

Департамент внутренней и кадровой политики Белгородской области
Областное государственное автономное
профессиональное образовательное учреждение
«Белгородский индустриальный колледж»

Рассмотрено
Цикловой комиссией
Протокол заседания №1
от «31» августа 20 20 г.
Председатель цикловой комиссии
_____ / Чобану Л.А. /

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

учебная дисциплина

ЕН.03 «Информационные базы данных»

по специальности

**11.02.10 Радиосвязь, радиовещание, телевидение
(углубленной подготовки)**

Разработчик:
Преподаватель ОГАПОУ «БИК»
Внукова Н.В.

Белгород 2020 г.

Пояснительная записка

Методические указания по выполнению лабораторных работ по дисциплине «Информационные базы данных», разработаны в помощь обучающимся для самостоятельного выполнения ими лабораторных работ, предусмотренных рабочей программой.

Обязательная аудиторная нагрузка (всего) обучающегося, предусмотренная учебным планом, составляет 72 часа, из них лабораторные занятия – 30 часов. Количество лабораторных работ и их тематика составлена в соответствии с рабочей программой по учебной дисциплине.

Лабораторные занятия проводятся преподавателем в подгруппах.

Выполнение лабораторной работы включает этапы:

1. Сбор данных и выполнение лабораторной работы;
2. Оформление отчета;
3. Защита лабораторной работы.

1. Сбор данных

Методическое пособие по лабораторным работам включает следующие разделы: № лабораторной работы, название, цель, теоретическую часть (на усмотрение преподавателя), практическую часть (порядок выполнения), контрольные вопросы.

Сбор данных выполняется в следующем порядке:

- изучается методическое пособие по выполнению работы;
- уясняется цель работы и последовательность действий;
- уточняются у преподавателя непонятные моменты;
- подготавливаются необходимые таблицы;
- выполняются действия согласно пунктам раздела «Порядок выполнения...».

Основные действия и выводы конспектируются и затем заносятся в отчет в соответствии.

2. Оформление отчета

Преподавателем разрабатывается шаблон отчета лабораторной работы (приложение 1), который заполняется вручную (строго синим стержнем) индивидуально каждым обучающимся.

Отчет по каждой работе включает следующие разделы:

1. № лабораторной работы.
2. Название работы.
3. Цель работы.
4. Отчет о выполнении заданий.
5. Перечень контрольных вопросов.
6. Ответы на контрольные вопросы.
7. Вывод по проделанной лабораторной работе.
3. Защита лабораторной работы.

Для защиты лабораторной работы обучающийся должен:

- представлять цель и порядок выполнения работы;
- изучить практический и теоретический материал согласно вопросам к защите;
- ответить на вопросы к защите и дополнительные вопросы по данной теме.

Защищенная лабораторная работа подписывается преподавателем с указанием числа защиты работы.

Лабораторная работа - это важный элемент учебного процесса. Именно на таких занятиях студенты учатся самостоятельно решать задачи и делать соответствующие выводы по их результатам, что, несомненно, будет способствовать лучшему усвоению и закреплению пройденного теоретического материала.

Тематическое планирование лабораторных работ

по дисциплине

ЕН.03 Информационные базы данных

№ п/п	Название лабораторной работы	Кол-во часов
1	Работа и особенности логических элементов ЭВМ. Построение ER-диаграммы	2
2	Создание базы данных в MS Office Access (создание таблиц и организация связей между таблицами)	2
3	Создание форм в MS Office Access	2
4	Создание простых запросов в MS Office Access	2
5	Организация выборки данных в MS Office Access	2
6	Создание отчетов в MS Office Access	2
7	Команды языка запросов SQL в MS Office Access	2
8	Использование стандартных функций для работы с конструктором запросов в MS Office Access	2
9	Создание базы данных в Visual FoxPro (создание таблиц и организация связей между таблицами)	2
10	Создание форм и простых запросов в Visual FoxPro	2
11	Создание кнопок навигации и управления данными в Visual FoxPro	2
12	Изучение стандартных функций. Изучение средств поиска данных. Использование вычисляемых полей в Visual FoxPro	2
13	Создание отчетов в Visual FoxPro	2
14	Создание базы данных в Base (создание таблиц и организация связей между таблицами)	2
15	Создание форм и отчетов в Base	2
	Всего:	30 ч.

Лабораторная работа № 1

Работа и особенности логических элементов ЭВМ. Построение ER-диаграммы

Цель работы: сформировать умения и навыки построения ER-диаграмм

Ход работы

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы, сделайте вывод о проделанной работе

Теоретическая часть

Информационная система (ИС) - программно-аппаратный комплекс для хранения и обработки информации о какой-либо предметной области.

Процесс создания ИС делится на ряд этапов. Обычно выделяют следующие этапы создания ИС:

- формирование требований к системе (анализ),
- проектирование,
- реализация,
- тестирование,
- ввод в действие,
- эксплуатация и сопровождение.

Важнейшим компонентом любой информационной системы является **база данных (БД)** – структурированный, организованный набор данных, объединенный в соответствии с некоторой выбранной моделью и описывающий характеристики какой-либо физической или виртуальной системы.

Проектирование ИС охватывает три основные области:

- проектирование объектов данных (**создание моделей данных**), которые будут реализованы в базе данных;
- проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;
- учет конкретной среды или технологии, а именно: топологии сети, конфигурации аппаратных средств, используемой архитектуры (файл-сервер или клиент-сервер), параллельной обработки, распределенной обработки данных и т. п.

Модель – искусственный объект, представляющий собой отображение (образ) системы и её компонентов.

Модель данных (Data Model) – это графическое или текстовое представление анализа, который выявляет данные, необходимые организации с целью достижения ее миссии, функций, целей, стратегий, для управления и оценки деятельности организации. **Модель данных** выявляет **сущности, домены (атрибуты)** и **связи** с другими данными, а также предоставляет концептуальное представление данных и связи между данными.

При создании моделей данных используется метод **семантического моделирования**. Семантическое моделирование основывается на значении структурных компонентов или характеристик данных, что способствует правильности их интерпретации (понимания, разъяснения). В качестве инструмента семантического моделирования используются различные варианты **диаграмм сущность-связь (ER — Entity-Relationship) — ERD**.

Существуют различные варианты отображения **ERD**, но все варианты **диаграмм сущность-связь** исходят из одной идеи — рисунок всегда нагляднее текстового описания. **ER -диаграммы** используют графическое изображение **сущностей** предметной области, их свойств (**атрибутов**), и **взаимосвязей** между сущностями.

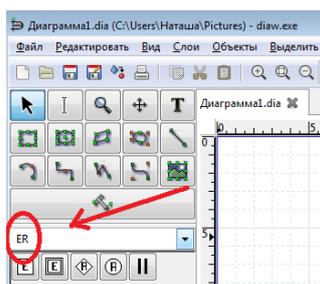
Практическая часть

Задание 1. Запустите программу *Dia* . Изучите интерфейс рабочего окна программы. Перечислите команды на панели команд:

Задание 2. Изучите Панель инструментов, запишите назначение пиктограмм

Пиктограмма	Назначение
	
	
	
	
	
	
	
	
	
	

Задание 3. Активизируйте пункт ER (как показано на рисунке)



Запишите назначение следующих пиктограмм

Пиктограмма	Назначение
	

Задание 4. Разработайте информационную систему «**Контингент студентов колледжа**». Используя возможности текстового процессора MS Word, изобразите конечный вид созданной диаграммы. Диаграмму сохраните под именем *Фамилия.docx* в папку с именем, созвучным вашей фамилии.

Главная задача системы – сбор и обработка информации об основных участниках учебного процесса: студентах и преподавателях, формирование необходимых печатных форм (документов), используемых преподавателями в период зачётной недели и экзаменационной сессии, генерация сводных отчётов по результатам сессии для работников деканатов, института. При разработке системы следует учитывать, что она основывается на документации, поступающей из приёмной комиссии, деканатов и других подразделений института. Информация об успеваемости студентов должна накапливаться и храниться в течение всего периода обучения. В системе должен использоваться справочник специальностей и дисциплин (предметов), изучаемых студентами.

Поектируемая система должна выполнять следующие действия:

1. Хранить информацию о студентах и их успеваемости.
2. На факультетах по определённой специальности печатать экзаменационные ведомости и другие документы.

Выделим все существительные в этих предложениях — это предполагаемые **сущности** и проанализируем их:

- **Студент** — явная сущность.
- **Успеваемость** — явная сущность.
- **?Факультет** — нужно выяснить один или несколько факультетов в институте? Если несколько, то это — предполагаемая новая сущность.
- **?Специальность** — нужно выяснить одна или несколько специальностей на факультете? Если несколько, то это — ещё одна сущность.
- **Предмет** — предполагаемая сущность.

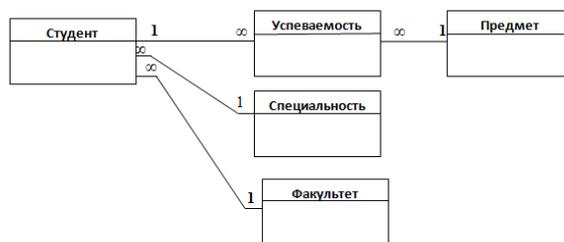
На первоначальном этапе моделирования данных информационной системы явно выделены две основные сущности: **Студент** и **Успеваемость**.

Первый вариант диаграммы выглядит так:



Мы знаем, что студенты учатся на факультетах, на определённой специальности и сдают экзамены по дисциплинам (предметам). Анализ предметной области показал, что студенты учатся на нескольких факультетах института по нескольким специальностям и сдают экзамены по определённому перечню предметов.

Исходя из этого, мы добавляем в **ER-модель** ещё несколько сущностей. В результате она будет выглядеть так:

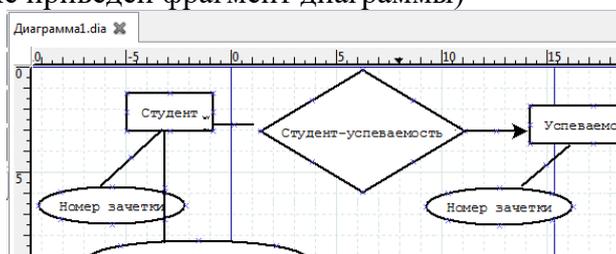


На следующей стадии проектирования модели вносим атрибуты сущностей в диаграмму (предполагаем, что атрибуты выявлены на стадии обследования объекта и при анализе аналогов существующих систем) и получаем окончательный вариант **ER— диаграммы**:



Задание 5. В программе Dia спроектируйте соответствующую ER-диаграмму в нотации Чена. Диаграмму сохраните под именем *Фамилия.dia* в папку с именем, созвучным вашей фамилии.

- Выберите пункт ER
- Выберите элемент Сущность, добавьте его на рабочее поле, дважды клинув на нем, вызовите меню Свойства, переименуйте в Студент
- Добавьте атрибуты, переименуйте соответственно
- Выполняя аналогичные действия, добавьте остальные сущности, их атрибуты, связи между ними (на рисунке приведен фрагмент диаграммы)



Контрольные вопросы

1. Что такое информационная система?
2. Перечислите этапы создания информационной системы.
3. Что помогает выявить модель данных?
4. Что выступает инструментом семантического моделирования при создании моделей данных?
5. Что выступает инструментом семантического моделирования при создании моделей данных?

Лабораторная работа № 2 Создание базы данных в MS Office Access (создание таблиц и организация связей между таблицами)

Цель работы: сформировать умения и навыки работы с системой управления базами данных (СУБД)

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

ACCESS – это реляционная СУБД. Это означает, что с ее помощью можно работать одновременно с несколькими таблицами базы данных, эти таблицы между собой связаны.

База данных состоит из следующих компонентов:

Таблицы – основные объекты базы данных (БД). В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц. В БД столбцы называются *полями*, а строки – *записями*.

Запрос – это средство, с помощью которого извлекается из базы данных информация, отвечающая определенным критериям. Результаты запроса представляют не все записи из таблицы, а только те, которые удовлетворяют запросу.

Формы – Обеспечивают более наглядную работу с таблицами, с помощью форм в базу вводят новые данные или просматривают имеющиеся.

Отчеты – средство представления данных таблиц. Отчеты могут быть оформлены надлежащим образом и распечатаны в том виде, в котором требуется пользователю.

Макросы – набор из одной или более макрокоманд, выполняющих определенные операции (открытие форм, печать отчетов)

Модули - это программы, написанные на языке программирования Visual Basic.

С таблицами можно работать в двух режимах — **таблицы** и **конструктора**.

В режиме таблицы можно начать ввод данных сразу, структура таблицы при этом будет создаваться автоматически.

На вкладке **Создание** в группе **Таблицы** щелкните **Таблица**. 

Приложение Access создаст базу данных с пустой таблицей с именем «Таблица1» и откроет эту таблицу в режиме таблицы. При этом будет выделена первая пустая ячейка столбца **Добавить поле**.

Создание таблицы в режиме конструктора

1. На вкладке **Создание** в группе **Таблицы** щелкните **Конструктор таблиц**. 
2. Для каждого поля в таблице введите имя в столбце **Имя поля**, а затем в списке **Тип данных** выберите тип данных.
3. Когда все необходимые поля будут добавлены, сохраните таблицу:
4. Сохраните базу данных.

Создание таблицы на основе шаблона

На вкладке **Создание** в группе **Таблицы** щелкните **Шаблоны таблицы** и затем выберите из списка один из доступных шаблонов.

Практическая часть

Задание 1. Откройте программу *MS Access* из пакета *MS Office*. Запишите, какие шаблоны вам доступны.

Задание 2. Вызовите команду *Файл/Создать/Доступные шаблоны/Образцы шаблонов*. Перечислите доступные вам образцы шаблонов

Задание 3. Создайте файл базы данных Анкета31РРТк.accdb.

1. Откройте программу Access.
2. Вызовите команду *Файл/Создать/Доступные шаблоны/Новая база данных.*
3. Введите имя Анкета31РРТк.
4. Появится окно <Таблица> (Рисунок 1).

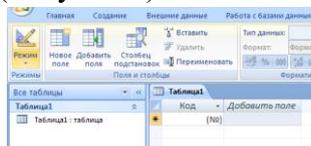


Рисунок 1

5. В появившемся окне откройте меню команды <Режим> и выберите вариант <Конструктор>  и сохраните будущую таблицу под названием <ВедомостьУспеваемости>. Появится окно Конструктора.
6. Заполните поля в **Конструкторе** данными из *таблицы 1*. Тип данных можно выбрать из меню, появившемся при нажатии на кнопку  в ячейке справа.

Обратите внимание: ключевое поле «Счетчик» внесен в таблицу автоматически. Если напротив поля отсутствует значок ключа, то на панели инструментов щелкните по этому значку  су

Таблица 1.

Имя поля	Тип данных
Код	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Математика	Числовой
Менеджмент	Числовой
Сервисная деятельность	Числовой
Информационные технологии	Числовой
Стандартизация	Числовой
Гостиничная индустрия	Числовой
Пропуски по уважительной причине	Числовой

7. Перейдите в режим таблицы, щелкнув по кнопке **Режим** на панели инструментов. Введите данные в этом режиме, заполняя клетки таблицы. Значение поля **Код** будет меняться автоматически.

8. Заполните базу данных значениями из *таблицы 2*. Напротив каждой фамилии выставьте по всем дисциплинам оценки от 2 до 5

Таблица 2

Код	Фамилия	Имя	Математика	Менеджмент	Сервисная деятельность	Информационные технологии	Стандартизация	Гостиничная индустрия	Пропуски по неуважительной причине	Пропуски по уважительной причине
1										
2										
3										
...

9. Отсортируйте:

- а) *фамилии* – по алфавиту (поставьте маркер на любую фамилию в столбце Фамилия и щелкните мышкой по кнопке  на панели инструментов или произведите сортировку с помощью контекстного меню)
- б) *имя* – по алфавиту

10. Сохраните текущую таблицу, щелкнув по кнопке «крестик» в правом верхнем углу окна таблицы.

Задание 4. Выполните поиск в базе данных Анкета31РРТк.accdb.

1. Выполните поиск записей по образцу: *найти студентку по фамилии Внукова*. Для этого установите курсор в поле фамилия, щелкните на кнопке  <Бинокль> на панели инструментов меню **Главная** и в появившемся диалоговом окне введите в поле <Образец> фамилию *Внукова* и щелкните по кнопке <Найти>.

Примечание: Если требуется найти следующую подобную запись, то щелкните мышкой по кнопке <Найти далее>. По окончании работы щелкните по кнопке <Отмена>.

2. Скройте столбец **Пропуски по неуважительной причине**, потом отобразите его назад.

3. Войдите в режим *Конструктора* и назначьте полю **Пропуски по неуважительной причине** и **Пропуски по уважительной причине**. *Маску ввода*

Для этого необходимо перейти к свойству поля, найти строку *Маска ввода* и ввести **00 «часов»**.

Заполните эти поля данными от 0 до 99.

Задание 5. Добавьте таблицу *Преподаватели* в файл базы данных Анкета31РРТк.accdb.

1. Создайте таблицу <Преподаватели> в *Режиме таблицы*. Для этого в меню Создание выберите кнопку **Таблица**. В появившейся таблице сделайте следующее:

- Добавьте два поля – Поле 1 и Поле 2, выполнив команду через контекстное меню (правая кнопка мыши).

- Переименуйте <Поле 1> на <Предмет>. Для этого поставьте курсор в любую ячейку столбца <Поля 1> и выполните команду *Переименовать столбец* из контекстного меню. Или щелкните два раза по имени поля, удалите старое название и введите новое.

- Переименуйте аналогично <Поле 2> на <Преподаватель>.

2. Сохраните таблицу с именем <Преподаватели>, щелкнув по кнопке <Сохранить>.

3. Перейдите в режим <Конструктор> и удалите строку с ключевым словом Счетчик. Посмотрите, как заданы поля. Сделайте поле <Предмет> ключевым, поместив курсор на имя этого поля и щелкнув  о кнопке - *Ключевое поле*. Тип данных поля задайте *текстовым*.

4. Перейдите в *Режим таблицы* и заполните таблицу <Преподаватели> записями из *Таблицы3*.

Таблица 3

Предмет	Преподаватели
Гостиничная индустрия	Проскурин П.О.
Информационные технологии	Внукова Н.В.
Математика	Бабич О.Р.
Менеджмент	Иванова К.Р.
Сервисная деятельность	Маркова П.Р.
Стандартизация	Бекетова Н.И.

5. Закройте таблицу <Преподаватели>, сохранив все изменения.

Задание 6. Добавьте таблицу *Личные данные* в файл базы данных Анкета31РРТк.accdb.

1. Используя <Шаблон таблиц>, создайте таблицу <Личные данные> студентов с ключевым полем. Для этого:

- Находясь на закладке <Создание> щелкните по кнопке <Шаблоны таблиц>, <Контакты>. Появится таблица уже с готовыми полями.

- Переименуйте предложенные поля на следующие поля: <Код студента>, <Фамилия>, <Имя>, <Город>, <Адрес>, <Телефон>, <Дата рождения>, <Фотография>, <Любимый предмет>, лишние поля удалите.

- Сохраните полученную таблицу под названием <Личные данные>. Ключевое поле задано автоматически.

2. Внесите данные в новую таблицу, заполнив поля <Фамилия>, <Имя>, <Город>, <Адрес>, <Телефон>, <Дата рождения>.

ПРИМЕЧАНИЕ. Поля <Фамилия> и <Имя> можно скопировать из таблицы <ВедомостьУспеваемости>. В поле <Город> внесите четыре разных города (например, Белгород, Алексеевка, Короча, Валуйки)

3. Перейдите в режим <Конструктор> и назначьте типы данных: для поля <Телефон> - *числовой*, для поля <Дата рождения> - *дата/время*, для поля <Фотография> – *поле объекта OLE*, для остальных – *текстовый*.

Для поля <Любимый предмет> выполните свойство выбор предмета из списка с помощью *Мастера подстановок*. Для этого в строке <Любимый предмет> в поле *Тип данных – текстовый* щелкните по кнопке  и в ниспадающем меню выберите команду <Мастер подстановок>.

- В диалоговом окне <Создание подстановки> поставьте флажок напротив способа <Будет введен фиксированный набор значений> и нажмите <Далее>.
- В следующем окне внесите в столбец все предметы (предметы из таблицы <Преподаватели>), нажмите <Далее>.
- В последнем окне, не изменяя имени столбца нажмите <Готово>.

4. Перейдите в режим таблицы и выберите для каждого студента с помощью пки из списка любимый предмет.

Задание 7. Создайте схему данных базы данных Анкета31РРТк.accdb.

Создание *схемы данных* - это установление связи между таблицами.

1. Щелкните по кнопке  - *Схема данных* на панели инструментов меню <Работа с базами данных>. В окне <Отобразить таблицу> выделите таблицу <Ведомость успеваемости> и щелкните по кнопке <Добавить>. Также добавьте таблицы <Преподаватели> и <Личные данные>. В окне <Схема данных> появиться условный вид этих таблиц. Закройте окно <Добавление таблицы>.

2. Произведите фильтрацию данных в таблице <Личные данные> *по выделенному*.

3. Закончите работу с базой данных Access.

Контрольные вопросы

1. Каково назначение программы MS Access?
2. Что является основным объектом базы данных?
3. Что такое поля таблицы базы данных?

Лабораторная работа № 3

Создание форм в MS Office Access

Цель работы: сформировать умения и навыки работы с системой управления базами данных (СУБД)

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

Реляционная база данных — это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа.

Таблицы – основные объекты базы данных (БД). В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц. В БД столбцы называются *полями*, а строки – *записями*.

Связи между таблицами устанавливаются на *Схеме данных*. Схема данных строится в соответствии с информационно-логической моделью данных. При построении схемы данных Access автоматически выбирает тип связи по выбранному полю таблиц.

Одна из связанных таблиц является *главной* (базовой), вторая — *подчиненной*. Access позволяет установить связи следующих типов:

связь один-к-одному, при которой одной записи из главной таблицы соответствует одна запись из подчиненной таблицы;

связь один-ко-многим, при которой одной записи из главной таблицы соответствует несколько записей из подчиненной таблицы.

В окне *Связи* можно задать параметры для устанавливаемой связи:

- обеспечение целостности связи;
- каскадное обновление полей;
- каскадное удаление связанных полей.

Целостность данных означает систему правил, которые используются для поддержания связей между записями в связанных таблицах, а также обеспечивают защиту связанных данных от случайного удаления или изменения.

Практическая часть

Задание 1. Откройте программу *MS Access* из пакета *MS Office*. Запишите путь доступа к этой программе.

Задание 2. Создайте базу данных *Заказы*. Запишите последовательность своих действий для создания новой базы данных.

Задание 3. Нажмите на пиктограмму *Режим*, запишите, какие режимы доступны.

Задание 4. Нажмите на пиктограмму *Создание* на *Ленте*, заполните таблицу.

Вкладка	Команды
Таблицы	
Запросы	
Формы	
Отчеты	

Задание 5. Создайте файл базы данных *Заказы.accdb*.

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

1. Откройте программу Access, вызовите команду *Файл/Создать/Доступные шаблоны/Новая база данных*.

2. Введите имя Заказы.

3. Создайте таблицу базы данных ТОВАРЫ с информацией о товарах. Для этого в окне **База данных** выберите выбрать объект **Таблицы**. Выбрать пункт **Создание таблицы в режиме конструктора**.

4. В окне конструктора указать имя поля и тип данных для каждого из полей таблицы. Имена и характеристики полей указаны ниже в таблице.

Описание структуры таблицы *Товары*

Таблица 1

Имя поля	Тип данных	Размер поля	
Код товара	Числовой	Целое	Ключевое поле
Наименование товара	Текстовый	50	
Вид товара	Текстовый	50	

5. Определить первичный ключ таблицы (Код товара). Для этого выделите строчку поля Код товара и нажмите пиктограмму «Ключевое поле» на панели инструментов.

6. Закройте окно Конструктора и сохраните таблицу под именем ТОВАРЫ.

7. Создайте другие таблицы базы данных аналогичным образом. Имена и типы полей таблицы введите, как представлено в таблицах.

Описание структуры таблицы *Поставщики*

Таблица 2

Имя поля	Тип данных	Размер поля	
Код поставщика	Числовой	Целое	Ключевое поле
Наименование поставщика	Текстовый	50	
Адрес	Текстовый	50	
Телефон	Текстовый	6	

Описание структуры таблицы *Поставки*

Таблица 3

Имя поля	Тип данных	Размер поля	
Код поставки	Числовой	Целое	Ключевое поле
Дата поставки	Дата/время		Краткий формат даты
Код товара	Числовой	Целое	
Код поставщика	Числовой	Целое	
Количество товара	Числовой	Целое	
Цена за единицу	Денежный		

Описание структуры таблицы *Продажи*

Таблица 4

Имя поля	Тип данных	Размер поля	
Код продажи	Числовой	Целое	Ключевое поле
Дата продажи	Дата/время		Краткий формат даты
Код товара	Числовой	Целое	
Количество товара	Числовой	Целое	
Цена за единицу	Денежный		

Задание 6. Уставите связи между таблицами базы данных

8. Установите связи между таблицами по полям: **Код товара** (для таблиц ТОВАРЫ и ПОСТАВКИ), **Код товара** (для таблиц ТОВАРЫ и ПРОДАЖИ), **Код поставщика** (для таблиц ПОСТАВЩИКИ и ПОСТАВКИ). Для этого выполните команду меню **Сервис – Схема данных**.

9. Добавить таблицы, участвующие в связях и закрыть окно **Добавить**.

10. Отбуксировать поле **Код товара** таблицы ТОВАРЫ на соответствующее поле **Код товара** таблицы ПОСТАВКИ.

В окне **Изменение связей** активизировать флажки **Обеспечение целостности данных, каскадное обновление связанных полей и каскадное удаление связанных записей**. Тип отношения должен быть **один-ко-многим**. Нажать кнопку **Создать**.

11. Аналогичным образом создайте связи между таблицами ТОВАРЫ и ПРОДАЖИ по полю **Код товара** (отбуксировать поле **Код товара** с таблицы ТОВАРЫ на таблицу ПРОДАЖИ и в окне **Изменение связей** активизировать флажки **Обеспечение целостности данных, каскадное обновление связанных полей и каскадное удаление связанных записей**, тип отношения **один-ко-многим**) и между таблицами ПОСТАВЩИКИ и ПОСТАВКИ по полю **Код поставщика** (отбуксировать поле **Код поставщика** с таблицы ПОСТАВЩИКИ на таблицу ПОСТАВКИ и в окне **Изменение связей** активизировать флажки **Обеспечение целостности данных, каскадное обновление связанных полей и каскадное удаление связанных записей**, тип отношения **один-ко-многим**).

12. Связи представлены в виде линий между связующими полями таблиц.

13. Закройте окно **Схема данных** и сохраните макет схемы данных.

Задание 7. Создайте форму для ввода данных в таблицы базы данных

14. Создайте форму для ввода данных в таблицу ТОВАРЫ как описано ниже.

15. Выбрать объект **Формы**

16. Выбрать пункт **Создание формы с помощью мастера**

17. Выбрать в качестве источника данных таблицу ТОВАРЫ.

18. Из списка доступных полей выбрать нужные поля (все поля таблицы – кнопка >>).

19. Выбрать внешний вид форм из предложенных (**ленточный**).

20. Выбрать стиль оформления.

21. Задайте имя формы ФОРМА ТОВАРЫ.

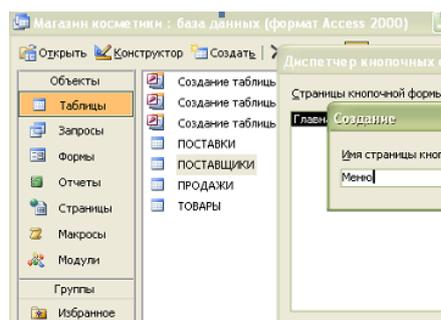
22. Создайте форму для ввода данных в таблицу ПРОДАЖИ аналогичным образом и задайте ей имя ФОРМА ПРОДАЖИ.

23. Создайте форму для ввода данных в таблицу ПОСТАВЩИКИ аналогичным образом и задайте ей имя ФОРМА ПОСТАВЩИКИ.

24. Создайте форму для ввода данных в таблицу ПОСТАВКИ аналогичным образом и задайте ей имя ФОРМА ПОСТАВКИ.

25. Откройте форму ФОРМА ТОВАРЫ (Выберите объект **Формы** в окне базы данных и откройте форму ФОРМА ТОВАРЫ.

26. Заполните данными соответствующие поля формы ФОРМА ТОВАРЫ как показано на рисунке.



27. Дополните таблицу данными до 6 записей.

28. Введите данные в таблицы, используя созданные формы (4-6 записей). Значения для одноименных полей в разных таблицах должны совпадать!

29. Предъявите результаты преподавателю:

открытые таблицы в режиме конструктора (кнопка на панели инструментов)

открытые формы

30. Закончите работу с базой данных Access.

Контрольные вопросы

1. Каково назначение программы MS Access?
2. Как создать структуру таблицы?
3. Что такое схема данных?

Лабораторная работа № 4

Создание простых запросов в MS Office Access

Цель работы: сформировать умения и навыки работы с системой управления базами данных (СУБД)

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

Запросы обеспечивают быстрый и эффективный доступ к данным, хранящимся в таблице. По функциональному признаку запросы можно разделить на:

1. запросы выбора (результат существует в виде таблицы до закрытия запроса)

- простые,
- запросы с критериями поиска,
- с вычислениями,
- с параметрами,
- итоговые,
- многотабличные,
- перекрестные.

2. запросы действия (приводят к изменениям данных в исходных таблицах или созданию новых таблиц, сохраненных как объект таблицы)

- на создание таблицы,
- на обновление,
- на добавление,
- на удаление.

Практическая часть

Задание 1. Создайте в MS Access новую базу данных ИС_Учет_Успеваемости, содержащую три таблицы:

1. Таблица «Кадры», содержит 7 полей, наименование полей и их тип:

КодСтудента – Счетчик (ключевое поле);

Фамилия, Имя, Отчество, Гражданство – Текстовые;

ДатаРождения – Дата/время;

НомУчГруппы – Текстовый.

2. Таблица «УчебныйПлан», содержит 4 поля, наименование полей и их тип:

КодДисциплины – Счетчик (ключевое поле);

Наименование, Преподаватель – Текстовые;

КолЧасов – Числовой.

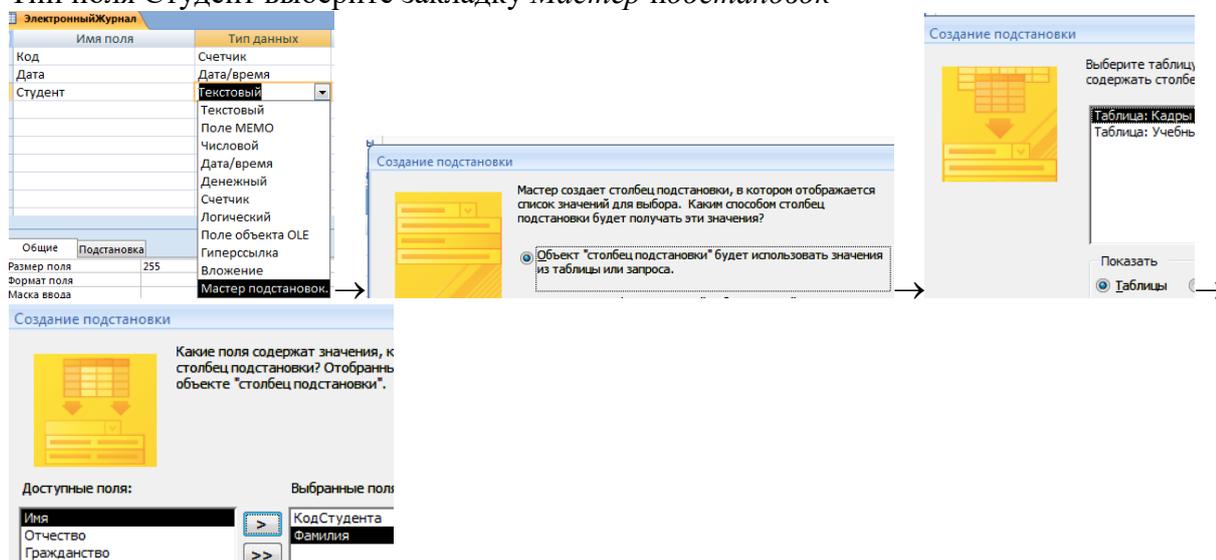
3. Таблица «ЭлектронныйЖурнал», содержит 6 полей, наименование полей и их тип:

Код – Счетчик (ключевое поле);

Дата – Дата/время;

Студент – смешанный тип данных (содержит поля КодСтудента и Фамилия из таблицы Кадры, создается с помощью *Мастера подстановок*);

Тип поля Студент выберите закладку *Мастер подстановок*



Дисциплина – смешанный тип данных (содержит поля КодДисциплин, Наименование из таблицы Учебный план, аналогично создается с помощью *Мастера подстановок*);

ВидЗанятия – Текстовый;

Оценка – Числовой.

Задание 2. Откройте таблицу *Кадры* в режиме Конструктор. Перечислите, какие типы данных предлагаются для каждого поля при создании таблицы с помощью Конструктора.

Задание 3. Настройте свойства полей созданных таблиц.

При создании нового поля в зависимости от его типа некоторые свойства имеют значения по умолчанию.

3.1. Открыть таблицу *Кадры* в режиме Конструктор (для этого в контекстном меню выбрать пункт Конструктор)

– Установить курсор на поле *КодСтудента*, на вкладке *Общие* (Свойства поля) установить свойство *Подпись* – № п/п.

– Для полей *Фамилия*, *Имя* и *Отчество* установить свойства: *Размер поля* – 255; *Обязательное поле* – Да; *Индексированное поле* – Да (Допускаются совпадения)

– Для поля *ДатаРождения* установить (выбрать) свойства: *Формат поля* (справа в строке будет символ) – *Краткий формат даты* (19.06.2007); *Маска ввода* – (справа в строке будет символ запустить мастер), подтвердить сохранение таблицы, нажать кн. *Да*, в окне *Создание масок ввода* выбрать *Краткий формат даты* (27.09.1969), нажать кн. *Далее>*, в следующем окне снова нажать кн. *Далее>*, затем кн. *Готово*. Свойство *Маска ввода* должна установиться – 00.00.0000;0_ ; *Подпись* – *Дата рождения*; *Обязательное поле* – Да; *Индексированное поле* – Да (Допускаются совпадения)

Для поля *НомУчГруппы* установить свойства: *Подпись* – *Уч. группа*; *Обязательное поле* – Да; *Индексированное поле* – Да (Допускаются совпадения), *Значение по умолчанию* – номер своей учебной группы.

Для поля Гражданство: Размер поля – 255; Значение по умолчанию – Российская_Федерация; Обязательное поле – Да; Индексированное поле – Да (Допускаются совпадения). Нажать кн. Сохранить, закрыть таблицу Кадры.

3.2. Открыть для редактирования таблицу Учебный план.

- Установить свойства полей:
- поле КодДисциплин – свойство Подпись – № п/п; поле Наименование – Размер поля – 255, Подпись – Дисциплина, Обязательное поле – Да, Индексированное поле – Да (Совпадения не допускаются);
- поле Преподаватель – Обязательное поле – Да, Индексированное поле – Да (Допускаются совпадения); поле КолЧасов – Подпись – Часы.
- Сохранить и закрыть таблицу.

3.3. Установка фиксированного набора значений. Открыть таблицу Кадры в режиме Конструктор.

- Для поля Гражданство, открыть список Тип данных и выбрать Мастер подстановок... .
- В окне Создание подстановки выбрать режим – Будет введен фиксированный набор значений, нажать кн. Далее. В следующем окне в Столбец 1 последовательно в строки столбца ввести гражданство: Российская_Федерация, Республика_Беларусь, Украина и т.д.
- Нажать кн. Далее, затем Готово.
- Сохранить и закрыть таблицу Кадры.

3.4. Откройте в режиме *Конструктора* таблицу Электронный Журнал.

- Имя поля – Код, Тип данных – Счетчик, Свойства: Подпись – № п/п. Сделать поле ключевым.
- Имя поля – Дата, Тип данных – Дата/время, Свойства: Формат поля – Краткий формат даты; Маска ввода – (используя мастер масок ввода) 00.00.0000;0;_ ; Значение по умолчанию – запустить мастер Функции , в окне Построитель выражений открыть папку Функции, затем Встроенные функции, выбрать тип функции Дата/время, затем функцию – Date, нажать кн. Вставить, затем ОК, в поле свойства “Значение по умолчанию” появится – Date() (эта функция генерирует текущую дату); Обязательное поле – Да, Индексированное поле – Да (Допускаются совпадения).
- Имя поля – Студент, Тип данных – выбрать Мастер подстановок ... , в окне Создание подстановки установить – столбец подстановки использует значения из таблицы или запроса, нажать кн. Далее>, выбрать таблицу Кадры, нажать кн. Далее>, из поля Доступные поля: в поле Выбранные поля: используя кн. переместить поля КодСтудента, Фамилия, нажать кн. Далее>, убедиться что в следующем окне установлен переключатель Скрыть ключевой столбец, нажать кн. Далее>, затем Готово, выполнить рекомендации по сохранению таблицы.
- Имя поля – Дисциплина, Тип данных – выбрать Мастер подстановок ... аналогично предыдущему полю связать с таблицей Учебный план (поля КодДисциплин, Наименование).
- Имя поля – ВидЗанятия, Тип данных – выбрать Мастер подстановок ... используя режим – фиксированный набор значений ввести следующие виды занятий: Лекция, Лаб_Зан, Практ_Зан, Семинар. Свойства: Подпись – Вид занятия; Значение по умолчанию – Практ_Зан. Имя поля – Оценка, Тип данных – выбрать Мастер подстановок ... используя режим – фиксированный набор значений ввести следующие виды оценок: 5, 4, 3, 2, установить Тип данных – Числовой. В свойствах полей Студент (КодСтудента), Дисциплина (КодДисциплин), Оценка убрать 0 из Значение по умолчанию, Обязательное поле – Да, Индексированное поле – Да (Допускаются совпадения).

После произведенных изменений сохраните таблицу

Задание 4. Отредактируйте схему данных.

Закрывать все таблицы. На ленточной панели инструментов выбрать вкладку Работа с базами данных на ленте инструментов В группе Показать или скрыть щелкните кнопку Схема данных.

В окне Схема данных должны располагаться таблицы со связями (соединениями одноименных полей в связанных таблицах). Выполнить коррекцию связей, для чего, установить указатель мыши на линию связи и дважды щелкнуть левой кнопкой мыши.

В окне Связи установить переключатели:

Обеспечение целостности данных;
каскадное обновление связанных полей;
каскадное удаление связанных записей, нажать ОК.

Аналогично настроить вторую связь.

Убедитесь, что тип отношений между базовыми таблицами и таблицей данных Один ко многим. Сохранить и закрыть схему данных.

Задание 5. Заполните таблицу.

Задание 6. Измените структуру таблиц базы данных.

Открыть таблицу Кадры в режиме Конструктор.

Установить курсор на поле Фамилия.

На вкладке Конструктор в разделе Сервис нажать кн. Добавить строки (или выполнить команду Вставка).

Добавить поле Пол с текстовым типом данных.

Сохранить таблицу.

Перейти в Режим таблицы.

Заполнить столбец Пол (мужской/женский).

Закрывать таблицу Кадры

Задание 7. Сформируйте простой запрос как расширенный фильтр

открыть таблицу Электронный журнал,

выполнить команду Главная/Сортировка и фильтр/Дополнительно/Расширенный фильтр.

В строке Поле: из раскрывающегося списка выбрать в отдельные столбцы поля: Код, Дата, Студент, Дисциплина, ВидЗанятия, Оценка.

В строке Сортировка: для поля Дата установить – по возрастанию.

Условие отбора: установить: для поля Оценка – 2, ВидЗанятия – Практ_Зан.

В строке «или:» для поля ВидЗанятия установить – Лаб_Зан, для поля Оценка – 2.

На панели быстрого пуска нажать кн. Сохранить как запрос.

В окне Сохранение в виде запроса ввести имя – Запрос по неудовлетворительным оценкам.

Закрывать окно фильтра.

Перейти на вкладку Запросы.

Выполнить запрос (кн. Открыть) Запрос по неудовлетворительным оценкам.

Проверить правильность выборки данных из таблицы Электронный журнал.

Самостоятельно сформируйте запросы по следующим критериям: 1) отличные оценки;

2) оценки 4 или 5.

Контрольные вопросы

1. Каково назначение запросов MS Access?
2. Опишите запрос с критерием поиска
3. Опишите запросы с параметрами

Лабораторная работа № 5

Организация выборки данных в MS Office Access

Цель работы: сформировать умения и навыки работы с системой управления базами данных (СУБД)

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

Запросы обеспечивают быстрый и эффективный доступ к данным, хранящимся в таблице. При создании таблицы с помощью MS Access можно заметить, что программа не разрешает использовать вычисляемые поля.

При работе с базами данных наиболее часто используются **простые запросы** – выбор записей из таблиц или других запросов удовлетворяющих заданному критерию.

Перекрестный запрос – запрос, результатом выполнения которого является таблица, столбцами и строками которой являются записи, находящиеся в заданных полях других таблиц или других запросов.

В Microsoft Access используются четыре основных способа создания нового запроса:

- создание запроса на основе существующего фильтра;
- создание запроса с использованием Мастера запросов;
- создание запроса с помощью Конструктора запросов;
- создание запроса в режиме SQL-редактора.

Практическая часть

Продолжаем работать с базой данных ИС_Учет_Успеваемости, созданной при выполнении лабораторной работы №4

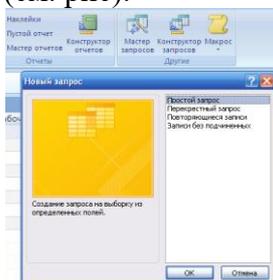
Формирование простого запроса с помощью мастера

Задание 1. Откройте программу *MS Access* из пакета *MS Office*. Откройте базу данных **ИС_Учет_Успеваемости** (или создайте ее по указаниям из лабораторной работы №4). Откройте на *Ленте* вкладку *Создание*, перечислите *способы создания* запросов.

Задание 2. Вызовите команду *Мастер запросов*, запишите предлагаемые типы новых запросов.

Задание 3. Создайте простой запрос с помощью Мастера запросов

1. Щелкните вкладку **Создание**.
2. В группе **Запросы** нажмите кнопку **Мастер запросов**. На экране появится первое диалоговое окно **Мастера запросов** (см. рис).



3. Убедитесь, что в окне Мастера отмечен вариант **Простой запрос**. Нажмите кнопку **ОК**. На экране появится следующее окно Мастера.

4. Раскройте список **Таблицы и запросы** и выберите источник данных для создания нового запроса. В поле **Таблицы/запросы:** выбрать **Таблица: ЭлектронныйЖурнал**. Из поля **Доступные поля:** переместить в поле **Выбранные поля:** все поля данных;

5. Снова открыть поле **Таблицы/запросы:** и выбрать **Таблица: Кадры**, из поля **Доступные поля:** переместить в поле **Выбранные поля:** – *Пол, НомУчГруппы*. Нажать кн. **Далее>**. Выбрать вид отчета – **подробный, Далее>**;

6. Задать имя запроса – *Учет успеваемости за (дата)*. Нажать кн. **Готово**;

7. Перейдите в режим **Конструктор**. Для поля *Дат* в строку **Условие отбора:** ввести [*За какое число?*](вводится в квадратных скобках). Сохранить запрос;

8. Выберите **Режим таблицы**. В окне **Введите значение параметра** ввести одну из дат, содержащихся в таблице, нажать кн. **ОК**. Закрывать окно запроса

9. Проверить работу запроса по другим датам.

Задание 4. Аналогично заданию 3, самостоятельно сформируйте запрос по датам, больше заданной даты.

Задание 5. Аналогично заданию 3, самостоятельно сформируйте запрос по датам, лежащим в диапазоне между заданными датами.

Задание 6. Аналогично заданию 3, самостоятельно сформируйте запрос по оценкам больше заданной оценки.

Задание 7. Сформируйте с помощью перекрестного запроса таблицу, в которой названиями строк являются записи, содержащиеся в полях *Фамилия, Имя* и *ВидЗанятия*, а названиями столбцов – записи, содержащиеся в поле *Наименования(дисциплин)*, также имеется еще один столбец – “*средний балл*”. Сама же таблица должна содержать средние значения записей поля *Оценка*, то есть, средний балл соответствующий данному студенту, виду занятий и дисциплине.

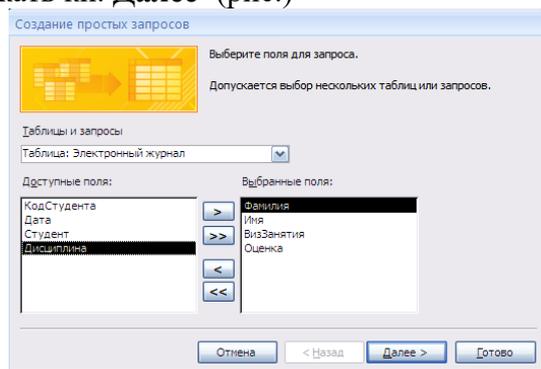
Подготовку данных для создания перекрестного запроса осуществим с помощью простого запроса, для чего выполните следующие действия:

1. Щелкните вкладку **Создание**.

2. В группе **Другие** нажмите кнопку **Мастер запросов**. На экране появится первое диалоговое окно **Мастера запросов** (см. рис.).

3. Убедитесь, что в окне Мастера отмечен вариант **Простой запрос**. Нажмите кнопку **ОК**. На экране появится следующее окно **Мастера**.

4. В окне **Создание простых запросов** в поле **Таблицы/запросы:** выбрать **Таблица: Кадры**, из поля **Доступные поля:** переместить в поле **Выбранные поля:** *Фамилия, Имя; Таблица: Учебный план –поле Наименование; Таблица: Электронный журнал –поля ВидЗанятия и Оценка*. Нажать кн. **Далее>**(рис.)



Выбрать вид отчета –*подробный, Далее>*.

Задать имя запроса –*Подготовка данных*.

Нажать кн. **Готово**;

6. На вкладке **Запросы** окна **База данных**, нажать на кн. **Создать**. В окне **Новый запрос** выбрать **Перекрестный запрос**. Нажать кн. **ОК**. В окне **Создание перекрестных таблиц** выбрать **Показать Запросы** и **Запрос: Подготовка данных Далее>**.

7. В следующем окне предлагается выбрать поля для **заголовков строк**. Выбрать в качестве полей заголовка строк –*Фамилия, Имя* и *ВидЗанятия*. Нажать кн. **Далее>**. В

качестве заголовка столбцов в следующем окне выбрать поле –*Наименование*. Нажать кн. **Далее**>.

8. В следующем окне в качестве поля вычислений выбрать *Оценка*, **Функции**: – **Максимум**. –**Да**. Нажать кн. **Далее**>. Задать имя запроса – *Таблица успеваемости*, нажать кн и изменить структуру запроса **Готово**.

Отредактируйте запрос

Просмотрите содержимое полученной перекрестной таблицы.

Сохраните и закройте запрос.

Задание 8. Сформируйте с помощью перекрестного запроса таблицу, в которой названиями строк являются записи, содержащиеся в полях *Фамилия*, *Имя*, а названиями столбцов – в поле *Дата*. Сама же таблица должна содержать записи поля *Оценка*, относящиеся к заданным *НомУчГруппы*, *Дисциплине* и *ВидуЗанятия* (для определенности, например, выберем группу 31, *математика* и *практика*).

Контрольные вопросы

1. Каково назначение запросов MS Access?
2. Опишите перекрестный запрос
3. Опишите способы создания запросов в MS Access

Лабораторная работа №6

Создание отчетов в MS Office Access

Цель работы: сформировать умения и навыки работы с системой управления базами данных (СУБД)

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

В отличие от других объектов базы данных, с помощью отчета невозможно изменять сведения, хранящиеся в базе данных.

С помощью отчета можно только *отображать* данные, содержащиеся в таблицах или выбранные из базы данных по запросу. Зато отчет лучше подходит для представления требуемых пользователю данных в виде полноценных документов, не уступающих подготовленным в Microsoft Word, для которых можно выбрать требуемый стиль оформления, а затем напечатать на принтере или отправить по электронной почте.

Практическая часть

Продолжаем работать с базой данных ИС_Учет_Успеваемости, созданной при выполнении лабораторной работы №4

Автоматическое создание простого отчета

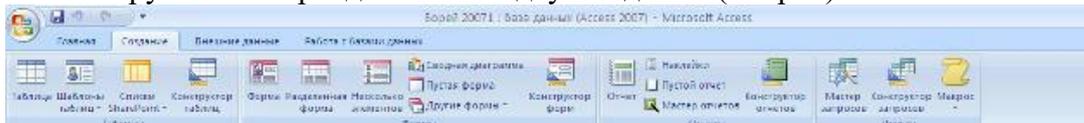
Задание 1. Откройте программу *MS Access* из пакета *MS Office*. Откройте базу данных **ИС_Учет_Успеваемости** (или создайте ее по указаниям из лабораторной работы №4). Откройте на *Ленте* вкладку *Создание*, перечислите *способы создания отчетов*.

Пиктограмма	Назначение
 Отчет 1	
 Конструктор отчетов	

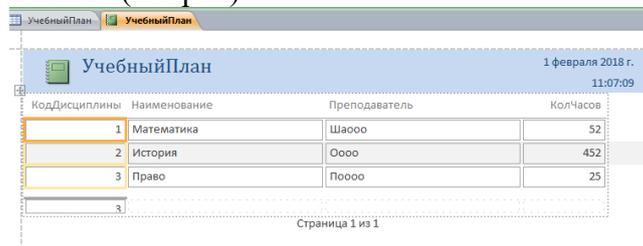
 Пустой отчет	
 Мастер отчетов	
 Наклейки	

Задание 2. Создайте автоматически простой отчет на основе таблицы *Учебный план* базы данных *ИС_Учет_Успеваемости*.

1. Откройте требуемую *ИС_учета_успеваемости*.
2. В области переходов щелкните название таблицы *УчебныйПлан*, которая будет использоваться в качестве источника данных для нового отчета.
3. На ленте инструментов перейдите на вкладку **Создание** (см. рис).

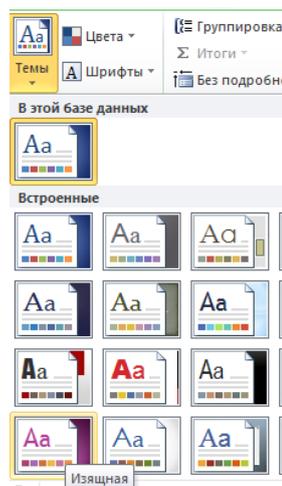


5. В группе **Отчеты** щелкните кнопку **Отчет**. Приложение Access создаст новый отчет и отобразит его в режиме макета (см. рис).



КодДисциплины	Наименование	Преподаватель	КолЧасов
1	Математика	Шаооо	52
2	История	Ооооо	452
3	Право	Пооооо	25

Настройте цветовую схему для отчета, используя Конструктор/Темы/Понравившаяся тема (см.рис.16).



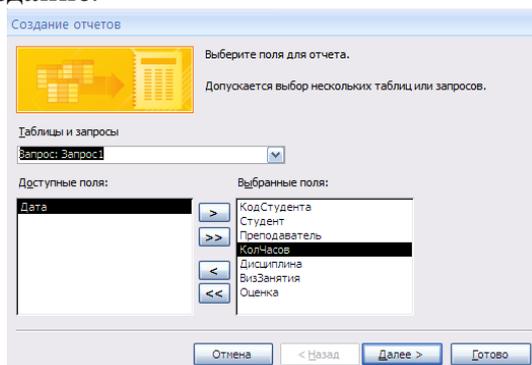
5. Нажмите кнопку **Сохранить** на панели быстрого доступа, чтобы сохранить новый отчет в текущей базе данных. На экране появится окно запроса имени нового отчета.
6. Введите желаемое название отчета (созвучное вашей фамилии) в поле *Имя отчета*.
7. Нажмите кнопку **ОК**. Новый отчет будет сохранен в текущей базе данных, а его название появится в области переходов.

Создание с помощью Мастера

Задание 3. Создайте новый отчет по таблице или запросу с помощью Мастера и выполните следующие действия:

1. Откройте требуемую *ИС_учета_успеваемости*.
2. В области переходов щелкните название таблицу *Электронный журнал и запрос_1* (который вы создали в предыдущих лабораторных работах).

3. Щелкните вкладку **Создание**.



5. В списке **Доступные поля** отметьте поля таблицы и запроса, которые должны быть включены в новый отчет. Если вы хотите использовать весь набор полей исходной таблицы, щелкните кнопку . Чтобы выбрать только часть полей таблицы, щелкните каждое требуемое поле и нажмите кнопку  для его переноса в список **Выбранные поля** и щелкните *Далее*.

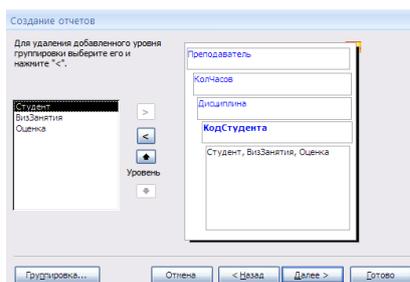
6. В следующем окне мастера выберите вид представления по своему усмотрению и щелкните *Далее*.

7. В следующем окне Мастера добавить уровни группировки. Выберите один из видов представления данных, предложенных в окне **Мастера**. Для просмотра образца представления данных щелкните имя представления в списке. Выбранный образец будет показан в правой части окна Мастера. Выбирайте уровни группировки отчета следующим образом:

- чтобы добавить уровень группировки по полю, щелкните название поля в списке доступных полей и нажмите кнопку ;

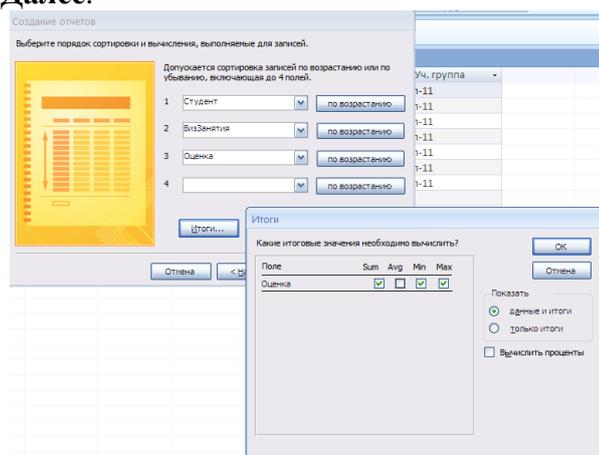
- чтобы изменить порядок уровней группировки, используйте кнопку  для перемещения уровня группировки на одну позицию вверх или кнопку  для перемещения на одну позицию вниз;

- чтобы удалить уровень группировки по полю, щелкните название уровня на образце представления данных отчета и нажмите кнопку . Нажмите кнопку *Далее*. На экране появится третье окно Мастера.



8. На экране появится четвертое окно Мастера, в котором необходимо установить порядок сортировки данных отчета. В окне Мастера вы можете выбрать до четырех полей сортировки, установив сортировку по возрастанию или по убыванию в каждом из отмеченных полей. Чтобы установить сортировку данных по какому-либо полю, раскройте список и щелкните название поля. По умолчанию данные в поле сортируются по возрастанию, для выбора сортировки по убыванию щелкните кнопку **По возрастанию**, расположенную справа от списка (при этом ее название изменится на По убыванию).

Нажмите кнопку **Итоги**, в открывшемся диалоговом окне поставить галочки Сумма, минимум, максимум, проверить что бы была активирована радиокнопка **Данные и итоги** (см.рис). Нажмите кнопку **Далее**.



9. Появится окно **Вид макета**. В группе Макет выберите вид макета отчета, отметив один из предложенных вариантов:

- **Ступенчатый** - при установке этого переключателя новый отчет будет оформлен по макету «Ступенчатый»;
- **Блок** - при установке этого переключателя новый отчет будет оформлен по макету «Блок»;
- **Структура** - при установке этого переключателя новый отчет будет оформлен по макету «Структура».

Отставьте макет **Ступенчатый**, предложенный по умолчанию и ориентация - книжная.

10. На экране появится шестое окно **Мастера** выбор стиля отчета В предложенном списке выберите стиль оформления нового отчета **Справедливость**. Нажмите кнопку **Далее**.

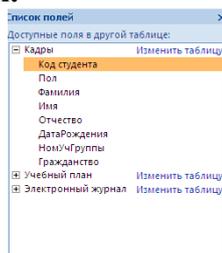
11. На экране появится последнее окно Мастера

Введите название нового отчета **Комбинированный** в верхнее поле ввода. В группе **Дальнейшие действия** оставьте вариант **Просмотреть отчет**, выбранный по умолчанию. В этом случае сразу после завершения работы Мастера можно будет просмотреть результирующий вид созданного отчета.

Нажмите кнопку **Готово**. Новый отчет будет сохранен в текущей базе данных. В рабочей области Access 2007 откроется окно предварительного просмотра созданного отчета.

Задание 4. Создайте пустой отчет для чего, выполните следующие действия:

1. Откройте требуемую **ИС_учета_успеваемости**.
2. В группе **Отчеты** щелкните кнопку **Пустой отчет**. Приложение Access создаст пустой отчет и отобразит его в режиме макета
3. В области **Список полей** щелкните знак плюс (+) рядом с таблицей, содержащей поля, которые нужно включить в отчет. Если в отчете должны выводиться данные из полей нескольких таблиц, раскройте списки полей всех нужных вам таблиц. Создайте отчет по таблице БД **Кадры**, поэтому щелкните знак плюс перед ее названием. В результате на экране появится перечень полей таблицы **Кадры**.

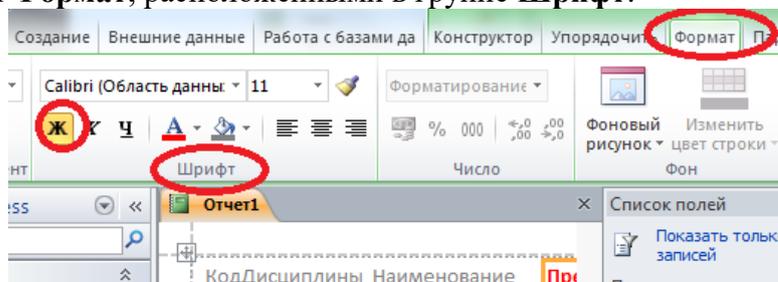


4. Добавьте в пустой макет нового отчета все нужные вам поля таблицы. Для вставки поля в отчет **дважды** щелкните название этого поля в области **Список полей**. Значения выбранного поля появятся в отчете.

Добавьте поля *Фамилия, Имя, Отчество, Дата рождения, Уч.группа, Гражданство*

6. Проведете сортировку данных в отчете. Фамилии людей в алфавитном порядке, для чего, щелкните правой кнопкой мыши название поля Фамилия и выберите команду **Сортировка от А до Я** в появившемся контекстном меню.

7. Настройте параметры отображения данных в отчете, воспользовавшись командами контекстной вкладки **Формат**, расположенными в группе **Шрифт**.



Последовательно щелкните заголовки столбцов отчета и установите для них полужирный шрифт.

Сохраните отчет и откройте его в режиме *конструктора*

8. Добавьте в отчет дополнительные элементы, воспользовавшись командами контекстной вкладки **Конструктор** (или **Формат**, в зависимости от версии субд). Для этого щелкните одну или несколько кнопок группы **Колонтитулы** (или **Элементы управления**, в зависимости от версии субд):

- **Эмблема** - служит для выбора изображения, которое будет использоваться в качестве эмблемы нового отчета;

- **(Заголовок)** - позволяет добавить или изменить заголовок нового отчета;

- **(Номер страницы)** - служит для вставки номеров страниц;

- **(Дата и время)** - добавляет в отчет элемент, отображающий текущее значение даты и / или времени;

- сетка;

- итог - рассчитать количество записей;

- номера страниц;

- добавьте поле;

9. Измените стиль оформления отчета. Для этого перейдите на контекстную вкладку **Формат**, в группе **Автоформат** нажмите кнопку **Автоформат** и в раскрывшемся списке щелкните образец схемы оформления отчета - **Поток**. Отчет будет отформатирован в соответствии с выбранной схемой.

Контрольные вопросы

1. Каково назначение отчета MS Access?
2. Перечислите способы создания отчета в Access
3. Перечислите разделы отчета в MS Access

Лабораторная работа №7

Команды языка запросов SQL в MS Office Access

Цель работы: научиться создавать запросы на выборку, перекрестные запросы; разработка эффективного инструмента выборки и корректировки данных; получение сводной информации из таблицы.

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

Рассмотрим инструкцию SQL (CREATE TABLE) запроса на изменение.

К этому типу запросов относятся запросы на создание таблицы, на добавление или на удаление записей в таблице и запросы на ее обновление. Структуру таблицы можно создать с помощью оператора CREATE TABLE.

Таблицы создаются командой CREATE TABLE.

Синтаксис:

```
CREATE TABLE <TABLE-NAME>  
    ( <COLUMN name><DATA type>[(<SIZE>)],  
<COLUMN name><DATA type> [(<SIZE>)] ... );
```

Практическая часть

Задание 1.

Рассмотрим создание структуры таблиц базы данных БД «Деканат» на основе модели «сущность – связь» в СУБД Access с помощью запросов SQL.

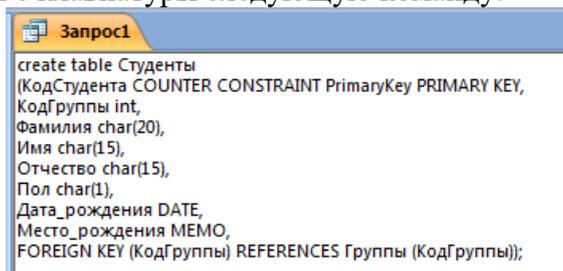
1. Создадим новую базу данных sql_training_st.mdb в приложении Access.
2. Сохраним файл новой базы данных в формате Access 2002-2003.
3. После создания новой БД, в окне приложения будет отображаться окно БД на вкладке Режим таблицы и новая пустая таблица с именем Таблица 1 в режиме таблица. Закрываем Таблицу1, щелкнув правой кнопкой мыши на Таблица1 в окне редактирования, и в контекстном меню выбрав команду Закрывать.
4. Далее создадим структуру таблицы Группы аналогичную структуре таблицы Группы, созданной в Конструкторе, используя команду SQL create table.
5. Для этого в окне БД щелкаем левой кнопкой мыши на вкладке Создание и выбираем команду "Конструктор запросов". В результате в окне редактирования откроется объект «Запрос1» и окно диалога «Добавление таблицы». Закроем окно диалога, щелкнув левой кнопкой мыши на пиктограмме «Закрывать» в правом верхнем углу этого окна.
6. Затем создаем структуру таблицы «Группы», для этого выберем режим SQL, выполнив команду Вид/Режим SQL.
7. Удаляем появившуюся в окне запроса команду SELECT и вводим с клавиатуры следующую команду (поле КодГруппы является счетчиком и первичным ключом):

```
create table Группы  
(КодГруппы COUNTER CONSTRAINT PrimaryKey PRIMARY KEY,  
Название char(6),  
Курс int,  
Семестр int);
```

8. Сохраняем запрос с именем "СозданиеГруппы". В результате в «Области переходов» появится несвязанный объект - «Создание Группы».
9. Необходимо выполнить этот запрос, щелкая на пиктограмме «Выполнить». В результате выполнения команды «create table Группы» в «Области переходов» появится объект - «Группы: таблицы».
10. Закроем окно «Создание Группы» и откроем объект – «Группы: таблица» в режиме конструктора.

Созданная с помощью запроса на изменение структура таблицы «Группы» аналогична структуре таблицы «Группы студентов», созданной в режиме «Конструктор».

11. Затем **аналогично** создаем структуру таблицы «Студенты», для этого выберем режим SQL, выполнив команду Вид/Режим SQL. Удаляем появившуюся в окне запроса команду SELECT и вводим с клавиатуры следующую команду:



```
create table Студенты
(КодСтудента COUNTER CONSTRAINT PrimaryKey PRIMARY KEY,
КодГруппы int,
Фамилия char(20),
Имя char(15),
Отчество char(15),
Пол char(1),
Дата_рождения DATE,
Место_рождения MEMO,
FOREIGN KEY (КодГруппы) REFERENCES Группы (КодГруппы));
```

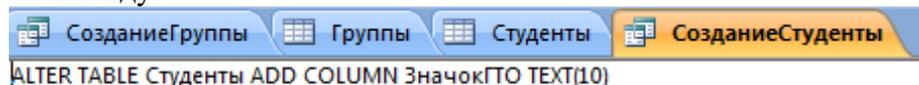
Для описания связей между таблицами "Группы" и "Студенты" через поле "КодГруппы" (отношение «один-ко-многим»), а также обеспечения целостности базы данных применена запись "FOREIGN KEY (КодГруппы) REFERENCES Группы (КодГруппы)".

12. Сохраняем запрос с именем "СозданиеСтуденты". В результате в «Области переходов» появится несвязанный объект - «Создание Студенты».

13. После сохранения запроса необходимо выполнить этот запрос, щелкая на пиктограмме «Выполнить». В результате выполнения команды «create table Студенты» в «Области переходов» появится объект - «Студенты: таблицы».

14. Не заполняя пока таблицы, изменим таблицу Студенты, добавив поле ЗначокГТО. Для этого создадим запрос ИзменениеСтуденты в режиме Конструктора.

14 Изменим команду:

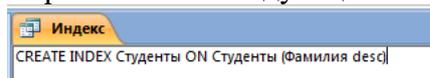


```
ALTER TABLE Студенты ADD COLUMN ЗначокГТО TEXT(10)
```

15. Сохраним запрос. Выполним его. Проверьте, что в таблице Студенты добавлен столбец ЗначокГТО

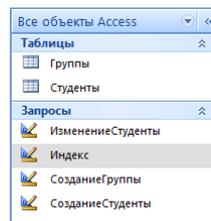
16. Создадим индекс для столбца Фамилия таблицы Студенты. Для этого в окне БД щелкаем левой кнопкой мыши на вкладке Создание и выбираем команду "Конструктор запросов". В результате в окне редактирования откроется объект «Запрос1». Перейдем в режим SQL.

17. Удаляем команду Select и прописываем следующее



```
CREATE INDEX Студенты ON Студенты (Фамилия desc)
```

18. Сохраним запрос. Выполним его. Что в результате? Так выглядит бд



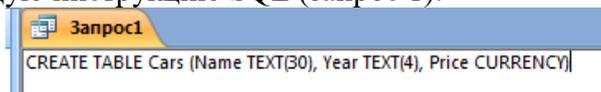
Задание 2.

Аналогично заданию 1, **используя команды SQL** создайте таблицы базы данных.

Предположим, необходимо создать таблицу для хранения марки, года выпуска и цены подержанных автомобилей, которые планируется приобрести. Название марки может занимать до 30 знаков, год — 4 знака. Чтобы составить запрос для создания этой таблицы, выполните указанные ниже действия.

1. Создание таблицы

Введите следующую инструкцию SQL (запрос 1):



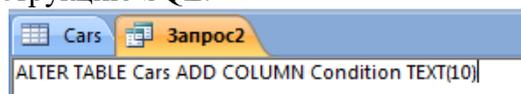
```
CREATE TABLE Cars (Name TEXT(30), Year TEXT(4), Price CURRENCY)
```

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

2. Изменение таблицы (запрос 2)

Предположим, требуется добавить 10-значное текстовое поле для хранения сведений о состоянии каждого автомобиля. Для этого можно выполнить указанные ниже действия.

1. На вкладке Создание в группе Другие нажмите кнопку Конструктор запросов.
2. Закройте диалоговое окно Добавление таблицы.
3. На вкладке Конструктор в группе Тип запроса щелкните элемент Определение данных.
4. Сетка конструктора пропадет, появится вкладка объекта режима SQL.
5. Введите следующую инструкцию SQL:

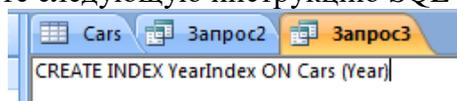


```
ALTER TABLE Cars ADD COLUMN Condition TEXT(10)
```

6. На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

3. Создание индекса

В новом запросе введите следующую инструкцию SQL



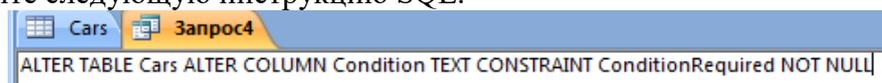
```
CREATE INDEX YearIndex ON Cars (Year)
```

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

4. Создание ограничения или связи

Предположим, есть таблица Cars с полями для хранения марки, года выпуска, цены и состояния подержанных автомобилей, которые планируется приобрести. Также предположим, что пользователь часто забывает ввести значение состояния автомобиля, хотя эти сведения необходимы. Чтобы запретить оставлять поле состояния пустым, можно создать ограничение для него. Для этого выполните указанные ниже действия.

1. На вкладке Создание в группе Другие нажмите кнопку Конструктор запросов.
2. Закройте диалоговое окно Добавление таблицы.
3. На вкладке Конструктор в группе Тип запроса щелкните элемент Определение данных.
4. Сетка конструктора пропадет, появится вкладка объекта режима SQL.
5. Введите следующую инструкцию SQL:

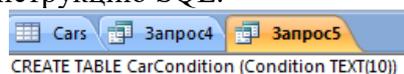


```
ALTER TABLE Cars ALTER COLUMN Condition TEXT CONSTRAINT ConditionRequired NOT NULL
```

6. На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Для приведения базы данных ко второй нормальной форме, можно создать таблицу с именем CarCondition с одним полем:

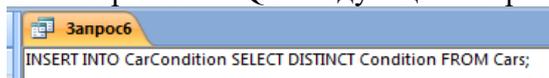
7. На вкладке Создание в группе Другие нажмите кнопку Конструктор запросов.
8. Закройте диалоговое окно Добавление таблицы.
9. На вкладке Конструктор в группе Тип запроса щелкните элемент Определение данных.
10. Сетка конструктора пропадет, появится вкладка объекта режима SQL.
11. Введите следующую инструкцию SQL:



```
CREATE TABLE CarCondition (Condition TEXT(10))
```

12. На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

13. Чтобы вставить в новую таблицу CarCondition значения из поля Condition таблицы Cars, введите на вкладке объектов режима SQL следующий запрос SQL:



ПРИМЕЧАНИЕ : Инструкцию SQL в этом шаге является запросом на добавление. В отличие от управляющего запроса, запрос на добавление заканчивается точкой с запятой.

14. На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Чтобы указать, что все новые значения, вставляемые в поле Condition таблицы Cars, должны совпадать с одним из значений в поле Condition таблицы CarCondition, создайте связь между таблицами CarCondition и Cars по полю Condition, выполнив указанные ниже действия.

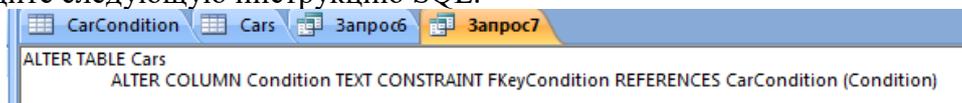
15. На вкладке Создание в группе Другие нажмите кнопку Конструктор запросов.

16. Закройте диалоговое окно Добавление таблицы.

17. На вкладке Конструктор в группе Тип запроса щелкните элемент Определение данных.

18. Сетка конструктора пропадет, появится вкладка объекта режима SQL.

19. Введите следующую инструкцию SQL:



20. На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Рассмотрим еще один пример использования таблицы автомобилей. Предположим, что вы хотите убедиться, что никакие две записи в таблице автомобилей не имеют один набор значений для имени, год, условия и цены. Можно создать уникальности, которые применяется для этих полей, выполнив следующие действия:

1. На вкладке Создание в группе Другие нажмите кнопку Конструктор запросов.

2. Закройте диалоговое окно Добавление таблицы.

3. На вкладке Конструктор в группе Тип запроса щелкните элемент Определение данных.

4. Сетка конструктора пропадет, появится вкладка объекта режима SQL.

5. Введите следующую инструкцию SQL:

```
ALTER TABLE Cars ADD CONSTRAINT NoDupes UNIQUE (name, year, condition, price)
```

6. На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Контрольные вопросы

1. Что такое SQL, назначение языка SQL?
2. Назначение команды SELECT?
8. Чем отличается использование WHERE от HAVING?

Лабораторная работа №8

Использование стандартных функций для работы с конструктором запросов в MS Office Access

Цель работы: научиться использовать стандартные функции для работы с конструктором запросов в СУБД MS Office Access.

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

Создание запросов

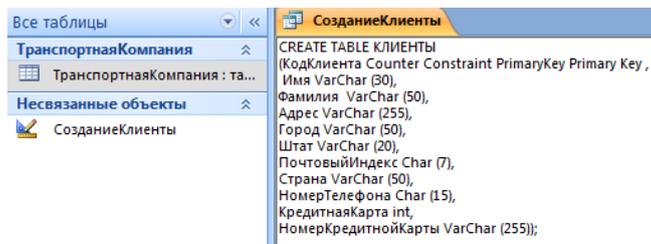
Для создания запроса необходимо перейти на вкладку «Создать» и выполнить «Создание запроса в режиме конструктора».

Практическая часть

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.

Ключ к заданию

1. Создание таблиц в режиме SQL выполняется с помощью запроса, содержащего аналогичный код



Для проверки использования вами режима SQL оставьте запросы для создания таблиц

2. Для создания маски ввода и значения по умолчанию откройте созданную таблицу в Конструкторе

3. Проверьте правильность типов полей при создании связей

Задание 2. Из таблицы «Транспортная_Компания» выбрать все транспортные компании, кроме «Кенгуру».

Ключ к заданию

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.

Задание 3. Создать запрос на увеличение тарифа на 0,8 таблицы «Доставка_Транспорт».

Ключ к заданию

1. Создание/Конструктор запросов

2. Добавьте необходимую таблицу

3. Выберите в Конструкторе тип запроса: На обновление

4 В строке Обновление добавьте запись *0.8

Задание 4. Удалить таблицу «*Доставка_Транспорт*»

Ключ к заданию

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*».

(Если запросы на удаление или обновление не выполняются, внимательно ознакомьтесь с теоретической частью)

Задание 5. Создать запрос на добавление данных в таблицу «*Клиенты*». Введите 2 новые записи. Сохраните запрос «*Добавление_Клиенты*»

Ключ к заданию

1. Создание/Конструктор запросов
2. Добавьте необходимую таблицу
3. Выберите в Конструкторе тип запроса: Добавление

Задание 6. Создать запрос на удаление данных из таблицы «*Сотрудники*». Удалите 2 записи. Сохраните запрос «*Удаление_Сотрудники*»

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

Ключ к заданию

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 MicrosoftAccess.

Вопросы

1. Что такое SQL, назначение языка SQL?
2. Назначение команды SELECT?

3. Что такое внешнее и внутреннее объединение, чем отличаются?

4. Назначение команды DROP?

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

Имя поля	Тип	Размер	Другие свойства
КодКлента	Текстовый	5	Длинное целое
Имя	Текстовый	Счетчик	
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

Имя поля	Тип	Размер	Другие свойства
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 №9

Создание базы данных в MS Server

Цель работы: научиться создавать реляционную базу данных в системе управления базами данных MS SQL Express.

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

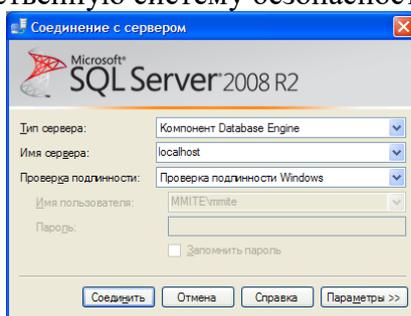
Утилита SQL Server Management Studio. Подавляющую массу задач администрирования SQL Server можно выполнить в *графической утилите* SQL Server Management Studio. В ней можно создавать базы данных и все ассоциированные с ними объекты (таблицы, представления, хранимые процедуры), можно выполнять последовательности инструкций Transact-SQL(запросы). В этой утилите можно выполнить типовые задачи обслуживания баз данных, такие как резервирование и восстановление, настраивать систему безопасности базы данных и сервера, просматривать журнал ошибок и многое другое.

Для запуска Management Studio в меню «Пуск» операционной системы выберите пункт «Microsoft SQL Server 2008 R2/Среда SQL Server Management Studio». Когда откроется окно программы, вас попросят подключиться к какому либо серверу баз данных SQL Server.

Подключение к серверу. В окне «Соединение с сервером» необходимо указать следующую информацию:

- *Тип сервера.* Здесь следует выбрать, к какой именно службе необходимо подключиться.
- *Имя сервера.* Позволяет указать, к какому серверу будет осуществляться подключение. По умолчанию имя SQL Server совпадает с именем компьютера.

- *Проверка подлинности.* Способ аутентификации, можно выбрать «Проверка подлинности Windows» или «Проверка подлинности SQLServer». Первый способ использует учетную запись, под которой текущий пользователь осуществил вход в Windows. Вариант SQL Server использует свою собственную систему безопасности.



Практическая часть

Задание 1. Запустите утилиту SQL Server Management Studio (Пуск/Все программы/ MS SQL Server 2008R2). Настройте подключение к серверу:

- компонент Database Engine
- локальный компьютер (localhost\SQLEXPRESS)
- проверки подлинности Windows

Перечислите имена сервера, которые предлагает. Какое имя использовали вы?

Задание 2. Перечислите команды *Меню* открывшегося окна SQL Server Management Studio

Задание 3. Вызовите команду *Вид*. Запишите назначение пиктограмм



Задание 4. Вызовите команду *Вид/Панели инструментов*. Перечислите предлагаемые панели инструментов.

Задание 5. Создайте базу данных *ОборотКомпанииФамилия* (вместо слова Фамилия добавьте вашу фамилию) с помощью графической утилиты SQL Server Management Studio.

Для этого:

1. В окне «Обозреватель объектов» найдите и раскройте папку «Базы данных». Щелкните на ней правой кнопкой мыши и выберите команду «Создать базу данных...».

2. В открывшемся диалоговом окне «Создание базы данных» на странице «Общие» введите следующую информацию:

Имя базы данных: *ОборотКомпанииФамилия*

Владелец: *sa*

В таблице «Файлы базы данных» измените путь к файлам данных и журнала на каталог *C:\Temp*.

Файлы базы данных:			
Размер (МБ)	Авторасширение	Путь	
1 МБ, рост не ограничен	...	C:\Temp	
10 %, рост не ограничен	...	C:\Temp	

Для всех остальных параметров оставьте значения по умолчанию.

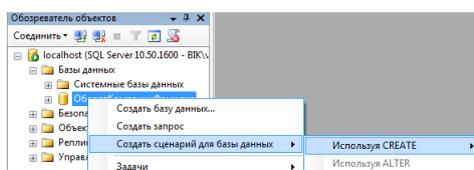
3. Для создания базы данных щелкните «ОК». Вы должны увидеть свою новую базу данных в окне «Обозреватель объектов».

4. Зайдите в каталог *C:\Temp*, найдите два файла созданной вами базы данных:

	ОборотКомпанииФамилия	15.02.2018 13:08	SQL Server Databa...	3 072 КБ
	ОборотКомпанииФамилия_log	15.02.2018 13:08	SQL Server Databa...	1 024 КБ

Задание 6. Создайте базу данных *ОборотКомпанииФамилия* (вместо слова Фамилия добавьте вашу фамилию) с помощью SQL команды CREATE. Для этого:

1. Сгенерируйте сценарий базы данных на основе уже существующей базы данных. Для этого в SQL Server Management Studio в контекстном меню узла «ОборотКомпанииФамилия»



2. Выберите команду «Создать сценарий для базы данных – Используя CREATE– Буфер обмена». В результате в буфер обмена будет сохранен текст запроса на создание новой базы данных с параметрами, указанными при создании базы данных в Management Studio.

3. Для проверки работоспособности сгенерированного запроса на создание базы данных удалите созданную вами базу данных. В контекстном меню базы данных выберите команду «Удалить» и в появившемся диалоговом окне нажмите кнопку «ОК». База данных со всеми файлами должна исчезнуть.

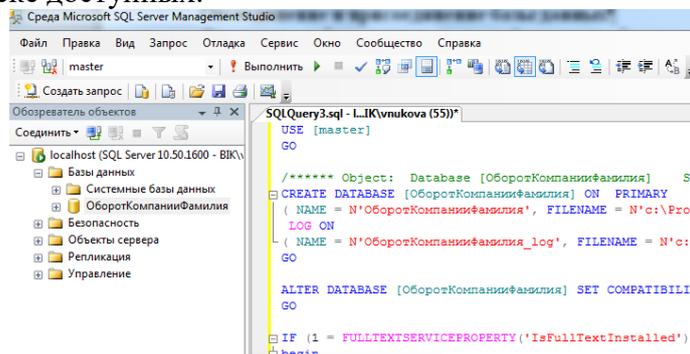
Чтобы воспользоваться сгенерированным заранее запросом на создание базы данных выполните следующие шаги:

4. На панели инструментов выберите команду «Создать запрос»  Создать запрос.

5. В открывшемся окне редактора SQL вставьте из буфера обмена (*Ctrl+V*) сгенерированный запрос.

6. Для запуска запроса на выполнение щелкните кнопку  Выполнить на панели инструментов или нажмите клавишу F5.

7. Обновите содержимое дерева обозревателя объектов командой «Обновить» из контекстного меню узла «Базы данных». База данных *ОборотКомпанииФамилия* должна вновь появиться в списке доступных.

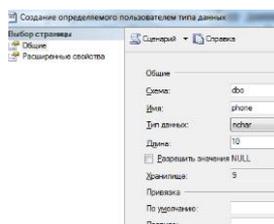


Задание 7. Создайте пользовательский тип данных *phone*, который будет использоваться в таблице *Customer* для хранения телефонного номера клиента. Для его создания воспользуйтесь утилитой Management Studio.

1. В дереве обозревателя объектов раскройте папки «Базы данных – ОборотКомпанииФамилия – Программирование - Типы». В контекстном меню узла «Определяемые пользователем типы данных» выберите команду «Создать определяемый пользователем тип данных».

2. В появившемся окне в текстовом поле «Имя» введите *phone*. В раскрывающемся списке «Тип данных» выберите *nchar*. В качестве длины введите 10. Отметьте параметр «Разрешить значения null», чтобы иметь возможность не указывать телефонный номер при добавлении нового клиента.

3. В секции «Привязки» оставьте пустые значения и щелкните на кнопке *Ok*. Созданный пользовательский тип данных должен появиться в дереве обозревателя объектов.



Задание 8. Создайте в базе данных пять таблицы. Первая таблица, *Customer*, будет хранить информацию о клиентах, вторая таблица *City* – справочник городов, третья, *Product*, - информацию о товарах, четвертая, *Order*, будет содержать подробную информацию о заказах и пятая, *OrderItem*, - о составе заказа (перечне товаров входящих в заказ). Ниже представлены все поля этих таблиц и их основные свойства.

Имя столбца	Тип данных	Разрешить null	Описание
Customer			
IdCust	int, identity	нет	Уникальный идентификационный номер клиента, на который можно ссылаться в других таблицах
FName	nvarchar(20)	нет	Имя клиента
LName	nvarchar(20)	нет	Фамилия клиента
IdCity	int	нет	Ссылка на номер города
Address	nvarchar(50)	нет	Адрес клиента
Zip	nchar(5)	нет	Почтовый индекс клиента
Phone	phone	да	Телефонный номер клиента
City			
IdCity	int, identity	нет	Уникальный идентификационный номер города
CityName	nvarchar(20)	нет	Название города
Product			
IdProd	int, identity	нет	Уникальный идентификационный номер для каждого товара
Description	nvarchar(100)	нет	Короткое текстовое описание товара
InStock	int	нет	Количество единиц продукта на складе
Order			
IdOrd	int, identity	нет	Уникальный идентификационный номер заказа
IdCust	int	нет	Ссылка на номер клиента

OrdDate	smalldatetime	нет	Дата и время размещения заказа
OrdItem			
IdOrd	int	нет	Ссылка на номер заказа
IdProd	int	нет	Ссылка на номер товара
Qty	int	нет	Количество единиц товара в заказе
Price	money	нет	Цена товара

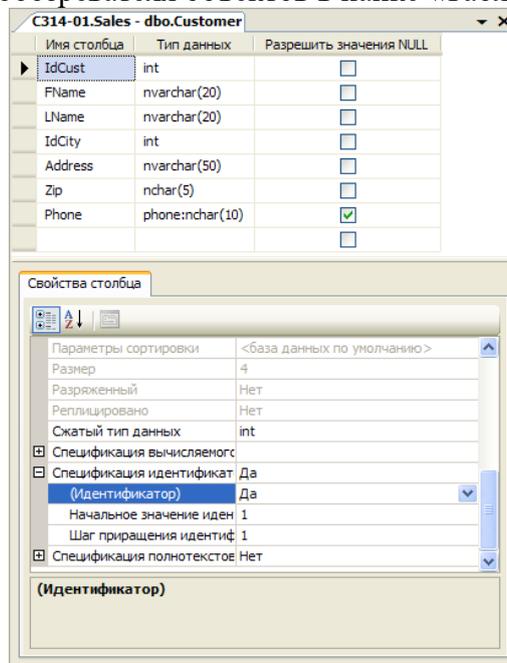
Таблицы можно создавать как в графическом интерфейсе (в утилите Management Studio), так и с помощью кода T-SQL. Воспользуемся самым простым, графическим способом. Сначала создадим таблицу *Customer*:

1. В дереве обозревателя объектов в базе данных в контекстном меню узла «Таблицы» выберите команду «Создать таблицу...». В рабочей области должна появиться вкладка с конструктором таблиц.

2. В первую строку в столбце «Имя столбца» введите IdCust, в столбце «Тип данных» выберите int. Убедитесь что параметр «Разрешить значения null» отключен.

3. В нижней половине экрана в разделе «Свойства столбцов» введите описание поля и измените значение параметра «Спецификация идентификатора / (Идентификатор)» на «Да» для того чтобы значения номера клиента формировались автоматически. Свойство «Идентифицирующий столбец» (Identity), обычно используемое совместно с типом данных int, предназначено для автоматического приращения значения на единицу при добавлении каждой новой записи. К примеру, клиент, добавленный в таблицу первым, будет иметь значение идентификатора 1, вторым – 2, третьим – 3, и т.д.

4. Аналогичным образом введите описания всех остальных полей и закройте окно конструктора таблиц. Введите в качестве имени таблицы Customer. Вновь созданная таблица должна появиться в дереве обозревателя объектов в папке «Таблицы».



В соответствии с вышеприведенным описанием создайте оставшиеся четыре таблицы: City, Product, Order и OrdItem.

Вопросы

1. Для чего предназначена утилита SQL Server Management Studio?
2. Перечислите предлагаемые программой SQL Server способы аутентификации.
3. Что отображается на панели Обзор серверов после подключения к серверу с помощью утилиты SQL Server Management Studio?
4. Опишите структуру окна утилиты Management Studio?

Лабораторная работа №10

Создание базы данных в MS Server

Цель работы: научиться создавать реляционную базу данных в системе управления базами данных MS SQL Express.

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

Утилита SQL Server Management Studio

Подавляющую массу задач администрирования SQL Server можно выполнить в графической утилите SQL Server Management Studio. В ней можно создавать базы данных и все ассоциированные с ними объекты (таблицы, представления, хранимые процедуры), можно выполнять последовательности инструкций Transact-SQL (запросы). В этой утилите можно выполнять типовые задачи обслуживания баз данных, такие как резервирование и восстановление, настраивать систему безопасности базы данных и сервера, просматривать журнал ошибок и многое другое.

Для запуска Management Studio в меню «Пуск» операционной системы выберите пункт «Microsoft SQL Server 2008 R2/Среда SQL Server Management Studio». Когда откроется окно программы, вас попросят подключиться к какому либо серверу баз данных SQL Server.

Практическая часть

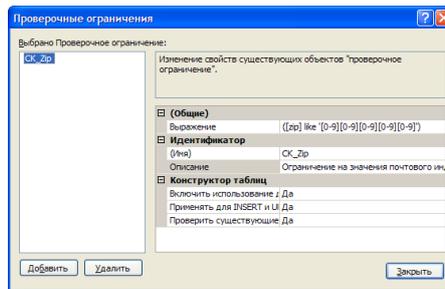
Задание 1. Запустите утилиту SQL Server Management Studio (Пуск/Все программы/ MS SQL Server 2008R2). Настройте подключение к серверу. Запишите последовательность ваших действий.

Задание 2. В Обзоре объектов найдите созданную вами на прошлой лабораторной работе базу данных *ОборотКомпанииФамилия.mdf*, которая должна состоять из пяти таблиц: *Customer, City, Product, Order, OrdItem*.

Создайте сценарий для первой таблицы, используя команду Create, в буфер обмена и в новое окно редактора запросов. Вставьте созданный сценарий в отчет по лабораторной работе.

Задание 3. Создайте ограничения для поля Zip (почтовый индекс) таблицы Customer, запрещающее вводить в поле буквы.

1. В контекстном меню папки «Ограничения» таблицы Customer выберите команду «Создать ограничение».
2. В открытом окне «Проверочные ограничения» заполните следующие поля:
 - Имя: CK_Zip
 - Выражение: ([zip] like '[0-9][0-9][0-9][0-9][0-9]'). Данное выражение описывает ограничение, принимающее пять символов, которыми могут быть только цифры от 0 до 9.
 - Описание: Ограничение на значения почтового индекса

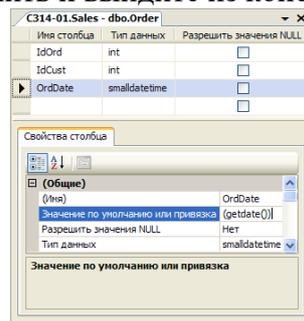


Задание 4. Создайте ограничения для полей *InStock* таблицы *Product*, *Price* таблицы *OrdItem*, запрещающие ввод в них отрицательных значений. В данном случае выражение проверки будет иметь вид (*Имя поля* > 0) для поля *Price* и (*Instock* >= 0) для столбца *InStock*.

Задание 5. Создайте ограничения *Использование значений по умолчанию*.

Установка для полей значений по умолчанию это отличный способ избавить пользователя от излишней работы, если значения этих полей во всех записях, как правило, принимают одни и те же значения. Так в таблице заказов *Order* вполне логично определить по умолчанию значение поля *OrdDate* (дата заказа) в виде текущей даты. В этом случае при добавлении записи о новом заказе в случае пропуска этого поля оно будет автоматически заполняться значением системной даты. Для создания такого свойства выполните следующие шаги:

1. Раскройте папку «Столбцы» таблицы *Order* и в контекстном меню поля «*OrdDate*» выберите команду «Изменить».
2. В свойстве столбца «Значение или привязка по умолчанию» введите `getdate()`. Эта функция T-SQL возвращает текущую системную дату.
3. Щелкните на кнопке Сохранить и выйдите из конструктора таблиц.



Задание 6. Установите для поля *InStock* (количество единиц продукта на складе) таблицы *Product* в качестве значения по умолчанию ноль.

Задание 7. Создайте первичный ключ для таблицы *Customer* - столбец *IdCust*. Для создания первичного ключа в таблице *Customer* выполните следующие шаги:

1. В контекстном меню таблицы *Customer* выберите команду «Проект».
2. В окне конструктора таблиц щелкните правой кнопкой мыши на поле *IdCust* и выберите команду «Задать первичный ключ» или нажмите кнопку  на панели инструментов. Обратите внимание на то, что слева от поля *IdCust* теперь отображается значок ключа, указывающий, что поле является первичным ключом.
3. Закройте конструктор таблиц с сохранением изменений

Задание 8. Аналогичным образом создайте первичные ключи для остальных таблиц в соответствие с ниже приведенной таблицей.

Таблица	Первичный ключ
City	IdCity
Product	IdProd

Order	IdOrd
OrdItem	IdOrd, IdProd (для выбора нескольких столбцов при установке составного ключа воспользуйтесь клавишами Shift или Ctrl)

Задание 9. Создайте ограничения на уникальность в таблице Customer по полю Phone, которое повторяется у разных клиентов не должно.

1. Для открытия конструктора таблиц в контекстном меню таблицы Customer выберите команду «Проект». На панели инструментов нажмите на кнопку «Управление индексами и ключами» .

2. В открывшемся окне «Индексы и ключи» щелкните кнопку «Добавить» и введите следующие параметры для нового уникального ключа:

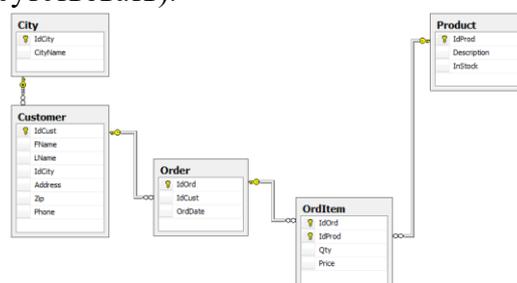
- Столбцы: Phone
- Тип: Уникальный ключ
- (Имя): СК_Phone

3. Закройте конструктор таблиц с сохранением изменений.

Задание 10. Аналогичным образом создайте ограничение уникальности по полю CityName таблицы City, чтобы обеспечить отсутствие в справочнике городов с одинаковыми названиями, а также по полю Description таблицы Product, чтобы иметь возможность отличить один товар от другого.

Задание 11. Создайте диаграмму базы данных:

1. В контекстном меню папки «Диаграммы базы данных» выберите команду «Создать диаграмму базы данных».
2. В диалоговом окне «Добавление таблиц» выберите все таблицы и нажмите на кнопку «Добавить».
3. Добавив таблицы, щелкните на кнопке «Заккрыть» и вы увидите созданную диаграмму базы данных (на рисунке представлен окончательный вид диаграммы: некоторые связи у вас могут отсутствовать).



Используя диаграмму базы данных ограничения внешнего ключа можно создавать значительно быстрее: лишь перетаскивая поля из одной таблицы в другую. В качестве примера создадим внешний ключ в таблице Customer по полю IdCity для связи с таблицей City:

1. Выделите в таблице City поле IdCity и, не отпуская кнопку мыши, перетащите его на поле IdCity таблицы Customer.
2. В диалоговых окнах «Таблицы и столбцы» и «Связь по внешнему ключу» примите настройки по умолчанию.
3. Сохраните диаграмму базы данных под именем ПМ.
4. Расположите таблицы в канонической форме (главные таблицы выше подчиненных) в соответствии с вышеприведенным рисунком.

Задание 12. Аналогичным образом создайте связь между таблицами Product и OrdItem по полю IdProduct. Окончательный список связей между таблицами со всеми их характеристиками представлен в следующей таблице:

Главная таблица	Подчиненная таблица	Поле связи (внешний ключ)	Правила каскадирования
City	Customer	IdCity	Без действия
Customer	Order	IdCust	Без действия
Order	OrdItem	IdOrd	Каскадное удаление
Product	OrdItem	IdProd	Без действия

Задание 13. После настройки всех ограничений можно наполнить таблицы данными. Для этого в контекстном меню таблицы выберите команду «Изменить первые 200 строк» и появившейся в рабочей области вкладке введите новые записи, заполняя все необходимые столбцы. В процессе внесения данных проверьте работоспособность всех созданных ранее ограничений:

- Ограничений проверки: попробуйте ввести в поле Zip (почтовый индекс) таблицы Customer нечисловые значения, а в поля InStock таблицы Product, Price таблицы OrdItem - отрицательные.
- Значений по умолчанию: убедитесь, что при пропуске полей OrdDate и InStock таблиц Order и Product для них устанавливаются значения по умолчанию в виде текущей системной даты и нуля соответственно.
- Ограничений первичного и уникального ключа: попробуйте ввести в таблицы записи с дублирующими значениями первичного или уникального ключа.
- Ограничений внешнего ключа: попробуйте ввести несогласованные данные в связанные таблицы, например, заказ для несуществующего клиента или удалить запись из любой главной таблицы при наличии связанных записей в подчиненной при отсутствии правил каскадирования.
- Правил каскадирования: убедитесь, что при удалении записи из таблицы Order все связанные записи из таблицы OrdItem удаляются автоматически.

Контрольные вопросы

1. Для чего предназначен первичный ключ?
2. В чем отличие ограничения на уникальность от первичного ключа?
3. Приведите примеры полей, которые хороши для установления ограничения на уникальность.

Лабораторная работа №11

Создание кнопок навигации и управления данными

Цель работы: научиться создавать элементы для навигации и управления данными реляционной базы данных.

Ход выполнения работы

1. Изучите теоретическую часть
2. Выполните практическое задание
3. Оформите отчет о выполнении лабораторной работы
4. Ответьте на контрольные вопросы

Теоретическая часть

Типичный сценарий работы с формой предполагает вывод только тех записей, которые удовлетворяют определенному признаку, например списка заказов конкретного клиента или перечень книг определенного автора. В настоящем сценарии пользователь вводит

необходимые сведения в поле формы, после чего выполняется запрос, использующий эти сведения в качестве условия поиска, другими словами, производится отбор данных на основании запроса с заданными параметрами. Запрос возвращает только те данные, которые интересуют пользователя.

Использование **запросов с параметрами** повышает эффективность приложения за счет того, что часть операций, а именно фильтрация и сортировка записей, осуществляется базой данных. Ситуация, в которой таблица базы данных запрашивается целиком, передается по сети и лишь затем обрабатывается приложением, приводит к снижению скорости работы и усложнению приложения.

В предлагаемом примере создается форма Windows Forms, в которой отображается список авторов, извлекаемый из стандартной базы данных SQL Server — *Pubs*. Пользователи имеют возможность ввести код штата США (например, CA для Калифорнии) и заполнить набор данных именами авторов, проживающих в указанном штате.

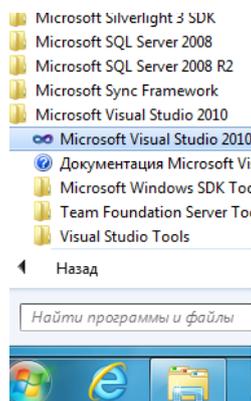


Практическая часть:

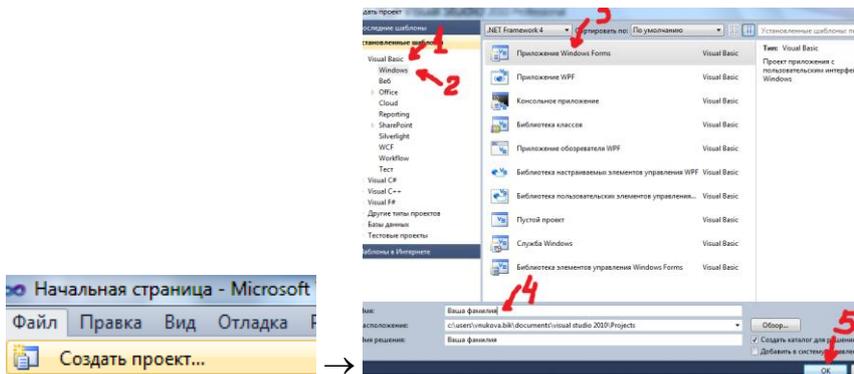
Задания выполняются в инструментальной среде MS Visual Studio

Настройка подключения данных и адаптера данных, создание набора данных

- 1.Создайте на рабочем столе папку с именем, созвучным вашей фамилии, поместите в вашу папку с сетевого диска папку База, содержащую файлы SQL базы данных: *pubs.mdf* и *pubs_log.ldf*
- 2.Запустите MS Visual Studio

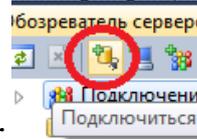


- 3.Создайте новый проект Windows Application (выберите язык программирования Visual Basic) с именем, созвучным вашей фамилии



Для того чтобы написать интерфейс к базе данных, необходимо подключиться к ней. Для этого используются провайдеры.

4. Вызовите **Обозреватель серверов** (Вид/Обозреватель серверов)



5. Вызовите команду **Подключиться к базе данных**

6. В окне **Выбор источника данных** выберите **Файл базы данных Microsoft SQL Server**, нажмите **Продолжить**

7. С помощью кнопки **Обзор**, найдите файл базы данных *pubs.mdf*, выберите ее, остальные параметры оставьте без изменения, проверьте подключение с помощью команды **Проверить подключение**

8. Если подключение установлено, нажмите **Ок**

9. На **Панели элементов** найдите элемент *SQLDataAdapter*, если его нет, вызовите контекстное меню панели элементов (правая кнопка мыши), выберите команду **Выбрать элементы...**

10. В открывшемся окне среди компонентов **Компоненты .Net Framework** найдите и отметьте *SQLDataAdapter*, нажмите **Ок**, после чего элемент появится на **Панели элементов**

11. Перетащите элемент *SQLDataAdapter* с панели **Toolbox** на форму.

12. В первом окне **Мастера настройки адаптера данных** выберите базу данных *pubs.mdf*, Мастер добавит базу данных в проект

13. Используйте инструкции **SQL**

14. В следующем окне, используя **Построитель запросов** (Query Bulder...) или вручную создайте инструкцию SQL следующего вида:

```
SELECT au_id, au_lname, state
FROM authors
WHERE (state = @state)
```

Мастер создает подключение (объект **SqlConnection1**) и адаптер данных (объект **SqlDataAdapter1**)

15. Используя команду меню **Данные**, создайте набор данных с именем *dsAuthors*: **Данные – Создать набор данных** (Data-Generate DataSet), добавив набор данных в конструктор (Add this dataset to the designer), оставив предлагаемое имя **DataSet1**

Добавление элементов управления, необходимых для отображения данных

В рассматриваемой в примере форме необходимо предусмотреть механизм, при помощи которого пользователь будет указывать код штата, выполнять запрос и выводить список авторов

16. Добавьте в форму перечисленные ниже элементы управления, присвоив им указанные имена

Элемент управления	Назначение	Имя	Текст
TextBox	Позволяет ввести код штата, в котором проживают интересующие пользователя авторы	txtStateParameter	(Пустая строка)
Button	Иницирует выполнение запроса и заполнение набора данных	btnShow	Показать
TextBox	Выводит идентификационный номер автора	txtAuthorID	(Пустая строка)
TextBox	Выводит фамилию автора	txtAuthorLName	(Пустая строка)
TextBox	Выводит название штата, где проживает указанный автор. (Используется в данном примере исключительно качестве иллюстрации.)	txtAuthorState	(Пустая строка)

17. Перед каждым текстовым полем следует добавить надписи, описывающие его функции (для этого используйте элементы управления Label, смотрите на скриншот)

Добавление кода, заполняющего набор данных

Отображение формы не сопровождается автоматическим заполнением набора данных сведениями, извлекаемыми из базы данных. Набор данных должен заполняться явным образом. В настоящем примере каждое нажатие кнопки **Показать** пользователем приводит к повторному выполнению запроса, хранящегося в адаптере данных, и занесению возвращенных им результатов в набор данных. В данном случае запрос имеет параметр. В качестве значения параметра берется информация, введенная пользователем в текстовом поле **txtStateParameter**.

Выполнение запроса иницируется в коде путем вызова метода **Fill** адаптера данных

12. Добавьте код к обработчику события **Click** кнопки **Показать**

```
SqlDataAdapter1.SelectCommand.Parameters("@state").Value = txtStateParameter.Text
DsAuthors1.Clear()
SqlDataAdapter1.Fill(DsAuthors1)
```

Привязка текстовых полей к набору данных

Для отображения одной записи таблицы Authors набора данных предусмотрено три текстовых поля со сведениями об авторах. Чтобы добиться их автоматического заполнения, необходимо привязать текстовые поля к столбцам исходной таблицы. Механизм привязки формы обеспечивает автоматическое обновление текстовых полей при каждом переходе между записями

13. Текстовое поле *Код автора* свяжите с полем **au_id** (для свойства **Text** узла **DataBindings** текстового поля Код автора разверните узлы **DsAuthors1** и **authors** и выберите из раскрывающегося списка пункт **au_id**.)

14. Аналогично свяжите текстовое поле *Фамилия* со столбцом **DsAuthors1.authors.au_lname**

15. Аналогично свяжите текстовое поле *Страна* со столбцом **DsAuthors1.authors.state**

Добавление элементов управления для перехода между записями

На завершающем этапе в форму добавляются элементы управления, обеспечивающие переход между записями. В настоящем примере это кнопки «Назад» и «Далее». Кроме этого, в форму добавляется текстовое поле, предназначенное для индикации положения текущей записи.

16. Добавьте в форму перечисленные ниже элементы управления, указав их свойства, как показано в приведенной таблице.

Элемент управления	Имя	Текст
Button	btnNext	Далее
Button	btnPrevious	Назад

17. Создайте метод обработки события **Click** («нажатие») кнопки **Назад**. Добавьте код, уменьшающий значение свойства **Position** объекта **BindingContext** на единицу

```
Me.BindingContext (DsAuthors1, "authors").Position -= 1
```

18. Создайте метод обработки события **Click** («нажатие») кнопки **Далее**. Добавьте соответствующий код

Отображение положения текущей записи

Теперь можно приступить к созданию метода **ShowPosition**, отображающего положение текущей записи.

Чтобы отобразить положение текущей записи

19. Добавьте в форму элемент управления **TextBox** и присвойте ему имя **txtPosition**.

Совет В свойстве **Enabled** задайте значение **False**, чтобы пользователи могли видеть текущее положение, но не имели возможности его изменить.

20. Создайте в форме метод с именем **ShowPosition**. Этот метод запрашивает положение текущей записи из объекта **BindingContext** и выводит в текстовом поле полученное значение. Следующий пример показывает, как выглядит конечный код обработчика.

Добавьте вызов метода **ShowPosition** в тех местах кода, где возможно изменение положения текущей записи. В настоящем примере это может происходить:

- после вызова метода **Fill** в обработчике события **Click** («нажатие») кнопки **Показать**;
- после изменения положения текущей записи в результате обработки события **Click** кнопок **Назад** и **Далее**. В качестве примера приводится код обработчика событий кнопки **Назад**.

```
Me.BindingContext (DsAuthors1, "authors").Position -= 1
ShowPosition()
```

Контрольные вопросы

1. Опишите бизнес-ситуацию проекта лабораторной работы № 11
2. Расскажите о создании подключения **OleDbConnection1** адаптера данных **OleDbDataAdapter1**
3. Расскажите о создании набора данных dsAuthors. Каково имя схемы, определяющей этот набор, и где она отображена?
4. Опишите процедуру связывания текстового поля с набором данных
5. Где создается SQL-запрос? Что заменяет знак (*) в запросе SQL?
6. Каково назначение кода

```
Me.BindingContext (DsAuthors1, "authors").Position -= 1
```

 Объясните его работу
7. Объясните назначение процедуры **ShowPosition** созданного проекта
8. Расскажите, какую информацию пользователю надо вводить, куда и что он может получить в ответ при запуске созданного вами проекта 😊

ЛАБОРАТОРНАЯ РАБОТА №12

ТЕМА: Изучение стандартных функций. Изучение средств поиска данных

Цель работы: Закрепить умения и навыки работы с базой данных в среде Visual Studio.Net, сформировать навыки обращения к различным источникам данных из одной формы

Теоретическая часть:

Бизнес-ситуация:

Когда компания Jones, Inc. приняла решение создать новую систему обработки данных, она уже имела прежнюю систему на основе базы данных Access, в которой директор компании Брэд Джонс хранил данные о заказах. Для директора очевидны преимущества перехода к новой системе на основе СУБД SQL Server, но прежде всего он хотел бы гарантировать сохранность унаследованных данных. Поэтому было решено вносить все изменения постепенно, особенно для сохранения данных компании за прошлые годы.

Итак, Джонс хотел бы создать новую систему на основе SQL Server, но иногда использовать старые данные из базы данных Access до тех пор, пока не будет закончена работа над новой системой. Поэтому часть данных должна храниться в одной базе данных, а часть — в другой. Две базы данных должны совместно использовать и даже объединять свои данные, хотя они относятся к разным типам баз данных.

К счастью, это требование легко удовлетворяется в модели ADO.NET. Для объекта DataSet совершенно не имеет значения, откуда, то есть из каких источников, берутся данные. Несмотря на то, что DataSet является фрагментом (кэшем) базы данных, он не имеет постоянной фактической связи с первоисточником. Объект DataSet — это контейнер, заполняемый информацией другим объектом — адаптером, данных - DataAdapter, который взаимодействует с первоисточником через SQL-запросы или хранимые процедуры. Один объект DataSet может взаимодействовать с несколькими объектами DataAdapter, каждый из которых обеспечивает наполнение данными таблиц контейнера.

Поскольку объект DataSet непосредственно не связан с источником данных, существует хорошая предпосылка для интеграции (объединения) данных, которые поступают из разных источников. Например, часть информации в DataSet может поступить из базы данных центрального офиса компании, часть из базы данных удаленного филиала, или вообще не из базы данных, а из другого источника, например, из электронной таблицы. Как только данные поступят в контейнер DataSet, пользователь может работать с ними как с единым информационным массивом, используя свойства и методы одного объекта, и абсолютно независимо от оригинального источника данных.

Создадим проект, выводящий информацию о клиенте по коду туристической путевки: в текстовое поле пользователь будет вводить код путевки (информация из базы Access таблицы Путевки), в поле со списком будет выводиться связанная информация о клиенте, который приобрел путевку (информация из базы SQL таблицы Информация о туристах), в сетке информация из таблицы Путевки.

 Для работы используйте базы данных BDTur_firm.mdb и BDTur_firmSQL.mdf, находящиеся в папке Базы

Практическая часть:

1. Создайте на рабочем столе папку с именем, созвучным вашей фамилии

Создание нового проекта и подключения данных

2. После запуска Visual Studio.Net создайте новый проект, выбрав шаблон *Приложение Windows Forms*. Назовите проект созвучно вашей фамилии.

В окне **Свойств** формы укажите значение *frmTurFirm* для свойства (Name) и значение *Выбор таблицы* для свойства Text формы Form1.

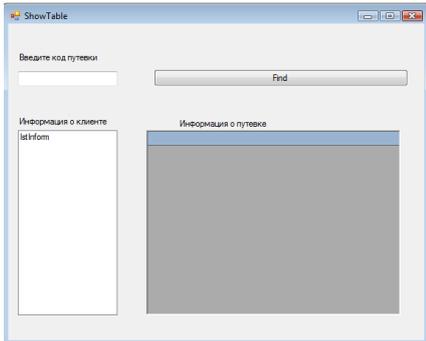
3. В верхнем левом углу формы создайте текстовое поле и укажите значение *txtCode* для его свойства (Name)

4. Создайте кнопку и укажите значение *btnFinds* для ее свойства (Name) и значение *Find* для ее свойства Text

5. Создайте поле со списком и укажите значение *lstInform* для его свойства (Name).

6. В правой части формы создайте сетку данных (элемент Data Grid) и укажите значение *grdTable* для ее свойства (Name).

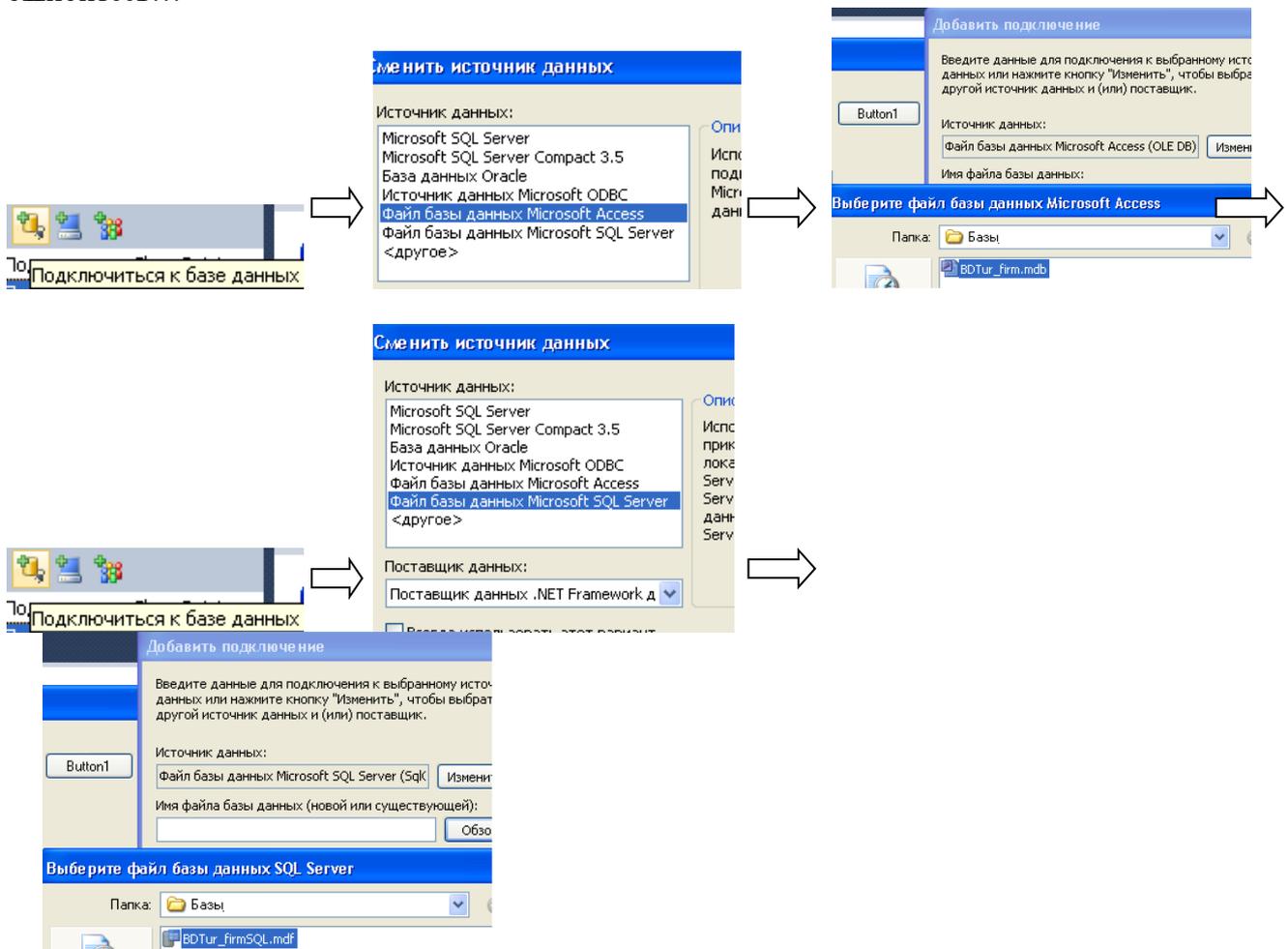
7. Над текстовым полем разместите метку с текстом «Введите код путевки», а над списком метку с текстом «Информация о клиенте»
8. Над сеткой разместите метку с текстом «Информация о путевке»
9. Расположите все эти элементы управления так, как показано на рисунке



В окне **Обозреватель серверов** подключитесь к базам данных из папки **Базы**.

Обратите внимание, что это две базы с разными расширениями: одна база данных формата .mdb, другая – .mdf.

Это означает, что для подключения вы будете использовать двух разных провайдеров. Не ошибитесь!!!



10. Перетащите в форму из панели элементов управления компонент DataSet (выбрав Нетипизированный набор данных) и укажите значение *dsTables* для его свойства (Name).

11. Создайте новый объект SqlDataAdapter для таблицы **Информация о туристах** базы данных (она лежит в папке **Базы**) **BDTur_firmSQL.mdf** СУБД SQL Server, перетаскивая его из панели элементов управления. После запуска программы-мастера настройки адаптера

данных выберите подключение к базе данных BDTur_firmSQL.mdf. Используйте инструкции SQL.

12. В диалоговом окне **Создание инструкций SQL** укажите команду `SELECT * FROM Информация о туристах` для загрузки данных в объект – набор данных.

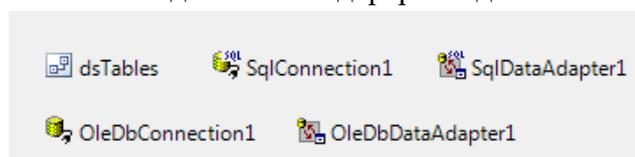
13. Для таблицы Путевки, которая находится в базе данных BDTur_firm.mdb Access создайте объект OleDbDataAdapter, перетаскивая его из панели элементов управления.

14. После запуска программы-мастера выберите подключение к файлу базы данных BDTur_firm.mdb

15. Затем в диалоговом окне **Выбор типа команд** выберите переключатель **Использовать инструкции SQL**

16. В диалоговом окне **Создание инструкций SQL** укажите команду `SELECT * FROM Путевки` для загрузки данных в объект – набор данных.

После этих действий под формой должны находиться следующие элементы:



17. Откройте код формы и введите:

```
Form1.vb* Form1.vb [Design]*
Imports System
Imports System.Data
Imports System.Data.SqlClient
Imports System.Data.OleDb
Public Class frmTurFirm
End Class
```

18. Затем в определении класса формы *frmTurFirm* введите код

```
Public Class frmTurFirm
    Private dvOrders As New DataView()
End Class
```

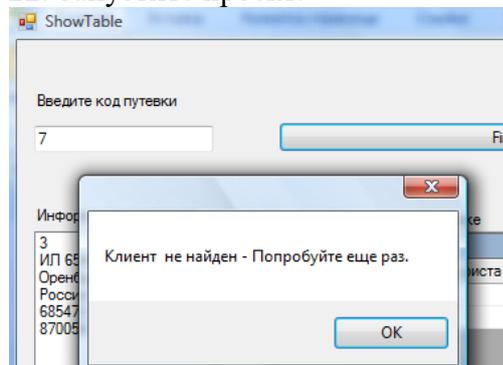
19. В событии Load формы введите следующий код

20. В событии Click кнопки Find введите следующий код:

Обратите внимание, что в данном примере создано отношение между двумя таблицами из баз данных совершенно разного типа!

21. Добавьте кнопку выхода

22. Запустите проект:



23. Создайте инсталляционный пакет.

При этом в папку Application Folder добавьте файлы используемых в проекте баз данных (в контекстном меню выберите команду **Add - File**). Это позволит при инсталляции установить и базы данных.

Контрольные вопросы

1. Опишите, как создается набор данных в проекте, как производится его заполнение?
2. Какими таблицами заполняется набор данных проекта, из каких баз данных? Укажите программный код для заполнения набора данных
3. Какие операции определены в коде подпрограммы btnFind_Click? Опишите их действия
4. Привяжите сетку данных к набору данных (как это сделать?), что изменится?
5. Объясните работу фрагмента кода:

```
Else  
    MessageBox.Show("Клиент не найден - Попробуйте еще раз.")  
    txtCode.Clear()  
End If
```

ЛАБОРАТОРНАЯ РАБОТА №13

Создание отчетов

Цель работы: сформировать умения и навыки работы с технологией Crystal Reports в среде Visual Studio.Net, сформировать навыки создания отчетов

Теоретическая часть:

Visual Studio поддерживает множество решений для составления отчетов, помогая реализовывать генерацию информативных и привлекательных отчетов в приложениях Visual Basic. Можно создавать и добавлять отчеты с помощью элементов управления ReportViewer, Crystal Reports или служб отчетов SQL Server.

ReportViewer — это отдельный, свободно распространяемый элемент управления данными, предоставляющий встроенные возможности обработки и просмотра отчетов.

Практическая часть:

1. Создадим отчет из таблицы Customers (Клиенты) базы данных MS SQL Server **AdventureWorks2008** (в папке к лабораторной работе №13). Для этого создайте на рабочем столе папку с именем, созвучным вашей фамилии

Задание 1. Создание проекта приложения Windows Forms

1. Откройте Visual Studio. В меню **Файл** щелкните **Создать** и выберите **Проект**.
2. В области **Установленные шаблоны** разверните узел **Другие языки** и выберите пункт **Visual Basic**.
3. Выберите шаблон **Приложение Windows Forms**.
4. В поле **Имя** введите **SimpleReport**.
5. В поле **Расположение** введите путь к папке, где будет сохранен проект, или нажмите кнопку **Обзор**, чтобы перейти в эту папку.
6. Нажмите кнопку **ОК**.
Откроется конструктор Windows Forms, отображая форму **Form1** созданного проекта.
7. Щелкните форму. В меню **Вид** выберите пункт **Окно свойств**. Разверните свойство **Размер**, чтобы отобразить поля **Ширина** и **Высота**. Установите в поле **Ширина** значение в 500 пикселей.

Задание 2. Определение набора данных и таблицы данных

1. В меню **Проект** выберите пункт **Добавить новый элемент**.
2. В окне **Добавление нового элемента** выберите пункт **Данные/Набор данных**. Введите имя набора данных и нажмите кнопку **Добавить**. Имя по умолчанию — **DataSet1.xsd**.
В результате в проект будет добавлен новый XSD-файл и откроется конструктор наборов данных.
3. Перетащите элемент управления **Адаптер таблицы** из области элементов в конструкторе набора данных в область конструктора.

- Запустится мастер настройки адаптера таблицы.
- На странице «Выбор подключения к данным» нажмите кнопку **Создать соединение**.
 - Если источник данных в Visual Studio создается впервые, откроется страница **Выбор источника данных**. В поле **Источник данных** выберите значение **Файл базы данных Microsoft SQL Server**.
 - Выберите файл базы данных **AdventureWorks2008_Data.mdf**
 - Использовать проверку подлинности Windows**
 - На странице **Сохранение строки подключения в файл конфигурации приложения** введите имя строки подключения или подтвердите имя по умолчанию - AdventureWorks2008ConnectionString. Нажмите кнопку **Далее**.
 - На странице **Выбор типа команд** выберите **Использовать инструкции SQL** и нажмите кнопку **Далее**.
 - На странице **Ввод инструкции SQL** введите следующий запрос Transact-SQL, чтобы получить данные о продажах из базы данных **AdventureWorks2008**, а затем нажмите кнопку **Готово**.

```
--SET DATEFORMAT mdy

SELECT
    PC.Name AS Category, PS.Name AS Subcategory,
    DATEPART(yy, SOH.OrderDate) AS Year,
    'Q' + DATENAME(qq, SOH.OrderDate) AS Qtr,
    SUM(DET.UnitPrice * DET.OrderQty) AS Sales
FROM Production.ProductSubcategory PS INNER JOIN
    Sales.SalesOrderHeader SOH INNER JOIN
        Sales.SalesOrderDetail DET ON SOH.SalesOrderID = DET.SalesOrderID INNER JOIN
            Production.Product P ON DET.ProductID = P.ProductID
            ON PS.ProductSubcategoryID = P.ProductSubcategoryID INNER JOIN
                Production.ProductCategory PC ON PS.ProductCategoryID = PC.ProductCategoryID
WHERE (SOH.OrderDate BETWEEN ('20020101') AND ('20031231'))
GROUP BY DATEPART(yy, SOH.OrderDate), PC.Name, PS.Name,
    'Q' + DATENAME(qq, SOH.OrderDate), PS.ProductSubcategoryID
```

--SET DATEFORMAT mdy

```
SELECT
    PC.Name AS Category, PS.Name AS Subcategory,
    DATEPART(yy, SOH.OrderDate) AS Year,
    'Q' + DATENAME(qq, SOH.OrderDate) AS Qtr,
    SUM(DET.UnitPrice * DET.OrderQty) AS Sales
FROM Production.ProductSubcategory PS INNER JOIN
    Sales.SalesOrderHeader SOH INNER JOIN
        Sales.SalesOrderDetail DET ON SOH.SalesOrderID = DET.SalesOrderID INNER
        JOIN
            Production.Product P ON DET.ProductID = P.ProductID
            ON PS.ProductSubcategoryID = P.ProductSubcategoryID INNER JOIN
                Production.ProductCategory PC ON PS.ProductCategoryID = PC.ProductCategoryID
WHERE (SOH.OrderDate BETWEEN ('20020101') And ('20031231'))
GROUP BY DATEPART(yy, SOH.OrderDate), PC.Name, PS.Name,
    'Q' + DATENAME(qq, SOH.OrderDate), PS.ProductSubcategoryID
```

Также можно нажать кнопку **Построитель запросов**, чтобы использовать построитель запросов для создания запроса, и проверить его с помощью кнопки **Выполнить запрос**.

- Нажмите кнопку **Готово**.
Теперь в конструкторе наборов данных появится определение DataTable для **DataTable1** с подкатегориями, имена которых получены

из столбцов и псевдонимов столбцов запроса (**Category, Subcategory, Year, Qtr** и **Sales**). Эти поля будут использоваться во время привязки данных к отчету.

Примечание

Если нужно изменить поля в таблице данных, щелкните правой кнопкой мыши заголовок **DataTable1** или **DataTable1TableAdapter** на странице конструктора наборов данных. Нажмите кнопку **Настроить**, чтобы перезапустить мастер настройки адаптера таблицы.

12. Сохраните файл DataSet1.

Задание 3. Добавление нового RDL-файла с помощью мастера отчетов

1. В меню **Проект** выберите пункт **Добавить новый элемент**.
2. В диалоговом окне **Добавление нового элемента** выберите пункт **Мастер отчетов**.
3. В поле **Имя** введите **Sales Orders.rdlc** и нажмите кнопку **Добавить**.
За диалоговым окном откроется область графического конструктора.
4. На странице **Свойства набора данных** выберите созданный набор данных из раскрывающегося списка **Источник данных**.
Окно **Доступные наборы данных** автоматически будет обновлено добавлением созданного набора **DataTable**.
5. Нажмите кнопку **Далее**.
6. На странице **Упорядочение полей** выполните следующие действия.
 - a. Перетащите поле **Category** из списка **доступные поля** в окно **Группы строк**.
 - b. Перетащите поле **Subcategory** из списка **доступные поля** в окно **Группы строк** под полем **Category**.
 - c. Перетащите поле **Year** из списка **доступные поля** в окно **Группы столбцов**.
 - d. Перетащите поле **Qtr** из списка **доступные поля** в окно **Группы столбцов** под полем **Year**.
 - e. Перетащите поле **Sales** из списка **доступные поля** в окно **Значения**.
7. Дважды щелкните **Далее**, затем щелкните **Готово**.
Будет создан RDL-файл, который откроется в конструкторе отчетов. Сконструированный табликс теперь отобразится в области конструирования.

Задание 4. Добавление заголовка к отчету

1. Щелкните табликс в области конструктора, чтобы рядом с таблицей и над таблицей появились маркеры строк и столбцов.

Примечание

Маркеры имеют вид серых прямоугольников над табликсом и рядом с табликсом. Они используются для выполнения различных действий с группами столбцов, группами строк и всем табликсом. Маркеры, расположенные над табликсом, являются маркерами столбцов. Маркеры, идущие сверху вниз вдоль табликса, являются маркерами строк. Маркер, расположенный на стыке маркеров строк и столбцов, называется угловым маркером.

2. Щелкните угловой маркер, чтобы выбрать весь табликс и показать перекрещенные стрелки.
3. Переместите табликс вниз по области конструктора, перетащив перекрещенные стрелки.

4. Перетащите элемент **Текстовое поле** из области элементов в область над таблицей в области конструктора, а затем расположите его, перетащив перекрещенные стрелки.
5. Выберите текстовое поле и введите с клавиатуры Company Sales. При необходимости перетащите границу текстового поля, чтобы расширить его.
6. Выберите текстовое поле, нажмите кнопку **Жирный** на панели инструментов **Форматирование отчета**, а затем выберите пункт **16пт** в поле **Размер шрифта**.
7. Выберите текстовое поле и нажмите кнопку **Основной цвет** на панели инструментов **Форматирование отчета**.
8. В диалоговом окне «Выбор цвета» выберите **Черно-синий** цвет и нажмите кнопку **ОК**.

Задание 5. Добавление элемента управления ReportViewer на форму

1. Щелкните файл **Form1.vb** в обозревателе решений.
2. В меню **Вид** выберите команду **Конструктор**.
3. Перетащите элемент управления **ReportViewer** из раздела **Отчет** области элементов на форму.
4. Откройте панель смарт-тегов в элементе управления ReportViewer1, щелкнув глиф смарт-тега в правом верхнем углу. Щелкните раскрывающийся список **Выберите отчет** и выберите файл **SimpleReport.Sales Orders.rdlc**.
5. На панели смарт-тегов щелкните элемент **Закрепление в родительском контейнере**. В оставшейся части этого пошагового руководства можно построить собственное приложение и в любое время просмотреть отчет на форме. Чтобы увидеть, как добавочные изменения в структуре отчета влияют на конечный отчет, постройте и просмотрите отчет в качестве последнего шага каждой из следующих процедур.
6. Нажмите клавишу F5, чтобы построить приложение и просмотреть отчет на форме (необязательно).

Контрольные вопросы:

1. Изучить и охарактеризовать возможности Report Design
2. Как добавить новый отчет. Пошагово объяснить работу мастера
3. Объект ReportViewer, его свойства

Лабораторная работа №14

Создание базы данных в Base

(создание таблиц и организация связей между таблицами)

Цель работы: сформировать умения и навыки работы в СУБД Base OpenOffice.org

Теоретическая часть

Находясь в базе данных OpenOffice.org, можно пользоваться данными, хранящимися в различных форматах файлов баз данных. Исходно база данных OpenOffice.org поддерживает некоторые обычные файловые форматы баз данных, например dBase. Кроме того, можно использовать базу данных OpenOffice.org для подключения к внешним реляционным базам данных, например, к базам данных MySQL или Oracle. Во время работы часто возникают вопросы по выполнению тех или иных задач. OpenOffice.org BASE имеет хорошую справочную систему, которая поможет ответить практически на любой вопрос. Для вызова справки выберете пункт **Вызов справки**.

Практическая часть

Запустите программу **OpenOffice.org Base**.

Задание 1. В окне Мастера выберите команду *Выбор базы данных – Подключиться к существующей базе данных*. Перечислите первые шесть пунктов раскрывающегося списка.

Задание 2. В окне Мастера выберите команду *Выбор базы данных – Создать новую базу данных – Далее – Нет, не регистрировать базу данных – Открыть базу на редактирование*. Задайте имя, созвучное вашей фамилии (оставьте автоматическое расширение). После открытия окна, ознакомьтесь с его интерфейсом. Запишите, какие команды меню предлагаются пользователю.

Задание 3. Используя команду *Сервис – Параметры – Загрузка/Сохранение*, узнайте параметры автосохранения, запишите их

Создайте базу данных фирмы «Атлант», которая занимается реализацией оборудования для организации цифрового телевидения в Белгороде. База данных должна включать список предлагаемого оборудования; перечень организаций-клиентов фирмы; данные о представителях этих организаций – заказчиках; данные о заказах за текущий год.

Задание 4. В режиме дизайнера создайте четыре таблицы базы данных. База состоит из 4 таблиц:

- таблица «*Оборудование*» (поля: КодОборудования, тип данных – текстовый (Varchar), размер 3 символа, обязательно поле, ключевое поле; Наименование – текстовый (Varchar), размер 20 символов, обязательное поле; Цена – денежный формат (Число, обязательное, значение по умолчанию – 0, пример формата – Деньги)),

- таблица «*Организация*» (поля: Организация – ключевое, Адрес – оба поля текстовые),

- таблица «*Заказчики*» - предполагается, что в каждой организации может быть несколько подразделений, заказывающих оборудование, в этой таблице приводятся данные для связи с ними (поля: КодЗаказчика – ключевое, Фамилия, Имя, Отчество, Телефон, Организация – все текстовые, для телефона полезно указать маску ввода, формат выберите произвольно),

- таблица «*Заказы*» (поля КодЗаказа – счетчик, ключевое поле, КодЗаказчика, КодОборудования, Дата – тип дата/время, Количество – тип числовой, целое). Следите, чтобы одинаковые поля в различных таблицах имели одинаковый тип.

Задание 5. Введите данные в таблицу *Организации*, используя режим таблиц. Введите не менее пяти из приведенных ниже организаций (обязательно включите Айпи Ноут).

<p>СЕРВИСНЫЙ ЦЕНТР</p> <p>Белгород, ул. Народный бульвар 64 Принять</p> <p>Работает круглосуточно</p> <p>Цены</p> <p>Установка и подключение телевизора от 500 руб.</p> <p>+7 (904) 093-09-90 Написать сообщение</p>	<p>ИТ-КОНТРОЛЬ</p> <p>Белгород, пр-кт. Б. Хмельницкого, д. 135 оф.31 Принять</p> <p>Открывается завтра в 09:00</p> <p>Цены</p> <p>Установка и подключение телевизора</p> <p>+7 (4722) 41-94-19 Написать сообщение</p>	<p>РЕМОНТ+</p> <p>Белгород, ул. Сумская 8/1 Принять</p> <p>Открывается завтра в 09:00</p> <p>Цены</p> <p>Установка и подключение телевизора от 1 000 руб.</p> <p>+7 (930) 090-06-96 Написать сообщение</p>
<p>СЕРВИС ПРО</p> <p>Белгород, ул. Б. Хмельницкого 130А Принять</p> <p>Работает круглосуточно</p> <p>Цены</p> <p>Установка и подключение телевизора от 499 до 999</p> <p>+7 (980) 379-40-59</p>		

Задание 6. Создайте схему данных (Сервис / Связи). Введите в схему все четыре таблицы (Добавить), установите связи между таблицами *Оборудование – Заказы* (по полю КодОборудования), *Организации – Заказчики* (по полю Организации), *Заказчики – Заказы* (по полю КодЗаказчика).

Для этого в первой (главной) таблице из пары связываемых таблиц выбирается соответствующее поле и буксируется с помощью левой кнопки мыши в соответствующее поле подчиненной таблицы. В открывшемся окне Связи устанавливается тип отношения, при этом установкой флажка можно потребовать обеспечение целостности данных.

В отчет вставьте скриншот полученной схемы.

Контрольные вопросы:

4. Опишите назначение пиктограмм , , .
5. Как создать базу данных в Base. Пошагово объясните работу мастера.
6. Какие режимы существуют для создания таблиц базы данных в `Base`.

Лабораторная работа №15

Создание форм и отчетов в Base

Цель работы: сформировать умения и навыки работы в СУБД Base OpenOffice.org

Теоретическая часть

На первом этапе создания базы данных была определена структура таблиц БД и установлены связи между ними. Второй этап предусматривает загрузку в БД оперативной (учетной) информации, а также просмотр и корректировку базы данных.

В **OpenOffice.org Base** существует два способа ввода данных в БД:

- непосредственно в таблицы базы данных;
- с помощью специально разработанных экранных форм.

Первый способ используют в том случае, если данные изменяются достаточно редко, а база данных имеет простую структуру, т.е. в ней нет сложных связей между таблицами.

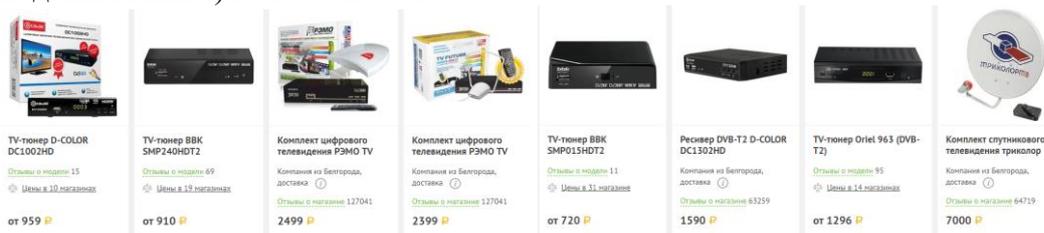
Второй способ предпочтителен в том случае, если приходится часто изменять данные, а база данных состоит из большого количества таблиц, связанных друг с другом.

Практическая часть

Запустите программу **OpenOffice.org Base**.

Продолжаем работу с базой данных фирмы «Атлант», которая занимается реализацией оборудования для организации цифрового телевидения в Белгороде. База данных должна включать список предлагаемого оборудования; перечень организаций-клиентов фирмы; данные о представителях этих организаций – заказчиках; данные о заказах за текущий год.

Задание 1. Создайте, используя мастер форм, простую форму (Форма / Создать / Мастер) «Оборудование», с ее помощью введите данные об имеющихся в продаже продуктах (из приведенных ниже) и их стоимости.



Просмотрите соответствующую таблицу.

Задание 2. Создайте составную форму *Организация – Заказчики*.

Для создания составной формы вначале определяется пара взаимосвязанных таблиц, одна из которых – главная, вторая – подчиненная. После нажатия кнопки Создать выбирается имя главной таблицы, на следующем диалоговом окне с помощью мастера для составной формы вводится имя подчиненной таблицы.

Поле, связывающее главную таблицу с подчиненной, вводится только для главной таблицы. По завершении создания составной формы мастер запросит сохранение формы для подчиненной таблицы.

При вводе записей с помощью составной формы можно вначале заполнять главную таблицу (если она не заполнена), а затем, листая ее записи, ввести соответствующие каждой записи из главной таблицы записи в подчиненную таблицу.

Введите для каждой организации по 2 – 3 заказчика (например, для организации заказчиками могут быть Мастер[НомерМастера], Отдел сбыта, а ФИО – данные заказчике или о руководителях соответствующего подразделения).

Задание 3. Создайте составную форму *Заказчики – Заказы*.

Для удобства ввода данных выведите рядом с формой окно с таблицей *Оборудование*, уменьшив его размер. Введите данные о заказах (по 1 – 4 заказу – оборудованию – на каждого заказчика, количество экземпляров каждого продукта – от 1 до 12).

Сделайте скриншот таблицы заказов и вставьте его в отчет.

Задание 4. Подготовьте и сохраните данные по таким запросам (при необходимости внесите свои изменения в запрос):

- какое оборудование, кем и в каком количестве заказаны в Айпи Ноут (введите таблицы Заказчики, Заказы, Оборудование; поля – КодОборудования, Наименование, Количество, Фамилия, Имя, Отчество, Организация; сортировка – Наименование – по возрастанию (по алфавиту); вывод на экран – все поля, кроме Организации; условие отбора – в поле Организация – Айпи Ноут; сохраните запрос под именем Айпи Ноут),

- какие организации, кто и в каком количестве заказал TV-тюнер ВВК SMP24HDT2,

- какое оборудование и кем было заказаны в первом полугодии текущего года,

- список организаций, заказавших Комплект цифрового телевидения РЭМО TV или Комплект спутникового телевидения триколор,

- опробуйте другие варианты запросов, при необходимости используйте справочную систему Base.

Контрольные вопросы:

1. Опишите назначение пиктограмм , , ,
2. Как создать форму в Base. Пошагово объясните работу мастера.
3. Какие режимы существуют для создания запроса к базе данных в Base.

Список литературы

Основные источники:

1. Кузин А. В. Базы данных. Учебник. Гриф УМО вузов России - М.: Академия, 2015.
2. Семакин И.Г. Основы программирования и баз данных [Текст]: учеб. для студентов учреждений среднего проф. образования / И. Г. Семакин. - М.: Академия, 2017. - 224 с. - (Профессиональное образование. Информатика и вычислительная техника)
3. Семакин И.Г. Основы алгоритмизации и программирования [Текст]: учеб. / И. Г. Семакин, А. П. Шестаков. - 4-е изд., стер.;. - М.: Академия, 2017. - 304 с. - (Профессиональное образование. Информатика и вычислительная техника)
4. Фуфаев Э. В. Базы данных. Учебное пособие для вузов - М.: Академия, 2015.

Дополнительные источники:

1. Агальцов В. П. Базы данных. Учебник. В 2-х книгах. Книга 2: Распределенные и удаленные базы данных. Гриф УМО вузов России; Форум, 2016.
2. Диго С.М. Access Учебно-практическое пособие Проспект, Велби, 2014.
3. Мальцев М.Г. Базы данных: учебник; ГРИФ; Корона-Век, 2014.
4. Пирогов В.Ю. Информационные системы и базы данных: организация и проектирование. Учебное пособие. - БХВ-Петербург, 2015.
5. Сеннов А.С. Access 2017: учебный курс (+CD).

Интернет- ресурсы:

1. Цветкова, А. В. Информатика и информационные технологии [Электронный ресурс]: учебное пособие для СПО / А. В. Цветкова. -Электрон. текстовые данные. - Саратов: Научная книга, 2019. - 190 с. - Режим доступа: <http://www.iprbookshop.ru/87074.html>