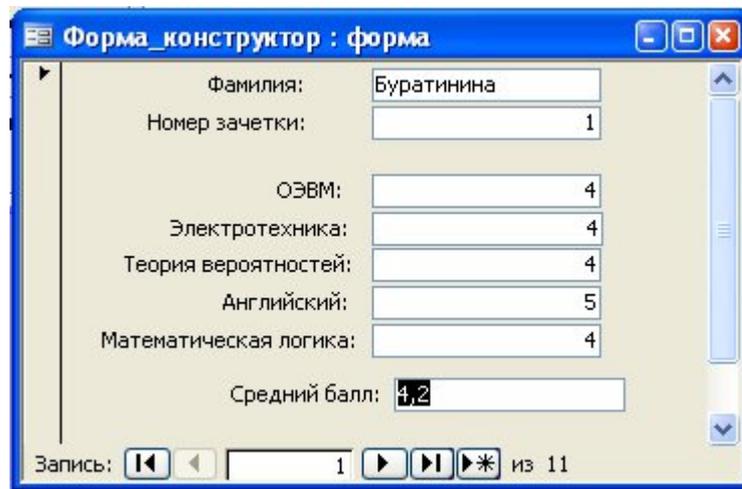


Рис 3. Пример формы, созданной в режиме конструктора

Поле «Средний балл» является вычисляемым. Выражение для вычисления среднего балла:

$$=([ОЭВМ]+[Электротехника]+[Теория вероятностей]+[Английский]+[Математическая логика])/5$$

На поля с оценками установлено ограничение на ввод цифр от 1 до 5.

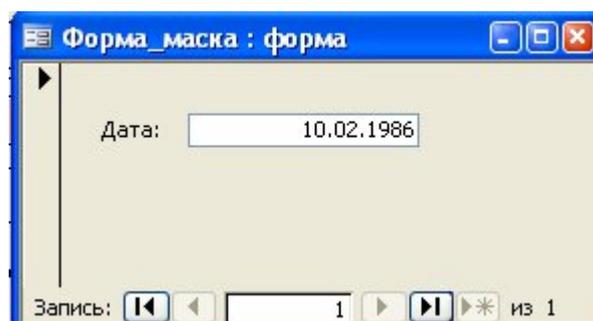


Рис 4. Привет поля с маской ввода

**Задание 3.** Создать кнопки перехода из одной формы в другую.

**Контрольные вопросы:**

1. Создание экранной формы с помощью мастера.
2. Создание экранной формы с помощью конструктора.
3. Элементы управления: назначение, создание, удаление.
4. Перечислить способы создания форм и виды форм.

Лабораторная работа №8-9  
Создание и печать отчетов.

**Цель работы:** Освоить приемы создания отчетов и графического представления данных в СУБД ACCESS. Создание отчетов разных типов и их редактирование

**Контрольные вопросы**

1. Что представляют собой отчеты в Access?
2. В чем их преимущества над другими методами вывода данных?
3. В чем отличие автоотчетов от других способов создания отчета?
4. Назовите разделы отчета.
5. В поле какого типа можно задать группировку по интервалу? По годам?
6. Для чего применяется свойство объекта **Не выводить повторы**?
7. Наличие каких объектов необходимо для внедрения подчиненного отчета?
8. Какие изменения можно внести в отчет в режиме предварительного просмотра?

**Содержание работы**

Запустить MS Access и создать базу данных **Кафедра** на основе характеристики объектов-таблиц.

**Характеристики таблицы-объекта СОТРУДНИК:**

- идентификационный код **Идент код** (10 символов - тип текстовый );
- фамилия **Фамилия** ( 20 символов - тип текстовый);
- имя **Имя** ( 15 символов - тип текстовый);
- отчество **Отчество** ( 15 символов - тип текстовый);
- пол **Пол** (1 символ - тип текстовый);
- дата рождения **Дата рожд** ( поле типа дата );
- место рождения **Место рожд** ( 15 символов - тип текстовый );
- образование **Образов** (15 символов - тип текстовый);
- должность **Должн** (15 символов - тип Мастер подстановок);
- стаж работы **Стаж работы** (длинное целое - тип числовой );
- семейное положение **Сем полож** (7 символов - тип текстовый) ;
- дата зачисления на работу **Дата зач** (поле типа дата/время);
- телефон **Тел** (8 символов - тип текстовый);
- домашний адрес **Адрес** (поле-МЕМО).

**Характеристики таблицы-объекта СОСТАВ СЕМЬИ:**

- идентификационный код **Идент код** (10 символов - тип Мастер подстановок );
- отношение **Отношение** (10 символов - тип текстовый);
- фамилия **Фамилия** (20 символов - тип текстовый);
- имя **Имя** (15 символов - тип текстовый);
- отчество **Отчество** (15 символов - тип текстовый);
- год рождения **Дата рожд** (поле типа дата/время).

**Характеристики таблицы-объекта ШТАТНОЕ РАСПИСАНИЕ:**

- № п/п **НПП** (длинное целое - тип числовой);
- название подразделения **Назв подр** (30 символов - тип текстовый);
- должность **Должн** (15 символов - тип текстовый);
- количество штатных единиц **Кол ед** (длинное целое - тип числовой);
- должностной оклад **Оклад** (длинное целое - тип числовой);
- фонд заработной платы за месяц **ФЗПМ** (длинное целое - тип числовой);
- фонд заработной платы на год **ФЗПГ** (длинное целое - тип числовой);

Таблица 1. Таблица-объект СОТРУДНИК

Идент код	Фамилия	Имя	Отчество	Пол	Дата рожд	Место рожд	Образов	Должн	Стаж работы	Сем полож	Дата зач	Телефон	Адрес
1314152347	Старченко	Светлана	Борисовна	ж	22.04.43	г.Казань	ср.технич.	статистик	36	замужем	24.09.95	65-12-13	ул.Гв.Широнинцев 21, кв.30
1545678990	Архипов	Сергей	Иванович	м	23.03.49	Харьков	высш.	директор	30	женат	10.12.88	нет	пер.Хрустальный 8
1624790203	Круговой	Геннадий	Иванович	м	22.04.45	г.Омск	высшее	зав.кафедрой	32	вдовец	01.09.90	68-14-13	ул.Светлая 14, кв.55
1748576413	Царева	Анна	Николаевна	ж	30.07.50	г.Харьков	высшее	гл.бухгалтер	29	замужем	01.01.96	47-23-15	ул.Артема 24, кв.1
1934789231	Каменева	Татьяна	Дмитриевна	ж	24.06.59	г.Курск	высшее	начальник ОК	20	замужем	30.12.90	65-67-72	ул.Героев труда 28-Б, кв.76
1955443781	Безродный	Владимир	Михайлович	м	05.09.53	г.Харьков	высшее	зам.директора	24	женат	01.09.92	32-32-14	пр.Правды 44, кв.55
2014654788	Садчиков	Аркадий	Викторович	м	10.01.57	г.Тамбов	ср.технич.	диспетчер	21	холост	15.10.98	10-12-10	пер.Короленко 2, кв.1
2055894321	Бронзов	Станислав	Иванович	м	12.11.60	г.Москва	высшее	преподаватель	15	женат	31.08.94	23-10-70	ул. Революции 6, кв.2
2178943214	Мапошенко	Юрий	Николаевич	м	21.11.64	г.Омск	высшее	специалист	13	женат	31.08.94	43-35-13	ул.Пушкинская 54, кв.2
2200987654	Коваль	Александр	Николаевна	ж	31.03.65	г.Киев	высшее	методист	10	замужем	01.10.92	47-67-33	ул.Иванова 5, кв.2



Таблица 2. Таблица-объект СОСТАВ СЕМЬИ

Идент код	Отношение	Фамилия	Имя	Отчество	Дата рожд
1314152347	отец	Старченко	Николай	Иванович	12/01/1917
314152347	мать	Старченко	Людмила	Яковлевна	25/12/1920
1545678990	сын	Архипов	Дмитрий	Сергеевич	01/09/1988
1748576413	муж	Царев	Петр	Алексеевич	14/11/1948
1934789231	муж	Каменев	Александр	Иванович	15/02/1952
1955443781	дочь	Безродная	Алла	Владимировна	24/06/1991
2014654788	мать	Садчикова	Мария	Ивановна	29/04/1930
2055894321	дочь	Бронзова	Инна	Станиславовна	15/12/1998
2178943214	сын	Мапошенко	Игорь	Юрьевич	22/06/1992
2178943214	сын	Мапошенко	Владимир	Юрьевич	23/08/1995
2233668943	дочь	Строкова	Юлия	Олеговна	28/07/1985

Таблица 3. Таблица-объект ШТАТНОЕ РАСПИСАНИЕ

Нпп	Назв подр	Должн	Кол ед	Оклад	ФЗПМ	ФЗПГ
1	Дирекция	бухгалтер	2	230	460	5520
2	Дирекция	гл.бухгалтер	1	430	430	5160
3	Дирекция	директор	1	530	530	6360

4	уч.кафедра	диспетчер	1	100	100	1200
5	уч.кафедра	доцент	1	500	500	6000
6	уч.кафедра	зав.кафедрой	1	430	430	5160
7	Дирекция	зам.директора	1	500	500	6000
8	уч.кафедра	методист	2	200	400	4800
9	Дирекция	начальник ОК	1	150	150	1800
10	уч.кафедра	преподпватель	4	350	1800	21600
11	уч.кафедра	статистик	1	100	100	1200
12	уч.кафедра	Специалист	2	150	300	3600



СПИСОК СОТРУДНИКОВ ПРЕДПРИЯТИЯ 26.02.01

Идентификационный код	Фамилия	Имя	Отчество	Телефон
-----------------------	---------	-----	----------	---------

*Группа фамилий, начинающихся на букву А*

1545678990                      Архипов Сергей Иванович нет

*Количество в группе 1*

*Группа фамилий, начинающихся на букву Б*

2314743296                      Бородулин    Андрей    Васильевич 27-14-12

1955443781 Безродный    Владимир    Михайлович 32-32-14

2055894321    Бронзов    Станислав    Иванович 23-10-70

*Количество в группе 3*

*Группа фамилий, начинающихся на букву К*

1624790203                      Круговой    Геннадий    Иванович 68-14-13

2200987654    Коваль    Александра    Николаевна 47-67-33

1934789231    Каменева    Татьяна    Дмитриевна 65-67-72

*Количество в группе 3*

*Группа фамилий, начинающихся на букву М*

2178943214                      Мапошенко Юрий Николаевич 43-35-13

*Количество в группе 1*

*Группа фамилий, начинающихся на букву С*

2014654788                      Садчиков    Аркадий    Викторович 10-12-10

2233668943    Строков    Олег    Викторович 69-05-03

1314152347    Старченко    Светлана    Борисовна 65-12-13

### Задание 1. Создать простой отчет, содержащий:

- список сотрудников предприятия с полями: **Идент код, Фамилия, Имя, Отчество, Телефон;**
- Сгруппировать данные по 1-ой букве фамилии;
- Вычислить количество фамилий в каждой группе.

Вид отчета представлен на Рисунке 1.

1. Открыть базу данных, для которой создается отчет ;

2. Начать создание отчета в режиме **Конструктор**:

3. Установить размеры отчета

- Переместить правую границу окна создания отчета с помощью указателя мыши так, чтобы на верхней линейке было видно число 19 (размер отчета 18 см);

- Выполнить **Файл -> Параметры страницы**:

- При выбранной вкладке **Страница** установить книжную ориентацию листа и размер А4 (210x297);

- При выбранной вкладке **Поля** установить размеры левого и правого поля по 10 мм;

- При выбранной вкладке **Столбцы** установить: количество столбцов-1; ширина

столбца-18 см; высота - 3 см; >  >;

- Переместить правую границу области данных отчета до значения 18 на верхней линейке;

4. Добавить в бланк отчета области **Заголовок отчета** и **Примечание отчета**:

- Для этого выполнить **Вид -> Заголовок/Примечание отчета**.

5. Переместить из таблицы в **Область данных** список нужных полей.

- В окне таблицы **Сотрудник** выделить в комбинации с клавишей **Ctrl** поля **Идент код, Фамилия, Имя, Отчество, Телефон** и отбуксировать их в **Область данных**. В **Области данных** появятся связанные элементы управления, т.е. элементы, связанные с полями таблицы **Сотрудник** (слева - подпись, справа - значение поля). Выполнить щелчок мышью на свободном пространстве в области данных, чтобы убрать выделение вставленных элементов управления;

- Можно просмотреть содержимое отчета на данном этапе, выбрав **Файл -> Предварительный просмотр**. В дальнейшем можно использовать эту команду для просмотра содержимого отчета после внесения каких-либо изменений;

- Переместить заголовки столбцов в область **Верхний колонтитул** для этого:

- Выделить подписи элементов управления (слева) в **Области данных**, для чего нажать клавишу **Shift** и выполнить щелчок на каждой подписи (или обвести их слева направо с нажатой левой кнопкой мыши). Выполнить команду **Вырезать**;

- Активизировать **Верхний колонтитул** щелчком мыши по заголовку и выполнить команду **Вставить**. Подписи будут вставлены в область **Верхнего колонтитула**;

- Расставить заголовки столбцов следующим образом: подпись **Идент код** переместить в левый верхний угол области. Остальные подписи расставить так, чтобы расстояние между левыми границами подписей было равно 3см;

- Выполнить редактирование и форматирование заголовков столбцов (в соответствии с Рисунком 1). Для этого выделить все подписи в строке (поместить курсор мыши слева от строки, чтобы курсор принял форму стрелки, направленной вправо и выполнить щелчок мышью), щелкнуть правой кнопкой мыши на выделении, в появившемся окне выбрать команду **Свойства** и установить во вкладке **Макет** следующие значения: Ширина 3см, Высота 1см, Тип границы Сплошная, Размер шрифта 12см. Для редактирования подписи выделить элемент, выполнить щелчок мышью на подписи и внести нужные изменения. Замечания: Форматировать можно и отдельный элемент, выделив только его. При форматировании можно использовать пиктограммы панели форматирования на панели инструментов, или команды меню **Формат -> Выровнять**, **Формат -> Размер**, или установить соответствующие свойства для данного элемента.

- Уменьшить размер области **Верхний колонтитул** по размеру высоты заголовков столбцов, добавив приблизительно 0,5 см, переместив границу следующей области Область данных вверх;

- Разместить поля в **Области данных** в ряд под заголовками соответствующих столбцов. Уменьшить размер **Области данных** мышкой переместив нижнюю границу области вверх. Отформатировать каждое поле в соответствии с Рисунком.1;

- Просмотреть содержимое отчета на данном этапе, выбрав **Файл -> Предварительный просмотр**. Внести изменения, если есть несоответствия Рисунку.1 или ошибки.

6. Определить поля, по которым будет производиться группировка и сортировка данных:

- Выполнить щелчок по пиктограмме **Сортировка и группировка** панели инструментов или выбрать **Вид -> Сортировка и группировка**. Открывается окно **Сортировка и группировка**;

- в столбце **Поле/выражение** (левый столбец) открыть список полей и выбрать поле **Фамилия**;

- в столбце **Порядок сортировки** установить порядок сортировки (по возрастанию)

- Выполнить установку свойств в области **Свойства группы**:

- Для **Заголовков группы** и **Примечание группы** установить значение Да. При этом в окне отчета появляется дополнительная область **Заголовок группы "Фамилия"** перед **Областью данных** и **Примечание группы** после **Области данных**;

- Для группировки по первому символу установить в строке **Группировка** значение **По первым символам**;

- В строке **Интервал** установить число начальных символов, по которым хотим образовывать группы, для группировки по одному первому символу это значение равно 1;

- В строке **Не разрывать** установить значение **Полную группу**;

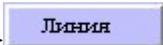
- Закрыть окно **Сортировка и группировка**.

7. Вставить в область **Заголовок группы** бланка отчета текст "*Группа фамилий, начинающихся на букву* ", а затем должна выводиться буква, по которой создавалась очередная группа:

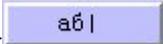
- Выделить область **Заголовок группы** в бланке отчета (щелкнуть на заголовке области);

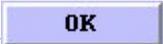
- Из окна **Список полей** перетащить поле **Фамилия** в бланк отчета в область **Заголовок группы**.. Поместить указатель с изображением руки с вытянутым указательным пальцем на маркер, расположенный в левом верхнем углу левого поля (подпись) и отбуксировать это поле в левый верхний угол области **Заголовок группы**, отступив 0,25см слева. Аналогично переместить правое поле вправо на 8см от левой границы области. В левом поле набрать текст "*Группа фамилий, начинающихся на букву*" и нажать клавишу

**Enter.** Установить параметры форматирования: курсив, размер 10. Затем выполнить действия: **Формат -> Размер -> По размеру данных**. Изменить размеры правого поля, перемещая маркер правой границы влево так, чтобы видна была одна буква Ф и немного следующая.

- В окне **Панель элементов** щелкнуть по кнопке , переместить курсор в область **Заголовок группы** под набранный текст и провести линию, подчеркнув оба поля;
- Просмотреть отчет, выбрав **Файл -> Просмотр**. Если видны 2 буквы фамилии, то уменьшить поле **Фамилия**, если где-либо не видно буквы, то поле **Фамилия** увеличить (как описано предыдущем пункте).

9. Вставить в область **Примечание группы** бланка отчета текст "*Количество в группе*", а затем должно выводиться количество фамилий, относящихся к данной группе:

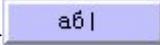
- Создать элемент управления. Для этого выполнить щелчок на **Панели элементов** по кнопке , а затем в области **Примечание группы** в месте расположения элемента. Появляется элемент, состоящий из 2-х частей. Переместить правую часть элемента вправо. В левую часть поля (подпись) ввести текст "*Количество в группе*". В правую часть элемента ввести формулу  $=Count([Фамилия])$ . Произвести форматирование.
10. Вставить рисунок в заголовок отчета:

- **Вставка -> Рисунок -> Из файла**. В открывшемся окне выбрать подходящий графический файл, выполнить щелчок по кнопке .

11. Вставить текст заголовка отчета и рядом дату создания отчета:

<u>Список по подразделениям сотрудников,</u>						26.02.01 15:15:00
<u>имеющих в составе семьи более 1 чел.:</u>						
Фамилия:	Имя:	Отчество:	Должность:	Оклад:	Состав семьи:	
<u>Название подразделения:</u>			Дирекция			
Царева	Анна	Николаевна	гл.бухгалтер	430	1	
Каменева	Татьяна	Дмитриевна	начальник ОК	150	1	
Безродный	Владимир	Михайловна	зам.директора	500	1	
Архипов	Сергей	Иванович	директор	530	1	
<u>Суммарный оклад по подразделению: Дирекция</u>				1610		
<u>Средний оклад по подразделению: Дирекция</u>				402,5		
<u>Суммарный оклад по предприятию:</u>				1610		
<u>Название подразделения:</u>			уч. кафедра			
Строков	Олег	Викторович	преподаватель	350	2	
Старченко	Светлана	Борисовна	статистик	100	2	
Садчиков	Аркадий	Викторович	диспетчер	100	1	
Мапошенко	Юрий	Николаевич	специалист	150	2	
Бронзов	Станислав	Иванович	преподаватель	350	1	
<u>Суммарный оклад по подразделению: уч. кафедра</u>				1050		

Средний оклад по подразделению:	уч. кафедра	210
Суммарный оклад по предприятию:		2660
Рисунок 2		

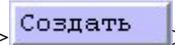
- На панели элементов выбрать кнопку , переместить курсор в область **Заголовок отчета** справа от рисунка. Отодвинуть правое поле ближе к правой границе области. В левое поле ввести текст заголовка. В правом поле с надписью **Свободный** ввести формулу *zte()*. Отформатировать поля. Вставить номер страницы в области **Нижний колонтитул**:

- Аналогично добавить элемент **Номер страницы**: в левое поле ввести текст "*Страница*", а в правое =*Page*. Сохранить отчет.

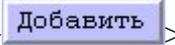
#### Задание 2. Создать сложный отчет, содержащий:

- список сотрудников по подразделениям, имеющих в составе семьи более 1 человека;
- суммарный оклад и средний по подразделениям;
- суммарный оклад по всему предприятию.
- Вид отчета представлен на Рисунке 2.
- В связи с тем, что создаваемый отчет использует информацию из всех трех таблиц базы данных, будем строить отчет на итоговом запросе **Запрос для отчета**, который создайте самостоятельно.

#### Создание запроса, используемого для создания сложного отчета

- При выбранной вкладке Запрос выполнить щелчок по кнопке  >
- Открывается окно **Новый запрос**, в котором выбрать режим создания запроса

**Конструктор**, затем - >  >;

- Открывается окно **Запрос3: запрос на выборку**, а затем активизируется окно **Добавление таблицы**, в котором выбрать из списка таблиц таблицу **Сотрудник** щелчком мыши по имени таблицы, а затем выполнить щелчок по кнопке  >, аналогично добавить таблицы **Состав семьи** и **Штатное расписание**, после чего закрыть окно **Добавление таблицы**;

- Из первой таблицы выбрать поля **Фамилия, Имя, Отчество, Должность**. При помощи мыши переместить на бланк запроса.

- Из второй таблицы выбрать поле **Идент код**, поместить его на бланк запроса.

- Из третьей таблицы выбрать поля **Назв Подр** и **Оклад** и также поместить их на бланк.

- Добавить в бланк раздел *Групповая операция* (**Вид -> Групповые операции**) или выполнив щелчок по соответствующей пиктограмме  >.

- В столбце **Идент код** выполнить щелчок мышью в строке *Групповая операция*, появится кнопка раскрытия списка. Выполнить по ней щелчок мышью и выбрать из списка функцию **Count** (количество записей с одинаковыми значениями для данного поля). Окно запроса будет иметь вид, представленный на Рисунке 3.



Рисунок 3

2. Установить размеры отчета.
3. Добавить в бланк отчета области **Заголовок отчета** и **Примечание отчета**.
4. Определить поля, по которым будем группировать и сортировать данные:
  - Установить группировку по полю **Назв подр** (название подразделения), порядок сортировки по возрастанию.
5. Добавить области **Заголовок группы** и **Примечание группы**;
6. Выбрать из окна запроса **Запрос для отчета** в **Область данных** список всех полей, кроме **Назв подр** и перенести заголовки колонок таблицы в **Верхний колонтитул**. Выполнить размещение и форматирование всех полей в областях **Верхний колонтитул** и **Область данных** в соответствии с Рисунком 3.
7. Заполнить область **Заголовок группы**:
8. Выбрать из окна запроса **Запрос для отчета** в **Заголовок группы** поле **Назв подр** и изменить содержимое поля подпись (левое поле) на **Название подразделения**. Выполнить форматирование полей в соответствии с Рисунком 2.
9. Заполнить область **Примечание группы**:
  - Переместить из списка полей запроса **Запрос для отчета** в **Примечание группы** поле **Назв подр** 2 раза, расположив связанные поля друг под другом. Изменить название полей подпись (левое поле) на **Суммарный оклад по подразделению** для первой строки и **Средний оклад по подразделению** для второй строки;
  - Рядом с 1-ой и 2-ой строками создать по одному не заполненному полю, щелкнув сначала на Панели элементов по кнопке , а затем в области **Примечание группы** в соответствующей строке. Убрать левые поля (подписи), выделив их и затем, щелкнув на кнопке , на панели элементов. В верхней строке в поле с надписью **Свободный** набрать формулу  $=Sum([Оклад])$  (для вычисления суммарного оклада по подразделению), в нижней -  $=Avg([Оклад])$  (для вычисления среднего оклада по подразделению). Разместить поля с формулами под заголовками соответствующих столбцов;
  - Аналогично создать строку для вывода в отчет **Суммарного оклада по предприятию** в разделе **Примечание отчета**, создав сначала связанное поле и введя соответствующую подпись и формулу. Для поля с формулой в качестве значения свойства **Сумма с накоплением** установить значение, отличное **Отсутствует**.

10. Вставить в раздел **Заголовок отчета** текст заголовка отчета, а рядом дату и время создания отчета, вставить номер страницы. Для вывода даты и времени использовать в формуле функцию *Now()*.

11. Выполнить форматирование полей в соответствии с Рисунком 2. Просмотреть отчет. Сохранить отчет.

## **Создание макросов. Основы программирования СУБД ACCESS.**

**Цель:** Знакомство с созданием простых макросов и их использованием для автоматизации работы базы данных. Использование групп макросов и макросов с условиями для автоматизации работы базы данных. Научиться создавать автоформы и автоотчеты. Знакомство с созданием кода VBA

### **Теоретические сведения:**

#### **Что такое Visual Basic for Applications.**

Visual Basic for Applications представляет собой полужункциональный язык программирования, являющийся неотъемлемой составной частью Access. Этот язык используется для разработки приложений, предназначенных для манипулирования БД и для настройки пользовательского интерфейса. VBA - это структурированный язык программирования высокого уровня. В нем, как и в других языках, есть операторы проверки условий, циклического выполнения повторяющихся операций, а также обмена данными с памятью и дисками. В языке VBA реализованы общие принципы объектно-ориентированного программирования. Это означает, что пользовательская среда, управляемая приложением, не подвергается изменениям путем выполнения последовательности процедур и операторов, но реагирует на события, связанные с различными объектами: полями ввода, кнопками, разделами форм и отчетов. В языке VBA программный код привязан непосредственно к объектам и срабатывает тогда, когда случается определенное событие. Все программирование в Windows основано именно на отклике на то или иное событие в системе.

Макросы – простой способ автоматизации вашего приложения.

В **MICROSOFT ACCESS** имеется свыше 40 макрокоманд, которые вы можете включать в макросы. Макрокоманды выполняют такие действия, как открытие таблиц и форм, выполнение запросов, запуск других макросов, выбор опций из меню, изменение размеров открытых окон и т.п.

Для выполнения, создания или редактирования макроса щелкните по корешку **Макрос**, находясь в окне вашей базы данных. В появившемся меню снизу вы увидите список уже созданных макросов, сверху – кнопки: **Создать** (для создания нового макроса), **Выполнить** (для выполнения действия в макросе, выбранном в окне базы данных, причем при запуске макроса выполняется его первая макрокоманда), **Конструктор** (для создания и изменения макросов).

## Создание простого макроса

В столбце **Макрокоманда** вы можете задать одну из предоставляемых макрокоманд, в столбце **Примечание** можете писать комментарии. При выборе макрокоманды в правой нижней части экрана появятся аргументы макрокоманды, которые вы можете установить нужным вам образом. Перед запуском макроса его следует сохранить. Для проверки работы макроса рекомендуется выполнить команды макроса в пошаговом режиме, для этого в окне создания или редактирования макроса щелкните по кнопке **По шагам** на панели инструментов, либо выполните команду **Макрос>По шагам**.

### Задание 1. Создание автоформ, автоотчетов, макросов.

1. Создать базу данных **База№6**, которая содержит следующие таблицы:

1.1. **Студенты** (поля: КодСтуд, ФамСтуд, ИмяСтуд, ПолСтуд, РождСтуд).

1.2. **Родители** (поля: КодСтуд, ФамРод, ИмяРод, ОтчРод, ПолРод, ТелРод).

1.3. **Оценки** (поля: КодСтуд, БаллХим, БаллИнф, БаллМат, Семестр).

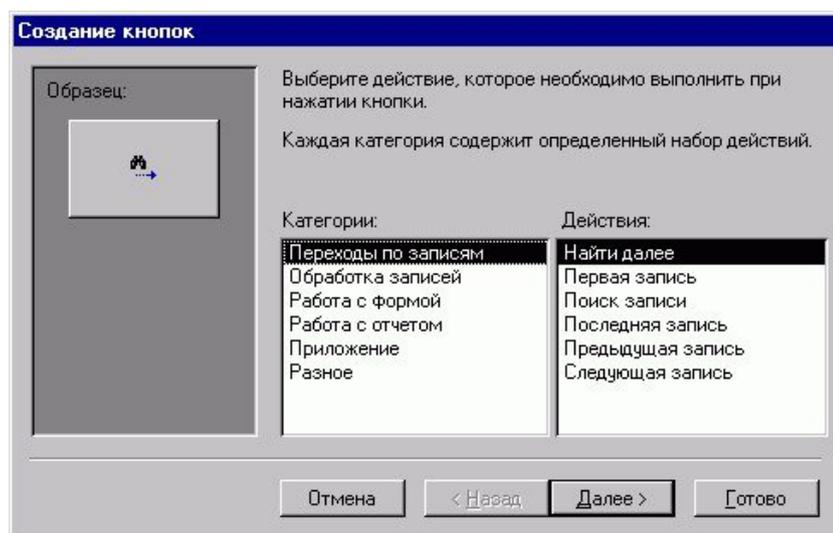
Заполнить таблица экспериментальным содержимым (5-7 записей).

Создайте для каждой таблицы форму, а для таблиц Студенты и Оценки – отчеты.

1. Выбрать таблицу и выполнить щелчок на кнопке Автоформа стандартной панели инструментов (команда меню Вставка ). Для таблицы будет создана форма по умолчанию.
2. Сохранить созданную форму, выполнив щелчок на кнопке закрытия окна, а затем на запрос о сохранении ответить утвердительно и ввести имя созданной формы или оставить предложенное для сохранения имя формы.
3. Используя команду Автоотчет создать указанные отчеты.

Создать в одной из созданных Форм кнопки:

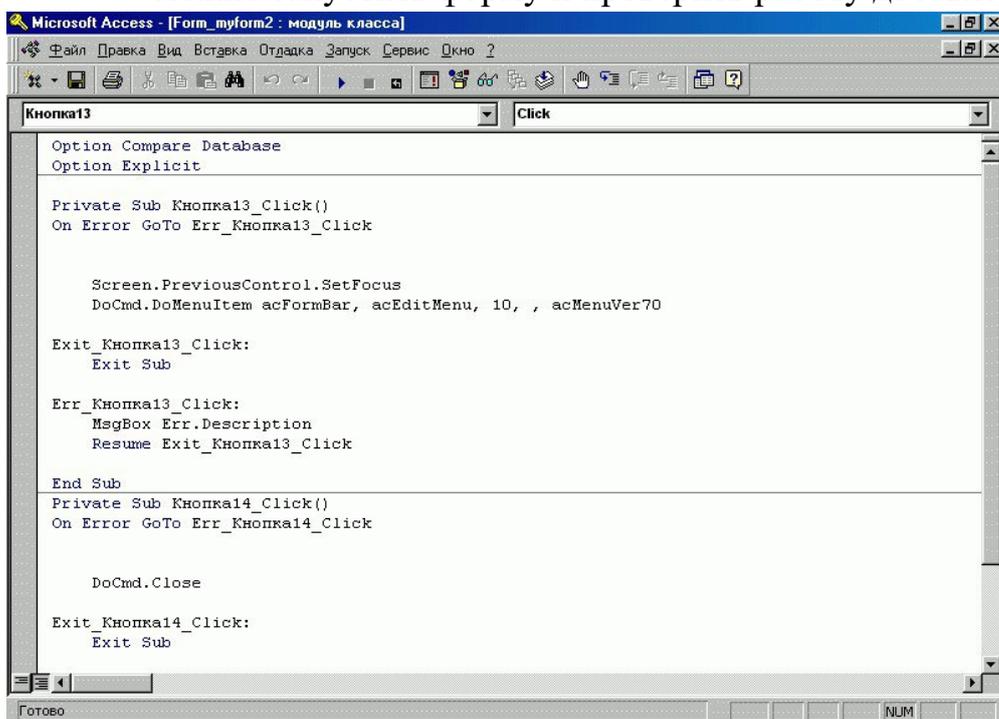
1. Выбрать вкладку Формы. Выбрать созданную форму.
2. Выполнить щелчок по кнопке Конструктор, для переключения в режим конструктора формы.
3. Выбрать команду Вид -> Панель элементов, если панель элементов не отображена на экране.
4. Проверить, активизированы ли мастера элементов управления (т.е. нажата ли кнопка Мастера).



5. Выполнить щелчок на элементе Кнопка.
6. Выполнить щелчок на свободном месте формы, чтобы вставить в нее кнопку. Откроется диалоговое окно мастера кнопок, показанное на рисунке 1.
7. В списке Категории выбрать пункт Переходы по записям, а затем в списке Действия -> Поиск записи. После щелчка на кнопке Готово в форму будет добавлена кнопка, выполняющая поиск.
8. Выполнить щелчок на элементе Кнопка.
9. Вставить кнопку на свободном месте формы ниже предыдущей, созданной ранее. Снова откроется окно кнопок.
10. В списке Категории выбрать пункт Работа с формой, а в списке Действия - пункт Закрытие формы. После щелчка по кнопке Готово в форму будет добавлена кнопка, выполняющая закрытие формы.

Рис. 2.

11. Сохранить форму, выбрав команду Файл -> Сохранить. После этого можно запустить форму и проверить работу добавленных кнопок.



12. Переключиться в режим Конструктора, выполнив щелчок по кнопке Конструктор.

13. Выбрать команду Вид -> Программа или выполнить щелчок по кнопке Программа на панели инструментов. Откроется окно редактора Visual Basic for Applications, представленное на рисунке 2.

В этом окне можно видеть три фрагмента программы на языке

VBA: раздел объявлений (в верхней части окна), который содержит код, относящийся ко всей форме в целом, а также два раздела кода, привязанных к двум кнопкам.

Первый раздел начинается с заголовка процедуры:

```
Private Sub Кнопка13_Click()
```

Заканчивается процедура оператором: End Sub. Операторы, заключенные между ними, выполняются после щелчка по кнопке Найти. Основная часть работы выполняется следующим оператором:

```
DoCmd.DoMenuItem acFormBar, AcEditMenu, 10, AcMenuVer70
```

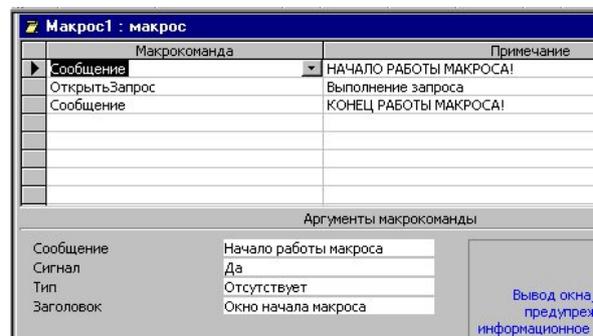
Этот оператор выполняет те же действия, что и команда меню Правка -> Найти, т.е. выводит на экран диалоговое окно Поиск в поле, с помощью которого можно выполнить поиск.

Открыв модуль VBA, его можно редактировать как текст в любом текстовом редакторе. В окне редактора Visual Basic в код можно вставить текст из другого файла. Для этого выбрать команду Вставка -> Файл, а в открывшемся диалоговом окне выбрать нужный файл и выполнить щелчок по кнопке ОК.

Создать три запроса: студенты женского пола; студенты, имеющие 2 по информатике; родители, имеющие двух детей – студентов.

**2. Создать макрос Макрос1, обеспечивающий выполнение следующих действий (рис.3):**

- выдачу сообщения “НАЧАЛО РАБОТЫ МАКРОСА”. Окно, в котором выводится текст, озаглавить как “Окно начала макроса”;
- выполнение запроса **Запрос1**, созданного в задании 1. В качестве первого аргумента - Имени запроса - выбрать Запрос 1. В качестве режима просмотра выбрать режим Таблица;
- выдачу сообщения “КОНЕЦ РАБОТЫ МАКРОСА”. Окно, в котором выводится текст, озаглавить как “Окно окончания макроса”.



Примечание. Для команды Сообщение обязательными параметрами, задаваемыми в нижней части диалогового окна (рис. 1) являются аргумент Сообщение, в окне которого записывается выводимый текст и аргумент Заголовок, в окне которого записывается текст заголовка окна.

**3. Выполнить макрос Макрос1. Убедиться в правильности выполнения.**

**4. Установить** с помощью кнопки на панели макросов режим пошагового выполнения и, выполнив макрос по шагам, пронаблюдать сообщения, выдаваемые на каждом шаге.

**5. Создать макрос Макрос2.**

5.1. Добавить в макрос **Макрос1**, после второй команды еще две команды:

- для выдачи информационного сообщения ПАЗУЗА после появления таблицы с результатом запроса;
- для закрытия окна с результатом выполнения запроса.

5.2. Сохранить макрос под новым именем **Макрос2**.

**6. Выполнить “Макрос2”. Убедиться в правильности выполнения.**

**7. Создать макросы Родители и Оценки, обеспечивающие открытие соответствующих форм.**

- a. На вкладке Макросы перейти в режим создания макросов.
  - b. Перетащить из вкладки Формы окна базы данных форму **Родители** в область Макрокоманда и проверить заполнение аргументов макрокоманды.
  - c. Сохранить макрос под именем **Родители**.
  - d. Проверить работоспособность макроса.
  - e. Повторить п. а. – d. для формы **Оценки**.
  - f. Создать командные кнопки с надписями “Родители” и “Оценки”, к которым присоединить соответствующие макросы.
  - g. Проверить работоспособность кнопок.
- 8. Сохранить** базу данных на диске для дальнейшего использования.

## Задание 2. Макросы с условиями

Для базы данных **База№6**, создать макрос **МакАдр\_1**, который выполняет следующие действия:

- выдает информационное сообщение о начале работы;
- открывает форму **Студенты**;
- выдает об этом информационное сообщение и приостанавливает работу до нажатия Кнопки;
- открывает отчет **Оценки**;
- выдает информационное сообщение об окончании работы.

1.1. Дать этому макросу имя **МакАдр\_1** и сохранить его.

- Выполнить макрос **МакАдр\_1** несколько раз и попытаться проследить последовательность действий. Обратит внимание на то, что форма **Студенты** и отчет **Оценки** остаются открытыми после работы макроса.

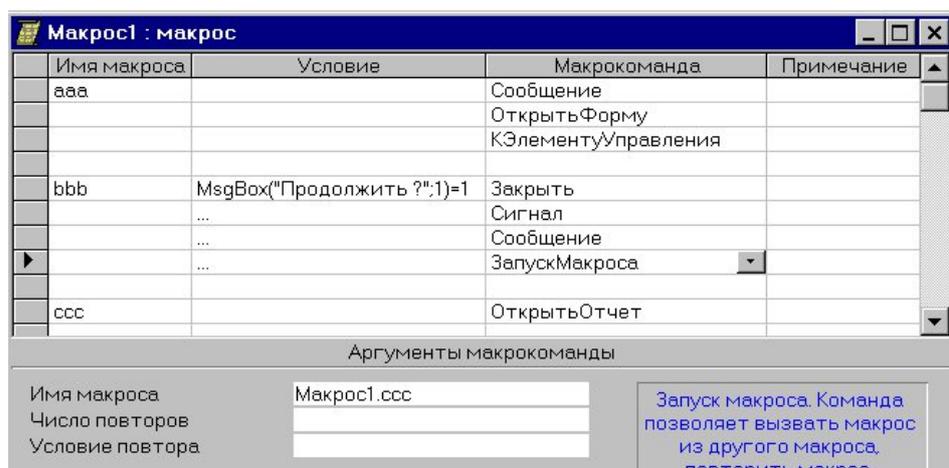
**2. Создать макрос МакАдр\_2**, выполняющий следующие действия:

- выдачу сообщения о начале работы;
- открытие формы **Родители**;
- выдачу информационного сообщения о предстоящем закрытии формы **Родители**;
- закрытие формы **Родители**;
- открытие отчета **Оценки**;
- выдачу предупреждающего сообщения о предстоящем закрытии отчета;
- закрытие отчета

**Математика.**

2.1. Дать этому макросу имя **МакАдр\_2** и сохранить его.

2.2. Выполнить макрос **МакАдр\_2** в



пошаговом и в автоматическом режимах.

### 3. Создать макрос **МакУсл\_1** с командой безусловного перехода.

3.1. Для макроса **МакАдр\_1**, пользуясь кнопками Панели инструментов **Макрос**, открыть столбец **Имя Макроса** (рис.4) и для каждой строки (команды) задать имена aaa, bbb, ccc, ddd, eee и т.д.

3.2. Вставить после первой команды Сообщение команду **Запуск макроса** с параметром eee, которая обеспечивает безусловный переход к команде eee после выполнения первой команды. Выполнить макрос и убедиться в правильности его работы.

3.3. Дать этому макросу имя **МакУсл\_1** и сохранить его.

### 4. Добавить в окне работы с макросами столбец **Условие** с помощью кнопок Панели инструментов **Макросы**.

### 5. Создать макрос **МакУсл\_2** (рис.4) с командой условного перехода, выполняющий следующие действия:

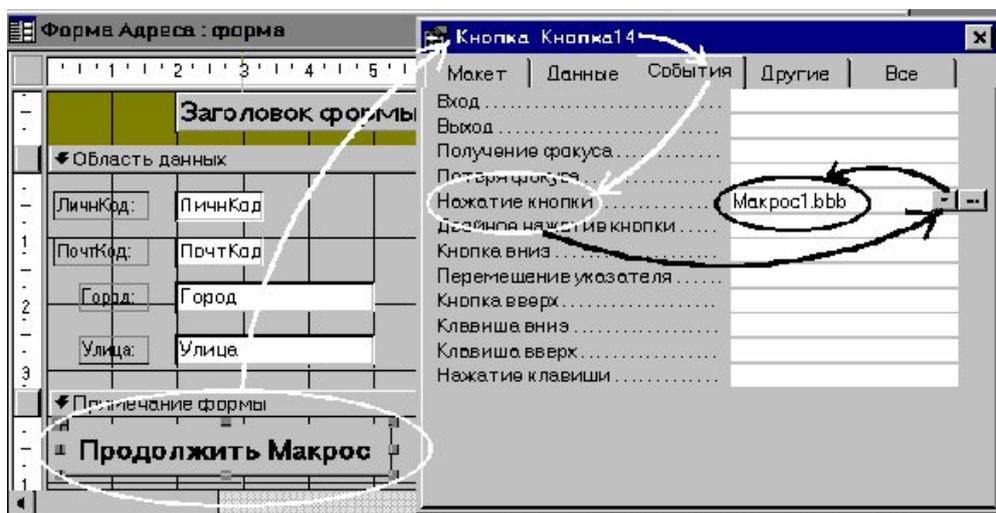
- выдачу стартового сообщения;
- открытие формы **Родители**;
- переход к элементу управления - полю **ТелРод**, представленному в форме **Родители**;
- закрытие формы **Родители**;
- подачу звукового сигнала;
- выдачу сообщения о завершении работы.

6.1. В качестве условия для выполнения закрытия формы **Родители** определить значение функции **MsgBox("Закреть ?";1)=1**. (Значение 1 функция **MsgBox()** принимает при нажатии кнопки, представленной в диалоговом окне, отображаемом этой функцией).

6.2. Каждой группе команд дать имена aaa, bbb, ccc (рис.4). Сохранить созданный макрос, не выполняя его.

6.3. Вызвать форму **Родители** и добавить в область примечания формы кнопку с надписью

“Продолжение макроса”, как показано на рис.5.



6.3.1. Для созданной кнопки в контекстном меню выбрать пункт Свойства и вкладку События. В разделе Нажатие кнопки указать необходимость выполнения команды bbb макроса **МакУсл\_2**.

6.3.2. Закрыть все вкладки и сохранить созданную форму.

Рис.5

6.4. Выполнить макрос **МакУсл\_2** в пошаговом режиме. Убедиться, что логика работы макроса верна и все его команды выполняются в нужной последовательности. Если макрос выполняется неправильно, проанализировать все сообщения, получаемые на каждом шаге, выявить причину и внести в макрос необходимые изменения.

## **7. Записать в отчет по практической работе:**

1. Определение, назначение макросов.
2. Последовательность создания макроса.
3. Отличие макросов с командами безусловного и условного переходов.
4. Как создать код VBA с помощью мастеров элементов управления.

## Лабораторная работа № 11.

### Создание базы данных и ее объектов средствами языка SQL

**Цель работы:** научиться создавать и изменять средствами SQL таблицы и индексы, задавать ограничения целостности.

#### Теоретические сведения:

Под созданием и изменением здесь подразумевается только создание и изменение структуры и параметров таблиц, а не хранящихся в них данных. Часть языка SQL, служащая для решения этих задач, называется языком описания данных (Data Definition Language – DDL).

**Создание таблицы.** Для создания таблицы в SQL служит команда CREATE TABLE. Синтаксис простейшего варианта команды CREATE TABLE:

CREATE TABLE <Имя таблицы > ( <имя поля > <тип данных>[( <размер>)], <имя поля > <тип данных>[( <размер>)] ... );

В стандартном языке SQL применяются следующие типы данных:

- INTEGER – до 10 цифр и знак;
- SMALL – до 5 цифр и знак;
- DECIMAL(p,q) –  $0 < p < 16$  всего позиций, q – цифр после запятой;
- FLOAT – вещественное, определяется СУБД (REAL в ACCESS);
- DOUBLE PRECISION – вещественное, определяется СУБД (FLOAT в ACCESS!!!), точность и диапазон больше, чем у FLOAT;
- CHAR(n) – строка из n ( $n < 256$ ) символов.

Практически во всех СУБД, поддерживающих SQL, применяются дополнительно следующие типы данных:

- VARCHAR(n) – строка из n символов ( $n_{\max} > 4096$  определяется СУБД);
- DATE – формат определяется специальной командой (по умолчанию mm/dd/yy);
- TIME – формат определяется специальной командой (по умолчанию hh.mm.ss);
- DATETIME – комбинация даты и времени;
- MONEY – денежный.

Подробнее о типах данных, поддерживаемых СУБД ACCESS, смотрите в справке ACCESS в ответе на вопрос «Типы данных SQL».

Пример команды на создание таблицы «Страна» (название, площадь, численность населения в млн чел.):

CREATE TABLE Страна (название CHAR(60), площадь REAL, население REAL);

*Самостоятельно* с помощью команды CREATE TABLE создайте таблицу «Изделие» со следующими атрибутами:

- название
- цена
- вес
- дата изготовления
- фирма-производитель

Подберите соответствующие типы данных.

Создание запросов на SQL в Access начинается вызовом конструктора запросов. Для этого в окне базы данных нужно выбрать объект «Запросы», пункт меню «Создать» и в окне «Новый запрос» пункт «Конструктор».

Далее, не выбирая таблицу, закройте окно «Добавление таблицы» и перейдите в режим SQL. Переход Режим SQL. → Вид → в режим SQL : меню Access

Набрав в окне SQL запрос на создание таблицы «Изделие», выполните его. Введите в созданную таблицу данные о трёх произвольных товарах. В режиме конструктора таблиц проверьте, как интерпретировал типы данных ACCESS.

**Внесение изменений в структуру таблицы** делается с помощью команды ALTER TABLE, имеющей следующий синтаксис:

```
ALTER TABLE <имя таблицы> {ADD {COLUMN <имя поля> <тип поля>[(<размер>)] [NOT NULL] [CONSTRAINT <имя индекса>]} | ALTER COLUMN <имя поля> <тип поля>[(<размер>)] CONSTRAINT <описание составного индекса>} | DROP {COLUMN <имя поля> | CONSTRAINT <имя индекса>}};
```

Команда, с помощью которой к таблице «Страна» добавляется поле «столица» выглядит так:

```
ALTER TABLE Страна ADD COLUMN столица VARCHAR(40) NOT NULL UNIQUE;
```

На поле «столица» наложены 2 ограничения: не допускается пустое поле (NOT NULL) и название столицы должно быть уникальным (UNIQUE).

Для добавления к таблице «Страна» поля, являющегося первичным ключом, служит команда

```
ALTER TABLE Страна ADD COLUMN Id_strana INTEGER NOT NULL PRIMARY KEY;
```

В поле Id\_strana должен храниться номер записи. Приведённая выше команда *не создаёт автоматического счётчика*.

*Самостоятельно* с помощью команд ALTER TABLE добавьте к таблице «Изделие» следующие поля:

- Id\_izdelie – номер записи и первичный ключ;
- сорт – сорт (1-й, 2-й, ...), не допускается пустых полей.

Замечание. Перед выполнением команды ALTER TABLE необходимо из таблицы «Изделие» удалить все записи, так как в имеющихся записях новые поля не могут иметь значение NULL, т.е. быть пустыми.

**Создание таблицы с ограничениями столбцов и ограничениями**

**таблицы.** В качестве примера создадим таблицу «отдых» связанную с таблицей «страна». Тип связи «многие к одному». Таблица «отдых» будет иметь следующие поля:

- Id\_otd – номер записи (первичный ключ);
- Id\_st - внешний ключ, связывающий с таблицей «Страна»;
- курорт – название курорта;

- гостиница;
- продолжит – продолжительность отдыха в днях.

Совокупность значений полей «Id\_st», «курорт» и «гостиница» должна быть уникальной, то есть, потенциальным ключом. Таким образом, в таблице «курорт» будет 2 ключа. Описанная таблица создаётся следующей командой: CREATE TABLE отдых ( Id\_otd INTEGER NOT NULL PRIMARY KEY, Id\_st INTEGER REFERENCES Страна(Id\_strana), курорт CHAR(80), гостиница CHAR(60), стоимость REAL, продолжительность SMALLINT, UNIQUE (Id\_st,курорт,гостиница));

*Самостоятельно* с помощью команд CREATE TABLE создайте таблицу «поставка», связанную с таблицей «Изделие».

Таблица «Поставка» должна иметь следующие поля:

- номер записи (первичный ключ),
- внешний ключ, связывающий с таблицей «Изделие»;
- адрес клиента,
- дату поставки,
- количество товара,
- стоимость доставки.

Совокупность значений внешнего ключа, адреса клиента и даты поставки должна быть уникальной, то есть, потенциальным ключом.

После создания таблицы «Поставка» введите в неё несколько записей. Для того, чтобы убедиться в правильной работе потенциального ключа, попытайтесь ввести две записи с одинаковыми значениями внешнего ключа, адреса клиента и даты поставки.

Добавьте таблицы «Изделие» и «Поставка» к схеме данных Схема→ (Меню ACCESS данных). ACCESS должен автоматически обнаружить связь «один ко многим», заданную при создании таблиц командой CREATE TABLE.

**Создание индекса.** Если таблица велика (обычно большой считается таблица, содержащая сотни тысяч записей), то для ускорения поиска в ней данных строятся индексы. Синтаксис команды для создания индекса следующий:

```
CREATE [UNIQUE] INDEX <имя индекса>
ON <имя таблицы> <(список полей)>
```

Для построения индекса по полю «название» в таблице «Страна» служит следующая команда:

```
CREATE UNIQUE INDEX названиеIn
ON страна(название)
```

*Самостоятельно* постройте уникальный индекс для поля «Название» таблицы «Изделие». Проверьте его действие, введя две записи с одним наименованием изделия.

## Лабораторная работа №12-13

### Выполнение простых запросов к БД в SQL.

**Цель:** научиться использовать операторы языка SQL для работы с данными БД.

**Задание 1.** в режиме таблицы создайте таблицы «Транспортная\_Компания» и «Доставка\_Транспорт»(см. Приложение1). Таблицу «Клиенты» и«Сотрудники» создайте в режиме SQL. (см. Приложение1). **Заполните таблицу восьмью записями.**Создайте связи между таблицами.

#### Создание запросов

Для создания запроса необходимо перейти на вкладку «Создать» и выполнить «Создание запроса в режиме конструктора».

В появившемся диалоге добавления таблиц в конструкторе запросов нажать «Заккрыть», а затем на панели «Конструктор запросов» выбрать SQL - режим (рис. 1). Переключаться между видом конструирования запроса можно, используя контекстное меню.

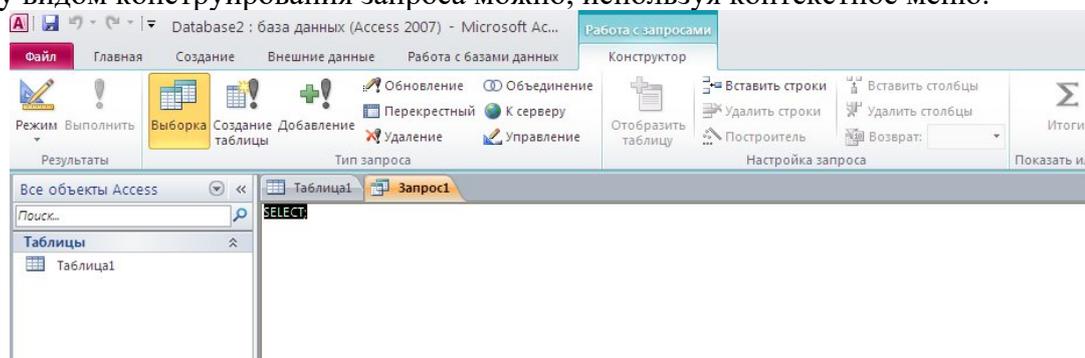


Рис. 1

#### Простой запрос на выборку

**Задание 2:** из таблицы «Транспортная\_Компания» выбрать все транспортные компании, кроме «Кенгуру».

1. Создайте запрос на выборку на основании таблицы«Транспортная\_Компания» (поле «Название компании»).

2. Выполните запрос.

3. Перейдите в режим SQL («Вид(View)» - SQL). Измените код таким образом:

```
SELECT Название компании  
FROM Транспортная_Компания  
WHERE Название компании <> 'Кенгуру';
```

4. Выполните запрос, убедитесь, что он выполняется корректно.

5. Вернитесь в режим SQL. Программно измените запрос таким образом, чтобы он выводил не только название компании, но и адрес (именя полей«Улица», «Город», «Штат»). Выполните запрос.

6. В режиме SQL измените запрос таким образом, чтобы выводились только транспортные компании, расположенные в городе Денвер. Выполните запрос.

7. Сохраните исходный и измененные запросы под именем «SQL\_Выборка 1», «SQL\_Выборка 2» и т.д.

8. Сделайте сортировку выбранных записей по названию компании в алфавитном порядке:

```
ORDER BY Название компании;
```

9. Выполните запрос. Сохраните изменения.

#### Запросы на обновление данных

**Задание 3:** создать запрос на увеличение тарифа на 0,8 таблицы «Доставка\_Транспорт».

#### Задание 4: удалить таблицу «Доставка\_Транспорт»

1. В конструкторе запросов выберите режим SQL.
2. Введите следующий код:  
DROP TABLE Доставка\_Транспорт
3. Выполните запрос, убедитесь, что выбранная таблица удалена.
4. Сохраните запрос под именем «Удаление\_Таблицы».

#### Запросы на добавление и удаление данных

**Задание 5:** создать запрос на добавление данных в таблицу «Клиенты». Введите 2 новые записи. Сохраните запрос «Добавление\_Клиенты»

**Задание 6:** создать запрос на удаление данных из таблицы «Сотрудники». Введите 2 новые записи. Сохраните запрос «Удаление\_Сотрудники»

#### Запрос на объединение

Данный вид запросов используется для объединения идентично структурированных данных.

**Задание 7:** создать запрос, возвращающий список всех контактов компании (клиенты, сотрудники) с номерами телефонов, отсортированный в алфавитном порядке.

1. В конструкторе запросов выберите тип запроса – объединение (автоматически работа будет происходить в режиме SQL).
2. Введите следующий код:  
SELECT Фамилия, Имя, НомерТелефона FROM Клиенты UNION  
SELECT Фамилия, Имя, Телефон FROM Сотрудники;
3. Для объединения двух полей «Фамилия» и «Имя» в одно – «ФамилияИмя» измените код следующим образом:  
SELECT Фамилия & " " & Имя AS ФамилияИмя, НомерТелефона FROM  
Клиенты  
UNION  
SELECT Фамилия & " " & Имя AS ФамилияИмя, Телефон FROM Сотрудники;
4. Сохраните запрос под именем «Список\_Телефонов».

ФамилияИмя	НомерТелефона
Акерс Ким	(604) 555-0193
Албук Стив	(604) 555-0192
Андерсон Эмми	(206) 555-0105
Аскерман Пилар	(206) 555-0194

Рис. 2.1. Результат выполнения запроса «Список\_Телефонов».

#### Сохранение и закрытие базы данных

1. Сохраните все изменения.
2. Завершите работу с Microsoft Access.

#### Вопросы

1. Что такое SQL, назначение языка SQL?
2. Назначение команды SELECT?
3. Что такое внешнее и внутреннее объединение, чем отличаются?
4. Что такое левое, правое и полное объединение?
5. Что такое перекрестный запрос?
6. Как применить агрегатную функцию?
7. Для чего в стандарт SQL2 были введены объединения?

8. Чем отличается использование WHERE от HAVING?
9. Чем отличается использование DISTINCT от группировки?
10. Назначение команды DROP?
11. Назначение команды DELETE?
12. Назначение команды UPDATE?
13. Назначение команды INSERT?
14. Назначение команды CREATE?

## Приложение 1.

**Таблица Доставка\_Транспорт**

Имя поля	Тип	Размер	Другие свойства
КодДоставки	Счетчик	Длинное целое	Ключ
КодТранспортнойКомпании	Числовой	Длинное целое	Обязательное поле
ДатаДоставки	Дата/время		Обязательное поле; Отображать элемент выбора даты (Show Date Picker) – для дат
Тариф	Денежный		Число десятичных знаков – 2.

**Таблица Транспортная\_Компания**

Имя поля	Тип	Размер	Другие свойства
КодТранспортнойКомпании	Счетчик	Длинное целое	Ключ
НазваниеКомпании	Текстовый	255	Обязательное, Индексированное – да (совпадения не допускаются)
Фамилия	Текстовый	20	Обязательное
Имя	Текстовый	20	Обязательное
Улица	Текстовый	50	Обязательное
Город	Текстовый	20	Обязательное
Штат	Текстовый	20	Обязательное
Индекс	Текстовый	7	Маска ввода 9999999;;
Телефон	Текстовый	14	Обязательное, Маска ввода (999) 000-0000

**Таблица Клиенты**

<b>Имя поля</b>	<b>Тип</b>	<b>Размер</b>	<b>Другие свойства</b>
КодКлиента	Текстовый	5	Ключ
Имя	Текстовый	30	Обязательное
Фамилия	Текстовый	50	Обязательное
Адрес	Текстовый	255	Обязательное
Город	Текстовый	50	Обязательное
Штат	Текстовый	20	Обязательное
ПочтовыйИндекс	Текстовый	7	Маска ввода 9999999;
Страна	Текстовый	50	Обязательное
НомерТелефона	Текстовый	15	Маска ввода (999) 999-9999
КредитнаяКарта	Числовой	Байт	Значение по умолчанию 0
НомерКредитнойКарты	Текстовый	255	Индексированное – ДА (совпадения не допускаются)

**Таблица *Сотрудники***

<b>Имя поля</b>	<b>Тип</b>	<b>Размер</b>	<b>Другие свойства</b>
КодСотрудника	Счетчик	Длинное целое	Ключ
Имя	Текстовый	20	Обязательное
Фамилия	Текстовый	20	Обязательное поле; Индексированное – Да (совпадения допускаются)
Должность	Текстовый	50	Обязательное поле;
			Индексированное – Да (совпадения допускаются)
Адрес	Текстовый	100	Обязательное
Город	Текстовый	20	Обязательное
Штат	Текстовый	20	Обязательное
Индекс	Текстовый	7	Маска ввода 9999999;;
Телефон	Текстовый	14	Обязательное, Маска ввода (999) 000-0000
ДатаРождения	Дата/время		Обязательное поле; Маска ввода – краткий формат даты.
ДатаНайма	Дата/время		Обязательное поле; Условие на значение <=Now(); Сообщение об ошибке – «Дата должна быть не позднее текущей». Отображать элемент выбора даты (Show Date Picker) – для дат
Заметки	Поле МЕМО		Необязательное

Лабораторная работа №14  
**Выборка данных из БД. Модификация содержимого БД.**

**Цель работы:** в ходе выполнения лабораторной работы научиться использовать операторы языка SQL для работы с данными БД.

**Задание:**

1. Средствами SQL создайте БД «Деятельность музея», состоящую из следующих таблиц:

Экспонаты : таблица			
	Имя поля	Тип данных	
?	Код экспоната	Числовой	
	Наименование	Текстовый	
	Код зала	Числовой	
	Дата поступления	Дата/время	
	Автор	Текстовый	
	Материал	Текстовый	
	Техника	Текстовый	

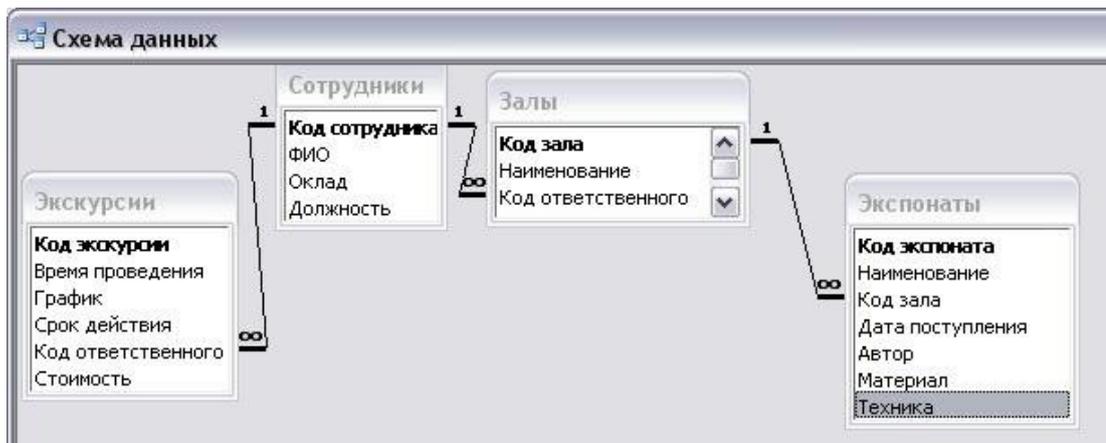
Экспонаты : таблица			
	Имя поля	Тип данных	
?	Код экспоната	Числовой	
	Наименование	Текстовый	
	Код зала	Числовой	
	Дата поступления	Дата/время	
	Автор	Текстовый	
	Материал	Текстовый	
	Техника	Текстовый	

Залы : таблица			
	Имя поля	Тип данных	
?	Код зала	Числовой	
	Наименование	Текстовый	
	Код ответственного	Числовой	

Залы : таблица			
	Имя поля	Тип данных	
?	Код зала	Числовой	
	Наименование	Текстовый	
	Код ответственного	Числовой	

Экскурсии : таблица			
	Имя поля	Тип данных	
?	Код экскурсии	Числовой	
	Время проведения	Дата/время	
	График	Текстовый	
	Срок действия	Дата/время	
	Код ответственного	Числовой	
▶	Стоимость	Денежный	▼

2. Создайте схему данных:



3. С помощью SQL-запроса изменим структуру таблицы Залы. Добавим атрибут Площадь, тип данных – числовой.

4. Заполните таблицы записями:

**Экспонаты : таблица**

Код эк	Наименование	Код зала	Дата поступления	Автор	Материал	Техника
1	Картина "Праздник русской зимы"	2	06.12.1991	Леднев В.	бумага	акварель
2	Картина "Удильщики"	1	12.12.1989	Васнецов В.	бумага	граф. и ит.
3	Картина "Дубровник. Храм"	3	28.04.1993	Леднев В.	бумага	пастель
4	Картина "Участник войны"	1	03.11.1993	Перов В.П.	картон	масло
5	Картина "Подсолнушек"	2	30.12.1989	Попов И.П.	холст	масло
6	Картина "Витязь"	3	26.05.1989	Васнецов В.	холст	масло
7	Картина "Крестьяне с колесом"	2	24.01.1995	Васнецов В.	бумага	граф. и ит.
8	Портрет графический- Валиков Иван Алексеевич	1	03.11.1993	Перов В.П.	бумага	стекло
9	Эскиз к рассказу В.М.Шукшина "Верую"	2	30.12.1989	Новодежкин И.Н.	бумага	тушь
10	картина "Вечер в Венеции"	1	20.10.1994	Айвазовский И.	картон	масло
0		0				

Запись: 10 из 10

---

**Залы : таблица**

Код зала	Наименование	Код ответственного	Площадь
1	Главный	1	185
2	Бронзовый	5	84
3	Рубиновый	7	112
4	Золотой	3	68
5	Современный	1	80
0		0	

Запись: 6 из 6

---

**Сотрудники : таблица**

Код сотрудник	ФИО	Оклад	Должность
1	Мухина Д.Н.	6 700,00р.	контроллер
2	Лисина Л.Б.	8 800,00р.	экскурсовод
3	Бондаренко А.	7 600,00р.	гл. контроллер
4	Коваленко Т.Ю	8 800,00р.	экскурсовод
5	Астахов Е.А.	6 700,00р.	контроллер
6	Быков А.М..	10 200,00р.	экскурсовод
7	Лучко К.С.	6 700,00р.	контроллер
8	Булгаков Р.М.	9 000,00р.	искусствовед
9	Мечникова П.И	4 500,00р.	тех. Сотрудник
10	Крамарова М.С	4 500,00р.	тех. сотрудник
0		0,00р.	

Запись: 11 из 11

Код экскурсии	Время проведения	График	Срок действия	Код ответственного	Стоимость
1	10:00:00	суббота	31.05.2010	2	150,00р.
2	11:30:00	ежедневно	31.05.2010	2	60,00р.
3	15:15:00	по четным	30.06.2010	4	80,00р.
4	12:00:00	выходные	30.06.2010	4	120,00р.
5	15:00:00	по нечетным	01.08.2010	2	90,00р.
6	11:00:00	ежедневно	01.08.2010	6	60,00р.
7	17:00:00	с 1-3 каж. Мес	01.08.2010	6	130,00р.
8	14:00:00	среда	30.06.2010	6	100,00р.
9	12:45:00	среда	30.06.2010	2	100,00р.
10	16:00:00	вторник	01.08.2010	6	120,00р.
* 0				0	0,00р.

Запись: 5 из 10

- Запрос на добавление данных в таблицу Залы:  
Добавьте запись [6, "Серебряный", 1, 145] в соответствующие столбцы;
- Запрос на удаление данных из таблицы Сотрудники Удалим записи с фамилией сотрудника, содержащей « Мечникова»:
- Запрос на обновление данных. В таблице Издания увеличим стоимость объявлений изданий с кодами 1711 и 1712.
- Запрос на создание таблицы Расписание:

В итоге должны получить:

Время проведения	График	ФИО
10:00:00	суббота	Лисина Л.Б.
11:30:00	ежедневно	Лисина Л.Б.
15:15:00	по четным	Коваленко Т.Ю.
12:00:00	выходные	Коваленко Т.Ю.
15:00:00	по нечетным	Лисина Л.Б.
11:00:00	ежедневно	Быков А.М..
17:00:00	с 1-3 каж. мес.	Быков А.М..
14:00:00	среда	Быков А.М..
12:45:00	среда	Лисина Л.Б.
16:00:00	вторник	Быков А.М..

**9. Запросы SQL для выборки данных из таблиц Залы и Сотрудники.**

В результате должны получить:

**Запрос4**

ФИО	Наименование	Площадь
Мухина Д.Н.	Главный	185
Мухина Д.Н.	Современный	80
Бондаренко А.С.	Серебряный	145
Бондаренко А.С.	Золотой	68
Астахов Е.А.	Бронзовый	84
Лучко К.С.	Рубиновый	112

**10. Выберем экскурсии, стоимость которых больше 100р и отсортируем по возрастанию цены:**

Результат:

**Запрос5**

Код экскурсии	Время проведения	График	Срок действия	Стоимость
10	16:00:00	вторник	01.08.2010	120,00р.
4	12:00:00	выходные	30.06.2010	120,00р.
7	17:00:00	с 1-3 каж. Мес.	01.08.2010	130,00р.
1	10:00:00	суббота	31.05.2010	150,00р.

**11. Вычислим средний оклад сотрудников музея по должностям:**

Результат:

**Запрос6**

Должность	Средний оклад
гл. контроллер	9 120,00р.
искусствовед	9 000,00р.
контроллер	8 040,00р.
тех. сотрудник	4 500,00р.
экскурсовод	9 266,67р.

**12. Вычислим, сколько экскурсий закреплено за каждым экскурсоводом с помощью групповой операции Count:**

Результат:

**Запрос7**

ФИО	Количество экскурсий
Коваленко Т.Ю.	2

Запрос7	
ФИО	Количество экскурсий
Лисина Л.Б.	4
Быков А.М..	4

### 13.Создание однотабличного подзапроса:

```
SELECT Сотрудники.ФИО, Экскурсии.График, Экскурсии.Стоимость
FROM Сотрудники INNER JOIN Экскурсии ON Сотрудники.[Код сотрудника]
= Экскурсии.[Код ответственного]
WHERE (Экскурсии.[Код экскурсии]) In (SELECT [Код экскурсии]
FROM Экскурсии
WHERE ([Время проведения]<#12:0:0#));
```

Запрос8		
ФИО	График	Стоимость
Лисина Л.Б.	суббота	150,00р.
Лисина Л.Б.	ежедневно	60,00р.
Быков А.М..	ежедневно	60,00р.

### 14.Создание многотабличного подзапроса:

Выберем тех сотрудников, которые ответственны за залы с площадью большей 100 м2:

```
SELECT Сотрудники.[Код сотрудника], Сотрудники.ФИО,
Сотрудники.Должность
FROM Сотрудники
WHERE (Сотрудники.[Код сотрудника]) In (SELECT Залы.[Код
ответственного]
FROM Залы
WHERE (Залы.Площадь)>100);
```

Запрос9		
Код сотрудника	ФИО	Должность
1	Мухина Д.Н.	контроллер
3	Бондаренко А.С.	гл. контроллер
7	Лучко К.С.	контроллер

**При необходимости в пунктах 13-14 исправьте ошибки.**

## Лабораторная работа №15

### Применение встроенных механизмов защиты информации в системах управления базами данных.

**Цель работы.** Изучение способов защиты информации в БД.

#### Краткие сведения из теории.

Система безопасности БД должна обеспечивать физическую целостность БД и защиту от несанкционированного вторжения с целью чтения содержимого и изменения данных.

Защита БД производится на двух уровнях:

- на уровне пароля;
- на уровне пользователя (защита учетных записей пользователей и идентифицированных объектов).

Для защиты БД Access использует файл рабочих групп `system.mdw` (рабочая группа - это группа пользователей, которые совместно используют ресурсы сети), к которому БД на рабочих станциях подключаются по умолчанию. Файл рабочих групп содержит учётные записи пользователей и групп, а также пароли пользователей. Учётным записям могут быть предоставлены права на доступ к БД и её объектам, при этом сами разрешения на доступ хранятся в БД.

Для обеспечения защиты БД Access необходимо создать рабочую группу, используя файл - администратор рабочих групп `wrkadm.exe`. При создании уникальной рабочей группы задается имя пользователя, название организации и код рабочей группы.

Файл рабочей группы MS Access содержит следующие встроенные учётные записи:

1. Admins - стандартная учётная запись пользователя. Данные записи являются одинаковыми для всех экземпляров Ms Access;
2. Admin - учётная запись группы администратора - является уникальной в каждом файле рабочей группы;
3. Users - содержит учётные записи пользователей.

Для создания файла рабочих групп необходимо выйти из Access и в папке system или system32 в каталоге windows найти файл рабочей группы и создать новую рабочую группу ( может быть до 20 цифровых или буквенных обозначений). Группа Admins может содержать произвольное число пользователей, но владелец объекта всегда один (владельцем объекта может быть учётная запись, которая создавала объект или которой были переданы права на его использование).

Так как чтение записи Admin возможно для всех рабочих групп и данные учётные записи являются одинаковыми, то пользователя ADMIN необходимо удалить из группы администраторов, для чего следует создать новую учётную запись администратора и задать пароль на его учётные записи и на учетные записи владельца. Зашифровать и дешифровать базу данных могут только её владелец и члены группы Admins. Для шифрования Jet использует алгоритм RSA (назван по первым буквам фамилий его изобретателей: Rivest, Shamir, Adelman) с ключом на основе идентификатора рабочей группы.

У шифрования базы данных имеется два негативных побочных эффекта. Во-первых, снижается её быстродействие - по оценкам Microsoft, процентов на 10-15. Во-вторых, зашифрованную базу данных нельзя сжимать такими программами, как PKZip, LHA, Stacker и DriveSpace. Точнее, сжимать можно, только в этом нет смысла - её размер уменьшится незначительно.

#### **Разграничение прав доступа пользователей**

Разрешения к доступу называются явными, если они принадлежат или присвоены учётной записи пользователя. Разрешения будут неявными, если они присвоены учётной записи группы, при этом пользователь, включённый в группу получает все её разрешения.

#### **ТИПЫ РАЗРЕШЕНИЙ НА ДОСТУП К БД**

Разрешения	Разрешённые действия	Объекты БД
Открытие и запуск	Открытие БД, формы или отчёта	БД, формы, отчёты, макросы
Монопольный доступ	Монопольное открытие БД	БД
Чтение макета	Просмотр объектов в режиме конструктора	Таблицы, запросы, формы, отчёты, макросы и модули
Изменение макетов	Просмотр и изменение макетов, удаление	Таблицы, запросы, формы, отчёты, макросы и модули
Разрешения администратора	Установка пароля в БД, репликация БД	Предоставление прав доступа другим пользователям
Чтение данных	Просмотр данных	Таблицы и запросы
Обновление данных	Просмотр и изменение данных без удаления и вставки	Таблицы и запросы
Вставка данных	Просмотр и вставка данных без	Таблицы, запросы

	удаления и изменения	
Удаление данных	Просмотр и удаление данных без из изменения и вставки	Таблицы, макросы

Полномочия пользователя определяются по минимальным разрешениям доступа. Изменить разрешения для пользователей могут члены группы Admins, владелец объекта и пользователь, получивший на этот объект разрешения администратора. При подключении к БД пользователи получают права групп, которым они принадлежат.

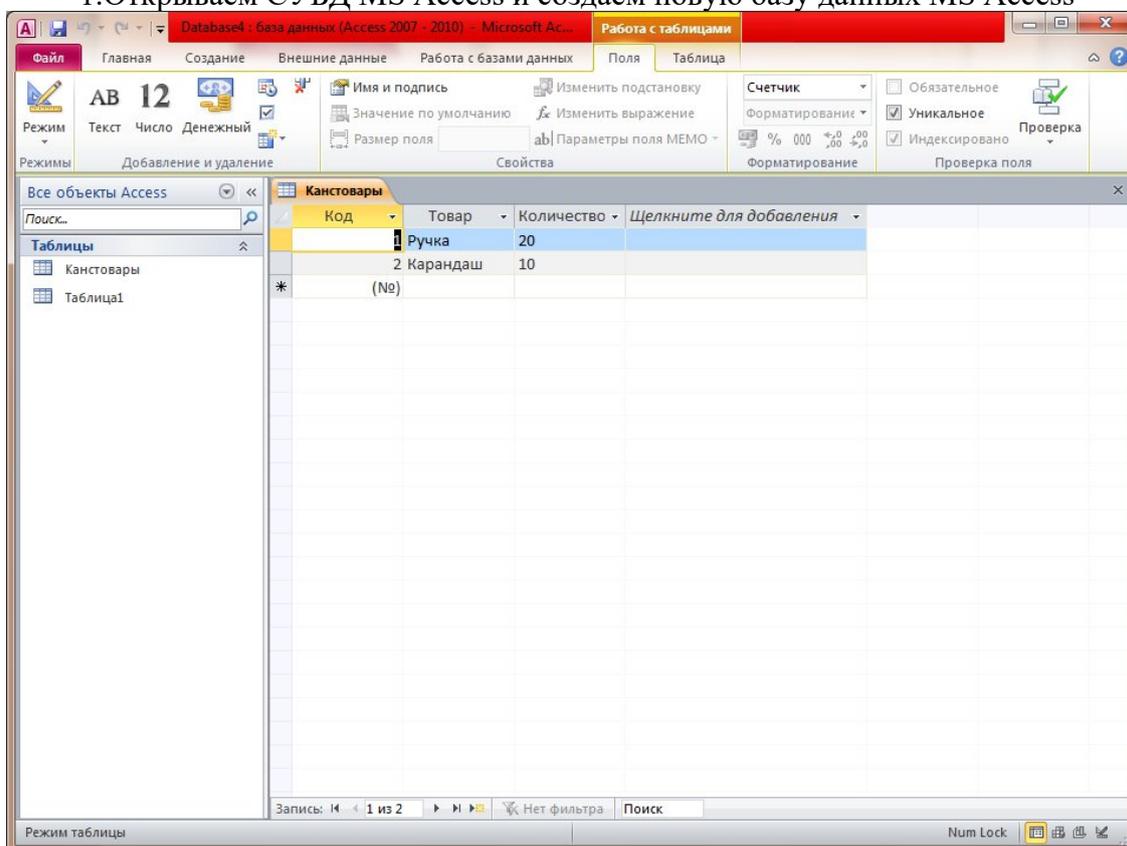
#### Задание к работе.

1. Создать новую базу данных MS Access
2. Зашифровать базу данных паролем
3. Расшифровать базу данных.

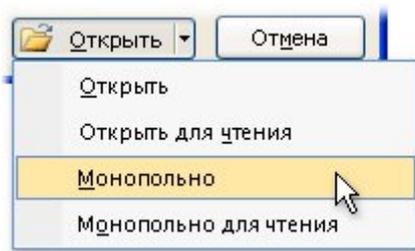
В Access 2010 не поддерживается защита на уровне пользователя для баз данных, созданных в новом формате (ACCDB и ACCDE-файлы). Однако при открытии базы данных из более ранней версии Access, имеющей защиту на уровне пользователя, в Access 2010 эти параметры будут продолжать действовать. Поэтому рассмотрим алгоритм защиты на уровне пароля.

#### Алгоритм защиты БД MS Access.

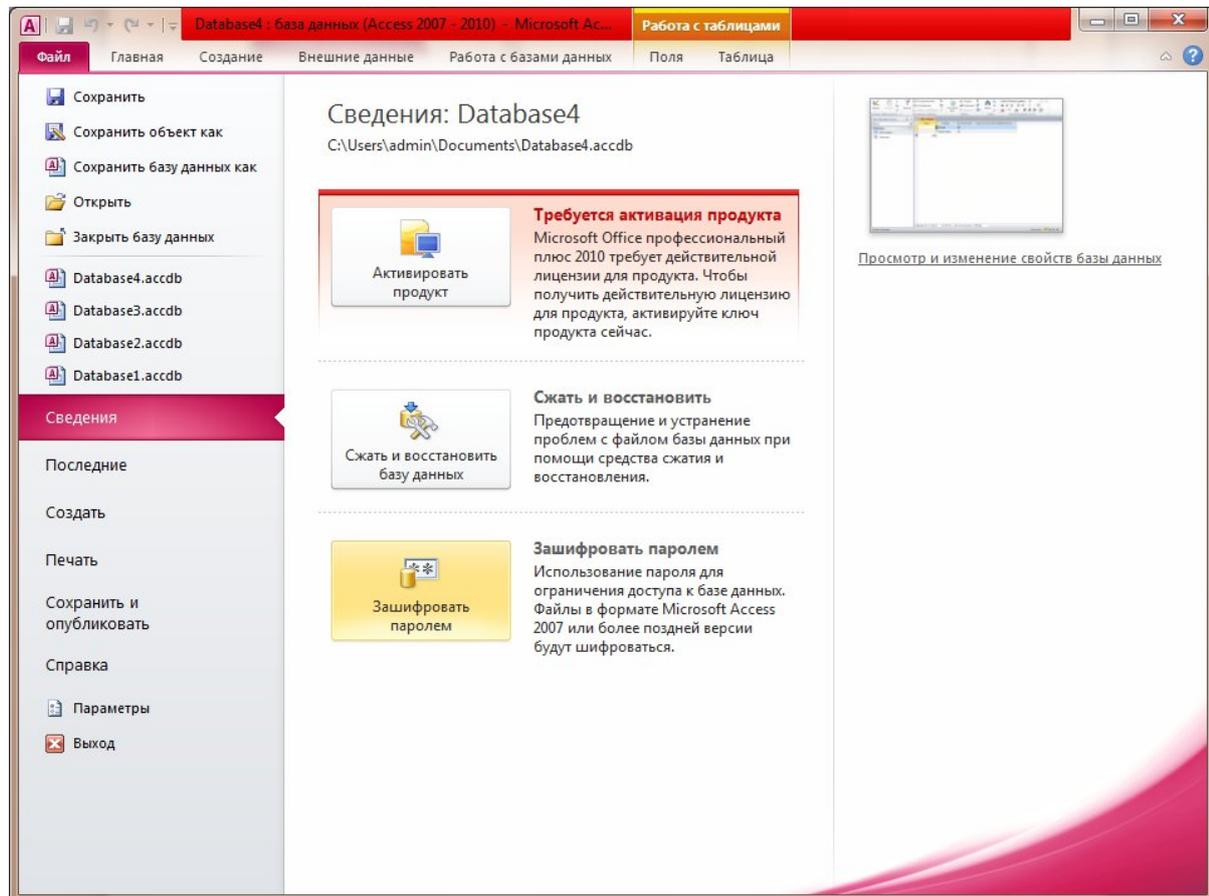
##### 1. Открываем СУБД MS Access и создаем новую базу данных MS Access



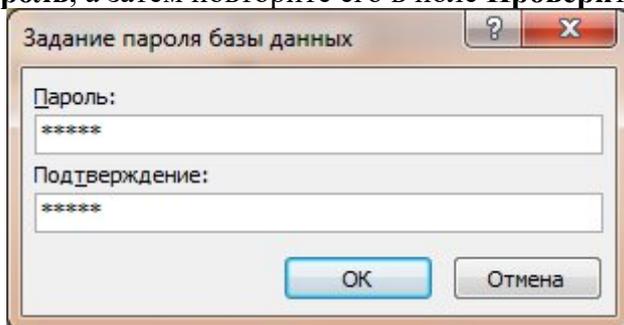
2. Сохраняем созданную БД и закрываем эту БД.
3. Вновь открываем БД но уже в монопольном режиме.
4. На вкладке **Файл** нажмите кнопку **Открыть**.
5. В диалоговом окне **Открыть** найдите файл, который нужно открыть, и выделите его.
6. Щелкните стрелку рядом с кнопкой **Открыть** и выберите команду **Монопольно**.



7. На вкладке **Файл** нажмите кнопку **Сведения** и выберите пункт **Зашифровать паролем**.

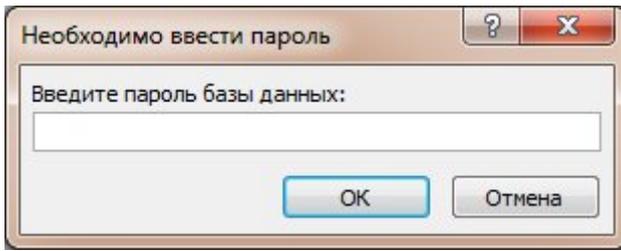


8. Откроется диалоговое окно **Задание пароля** базы данных. Введите пароль в поле **Пароль**, а затем повторите его в поле **Проверить**.

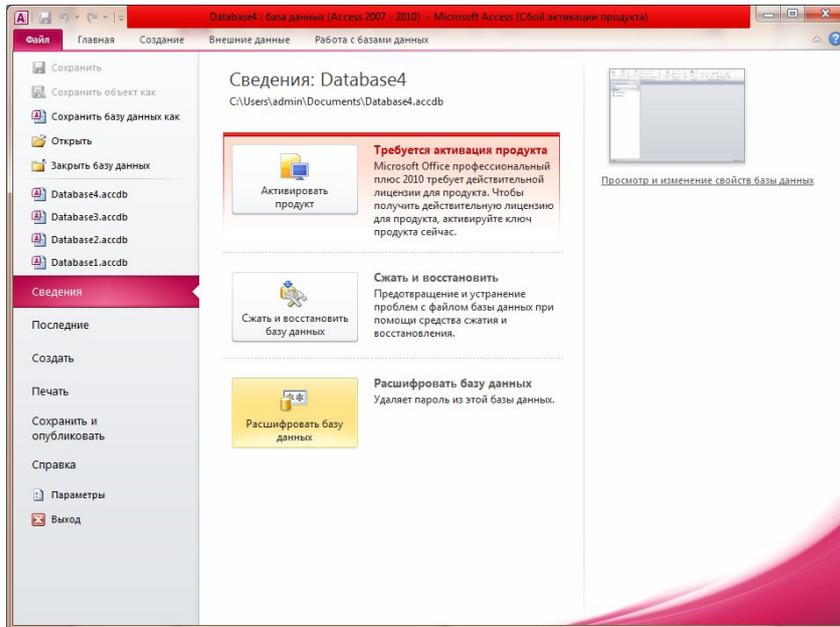


9. Закрыть БД.

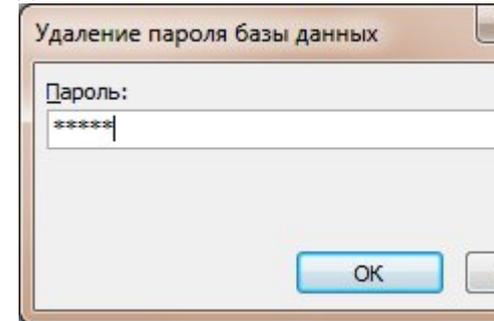
10. Открыть защищенную паролем БД в монопольном режиме. Появится окно «Необходимо ввести пароль».



11. Для удаления созданного пароля необходимо зайти во вкладке **Файл** нажать кнопку **Сведения** и выбрать пункт **Расшифровать базу данных**.



12. Появится окно **Удаления пароля баз данных**.



**Контрольные вопросы:**

1. Способы защиты информации в БД Access.
2. Группы и пользователи БД Access .
3. Файл рабочей группы.
4. Объекты БД Access и права доступа к объектам.
5. Понятие владельца объекта.

Алгоритм защиты БД Access

#### 4. Информационное обеспечение обучения

##### Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

###### Основные источники:

1. Администрирование Microsoft SQL Server : учебный курс : пер. с англ. – 2-е изд., испр. – М.: Издательско-торговый дом «Русская редакция», 2012. – 640 с.
2. Аскеров, Т. М. Защита информации и информационная безопасность : учебное пособие / Под общей ред. Курбакова К. И., Т. М. Аскеров. – М.: Рос.экон. академия, 2011. – 435 с.
3. Бормотов, С. В. Системное администрирование на 100% / С. В. Бормотов. – СПб.: Питер, 2012. – 256 с. (+CD).
4. Вискадул, Б. Д. Основы компьютерных сетей : учебное пособие / Б. Д. Вискадул [и др.]. – М.: Издательский дом «ФОРУМ», ИНФРА-М, 2013. – 272 с.
5. Галатенко, В. А. Основы информационной безопасности : курс лекций / В. А. Галатенко. – М.: ИНТУИТ.РУ «Интернет-Университет Информационных Технологий», 2013. – 280 с.
6. Галкин, В.А. Телекоммуникации и сети : учебное пособие для вузов / В. А. Галкин, Ю. А. Григорьев. – М.: Издательство МГТУ им. Н.Э. Баумана, 2012.
7. Голицына, О. Л. Базы данных : учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – 2-е изд., испр. и доп. – М.: ФОРУМ: ИНФРА-М, 2012. – 400 с.: ил. – (Профессиональное образование).
8. Горев, А. Эффективная работа с СУБД / А. Горев, Р. Ахаян, С. Макашарипов. – СПб.: Питер, 2012. – 740 с.
9. Дейт, К. Дж. Введение в системы баз данных : пер. с англ. / К. Дж. Дейт. – 7-е издание. – М.: «Вильямс», 2011. – 1072 с.: ил. – Парал. тит. англ.
10. Дюбуа, MySQL : пер. с англ. / Дюбуа, Поль. – 2-е изд. – М.: Издательский дом «Вильямс», 2012. – 1056 с.
11. Мартин, Г. SQL.: перевод П. Быстров / Г. Мартин. – М: Изд. Лори, 2013. – 644с.
12. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы. / В. Г. Олифер, Н. А. Олифер. — СПб.: Питер, 2011.
13. Опель, Э. Изучаем SQL / Джим Киу : пер. с англ. / Д. А. Терентьева, Э. Опель. – М. НТ Пресс, 2012. – 320.: ил. – (Раскрытие тайн).

###### Дополнительные источники:

1. Агальцов, В. П. Базы данных : учебное пособие / В. П. Агальцов. – М.: Мир, 2008.
2. Агальцов, В. П. Базы данных: учебное пособие / В. П. Агальцов. – М.: Мир, 2008.
3. Андерсон, К. Локальные сети : полное руководство : пер. с англ. / К. Андерсон, М. Минаси. – К.: ВЕК+, М.: ЭНТРОП, СПб.: КОРОНА принт, 1999. – 356 с.
4. Анин, Б. Ю. Защита компьютерной информации / Б. Ю. Анин. – СПб.: БХВ - Санкт-Петербург, 2000. – 325 с.
5. Бабаш, А.В., Информационная безопасность. Лабораторный практикум + CD/Бабаш А.В, Баранова Е.К., Мельников Ю.Н –КноРус, 2012 -136 с.
6. Блек, Ю. Сети ЭВМ: протоколы, стандарты, интерфейсы / Ю. Блек – Мир, 1990. – 317 с.
7. Галатенко, В.А. Стандарты информационной безопасности: курс лекций: учебное пособие/ В.А. Галатенко – М.: ИНТУИТ. РУ «Интернет- университет

Информационных технологий», 2009. – 264 с.

8. Документация по MySQL. – Режим доступа: [http:// www.mysql.ru/docs](http://www.mysql.ru/docs).
9. Евсеев, Д.А., Трофимов В.В. Web – дизайн в примерах и задачах: учебное пособие/ Евсеев Д.А., Трофимов В.В.: под ред. Трофимова В.В. – М.: КНОРУС, 2010 г. – 272
10. Калинин, И. А. Основы информационной безопасности при работе в телекоммуникационных сетях. Элективный курс : учебное пособие / И. А. Калинин. – М.:БИНОМ. Лаборатория знаний, 2008. – 199 с.
11. Касперский, Е. В. Компьютерные вирусы: что это такое и как с ними бороться / Е. В. Касперский. – М: СК Пресс, 1998. – 150 с.
12. Компьютерные сети : учебный курс. – MicrosoftPress: Русская редакция, 1998.
13. Кузнецов, М. Самоучитель MySQL 5 / М. Кузнецов, И. Симдяков. – СПб.: БХВ-Петербург, 2007 . – 560с.