

**Уважаемые студенты групп!**

**Вашему вниманию представлена лекция на тему «Основы БД в Delphi». Лекция рассчитана на 6 часов**

### **Задание**

1. Прочитать внимательно лекцию.  
2. Законспектировать лекцию в рабочую тетрадь не менее 3-6 страницы рукописного текста. В конспекте лекции обязательно должно быть приведены примеры.

3. Ответить письменно в рабочей тетради на контрольные вопросы.

С уважением Ганзенко Ирина Владимировна

!!! Если возникнут вопросы обращаться по телефону +79591134803 (телеграмм)

[disobuch.ganzenko2020@mail.ru](mailto:disobuch.ganzenko2020@mail.ru)

## **Основы БД в Delphi (Лекция)**

### **ПЛАН ЛЕКЦИИ**

1. Основные понятия
2. Создание псевдонима базы данных
3. Создание таблиц базы данных
4. Объявление полей
5. Изменение языкового драйвера
6. Определение индексов
7. Определение ссылочной целостности между таблицами
8. Изменение структуры таблицы
9. Открытие и сохранение таблицы
10. Дополнительные утилиты

### **1. Основные понятия**

На сегодня среда Delphi является одним из самых распространенных средств создания приложений баз данных.

**База данных** (БД) - это структурированный набор постоянно хранимых данных. Постоянность означает, что данные не уничтожаются по завершении программы или пользовательского сеанса, в котором они были созданы.

**Системы управления базами данных** (СУБД) - это программные средства, предназначенные для создания, наполнения, обновления и удаления баз данных.

Информация в БД хранится в виде совокупности связанных **таблиц**. Каждая таблица представляет собой совокупность строк и столбцов. Строки соответствуют экземпляру объекта, конкретному событию или явлению. Столбцы таблицы - атрибутам (признакам, характеристикам, параметрам)

объекта, события, явления. В терминах БД столбцы таблицы называются *полями*, а строки - *записями*.

Базы данных, между отдельными таблицами которых существуют связи, называются реляционными. Реляционная модель была предложена в 1970-е годы Тедом Коддом.

Связанные отношениями таблицы взаимодействуют по принципу главная-подчиненная. Главную таблицу называют родительской, подчиненную – дочерней. Одна и та же таблица может быть главной по отношению к одной таблице БД и дочерней по отношению к другой таблице.

## 2. Создание псевдонима базы данных

При работе с таблицами локальных БД (в число которых входят таблицы СУБД Paradox и dBase) сама база данных размещается на диске и хранится в виде набора файлов. Для хранения одной таблицы создается отдельный файл. Обращение к БД из утилит и программы осуществляется по *псевдониму* базы данных. Псевдоним должен быть зарегистрирован в файле конфигурации конкретного компьютера при помощи утилиты *BDE Administrator*. Для присвоения псевдонима необходимо запустить утилиту *BDE Administrator* (пункт меню Пуск/Программы/Borland Delphi 5/ BDE Administrator). В главном меню утилиты необходимо выбрать элемент *Object / New*. В появившемся окне задается тип создаваемой БД - *Standard*.

После этого в левом поле окна администратора БД (рис. 4.1) появится строка с именем STANDARD1, стандартным именем для псевдонима. Это имя целесообразно изменить на имя псевдонима создаваемой БД.

В правом поле указаны параметры БД. Необходимо изменить параметр PATH, который указывает путь, в котором располагается БД.

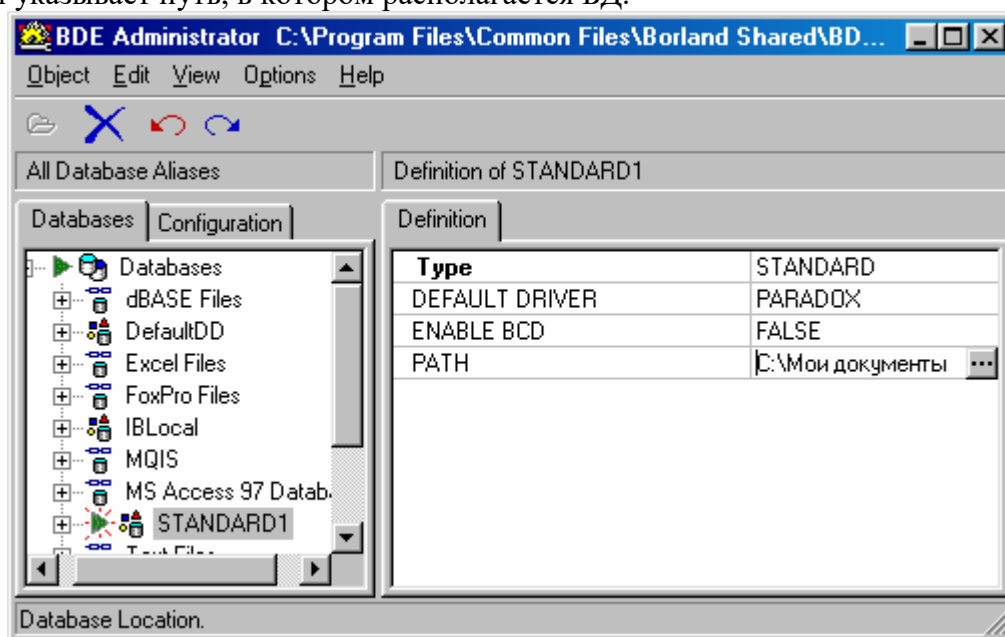


Рис. 4.1

Далее необходимо запомнить определение псевдонима. Для этого в левом окне администратора БД необходимо щелкнуть по имени псевдонима правой кнопкой мыши и в коротком меню выбрать опцию *Apply* и подтвердить выбор в появившемся диалоговом окне. На этом создание псевдонима БД завершается.

### 3. Создание таблиц базы данных

Для создания таблицы необходимо сформировать ее *структуру*, т.е. задать совокупность полей таблицы и определить их типы. Для создания таблиц базы данных необходимо запустить утилиту *Database Desktop (DBD)*. После запуска утилиты необходимо установить псевдоним той базы данных, с таблицами которой предполагается работать. Для этого нужно выбрать элемент главного меню *File / Working Directory* и в выпадающем списке *Aliases* выбрать имя псевдонима, после чего нажать ОК. *Database Desktop* – мощное средство, обеспечивающее различные способы работы с таблицами БД. Рассмотрим лишь некоторые возможности.

### 4. Объявление полей

Для создания таблицы БД нужно выбрать элемент главного меню *File / New / Table*. В появившемся окне *Create Table* задать тип создаваемой таблицы (по умолчанию *Paradox 7*) и нажать ОК. После этого появится окно определения структуры таблицы БД (рис. 4.2)

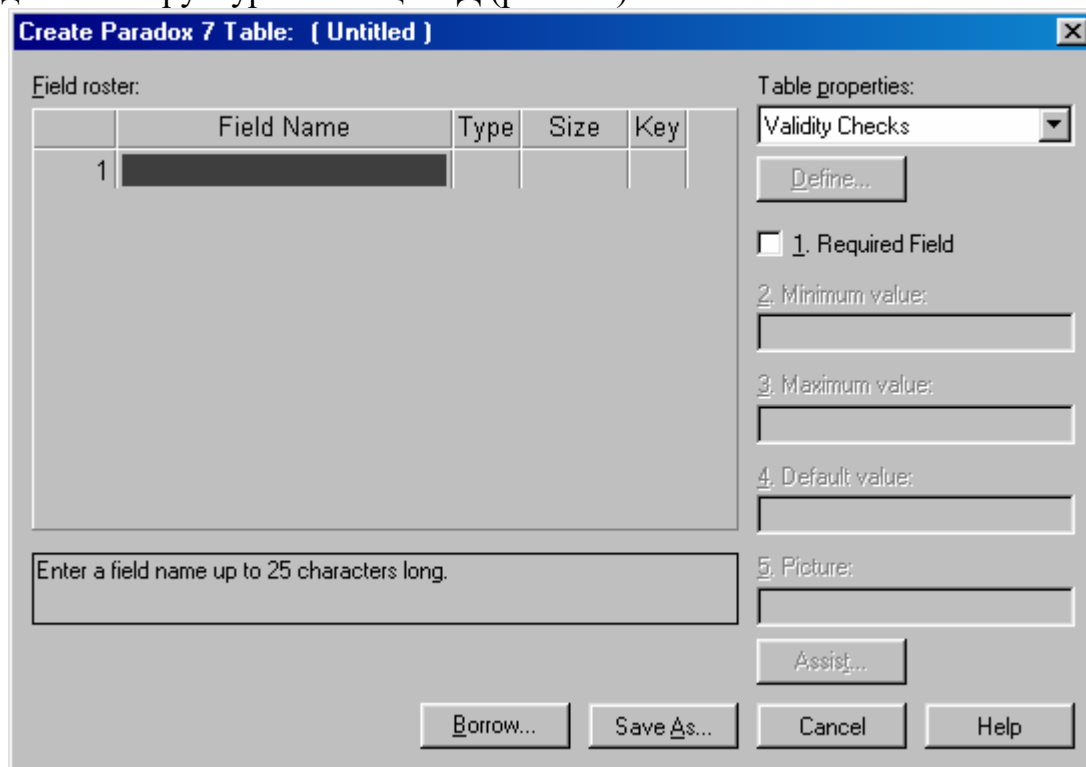


Рис. 4.2

Каждая строка соответствует полю таблицы. Назначения столбцов:

- **Field Name** – имя поля; необходимо для обеспечения доступа к полю. Имена полей в таблице должны быть уникальны.
- **Type** – тип поля. Тип определяет совокупность операций, которые можно производить со значениями поля.
- **Size** – размер поля (для строковых полей).
- **Key** – содержит звездочку «\*», если поле входит в состав *первичного ключа*. Первичный ключ состоит из одного поля или нескольких полей, однозначно идентифицирующих запись. Это означает, что в таблице *не может быть* двух записей с одними и теми же значениями ключевых полей. Ключ может состоять из любого количества подряд идущих полей, начиная с первого. Подробнее способы использования ключевых полей рассматриваются в лекции 6.

Типы полей, используемых в СУБД *Paradox*, представлены в Таблице 1:

Таблица 1

Тип поля и обозначение		Хранимые значения
Alpha	A	Символьные значения длиной до 255 символов
Number	N	Числовые значения с плавающей точкой в диапазоне –10307 ... +10308. Точность до 15 значащих цифр
Money	\$	Аналогичен типу Number, но предназначен для хранения денежных сумм. Число знаков после запятой по умолчанию равно двум. При показе значения выводится знак денежной единицы
Short	S	Целочисленные значения в диапазоне -32768 ... 32767
LongInteger	I	Целочисленные значения в диапазоне -2147483648 ... 2147483647
BCD	#	Числовые значения, в том числе и дробные, в двоично-десятичном формате
Date	D	Значения даты
Time	T	Значения времени
Timestamp	@	Значения даты и времени
Memo	M	Строковые значения длиной более 255 символов. Максимальная длина не ограничена
Formatted Memo	F	Форматированный текст произвольной длины, в котором отдельные фрагменты текста могут использовать разные шрифты, цвета и стили
Graphic Fields	G	Графические изображения в форматах BMP, PCX, TIF, GIF, EPS, которые при хранении преобразуются в формат BMP. Хранятся отдельно от основной таблицы БД
OLE	O	Информация в форматах, поддерживаемых технологией OLE
Logical	L	Логические значения (True и False)
Autoincrement	±	Автоинкрементное поле. Значения доступны только для

		чтения. Обычно – ключевое поле в составе первичного ключа. При добавлении новой записи значение поля вычисляется автоматически, так, чтобы в одной таблице не было одинаковых значений. Значения из удаленных записей повторно не используются
Binary	B	Двоичные значения произвольной длины. Хранятся отдельно от основной таблицы
Bytes	Y	Произвольные двоичные значения. Хранятся вместе с таблицей БД

Для любого поля можно определить требование обязательного его заполнения значением. В этом случае для поля включается переключатель *Required Field*.

Строки ввода, расположенные ниже, служат для наложения ограничений на значение поля:

- *Minimum value* – определяет минимальное значение поля;
- *Maximum value* – определяет максимальное значение поля;
- *Default value* - определяет значение поля по умолчанию;
- *Picture* – определяет шаблон изображения поля.

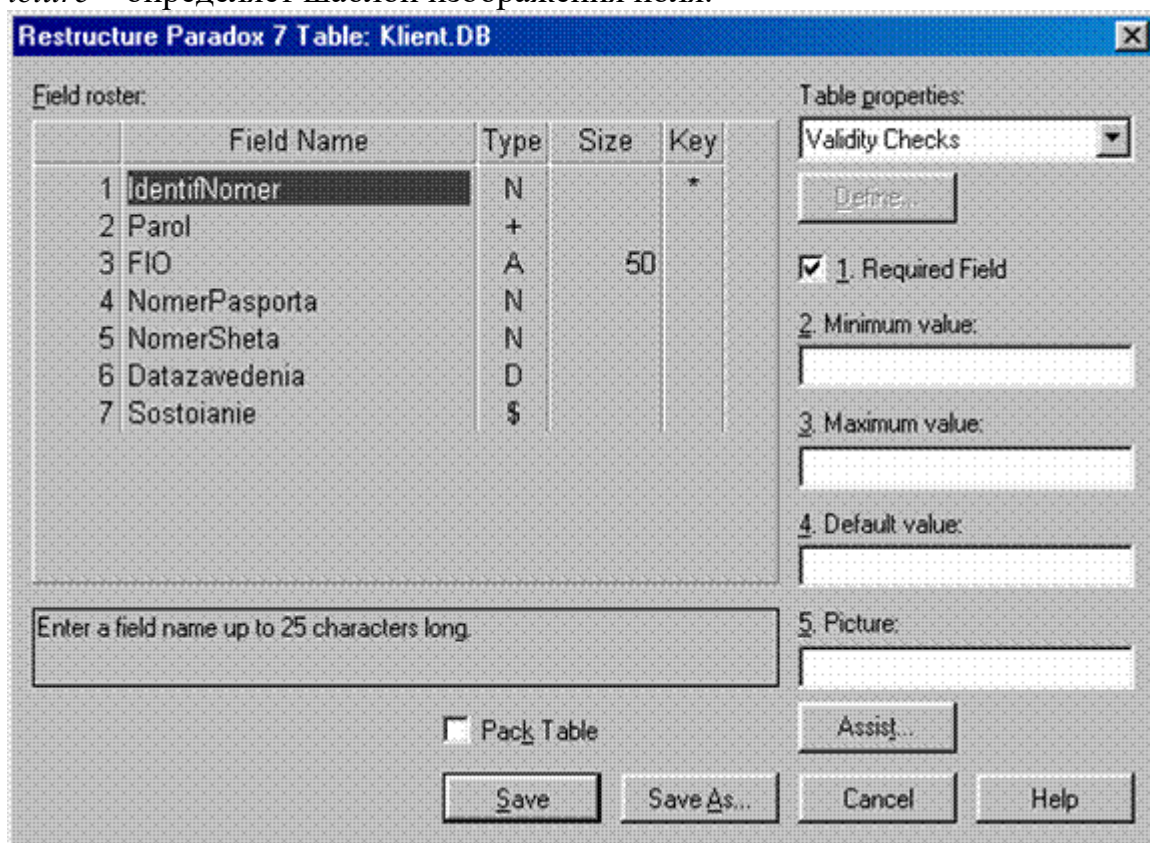


Рис. 4.3

На рис. 4.3 приведен пример использования ключевого поля при создании таблицы. В данном примере ключевым полем будет являться Идентификационный номер (поле Key отмечено \*). Выбор этого поля ключевым

определяется тем, что в данной таблице не должно быть двух записей имеющих одинаковые идентификационные номера.

## 5. Изменение языкового драйвера

Для того, чтобы строки таблицы могли быть заполнены русским текстом (кириллица), необходимо установить соответствующий языковой драйвер, чтобы русский текст отображался на экране без искажений.

Для этого необходимо в комбинированном списке *Table Properties* щелкнуть элементу *Table Language*, после чего щелкнуть по кнопке *Modify* и в появившемся окне *Table Language* (рис. 4.4.) указать драйвер *Pdox ANSI Cyrillic* или *Paradox Cyrr 866*.

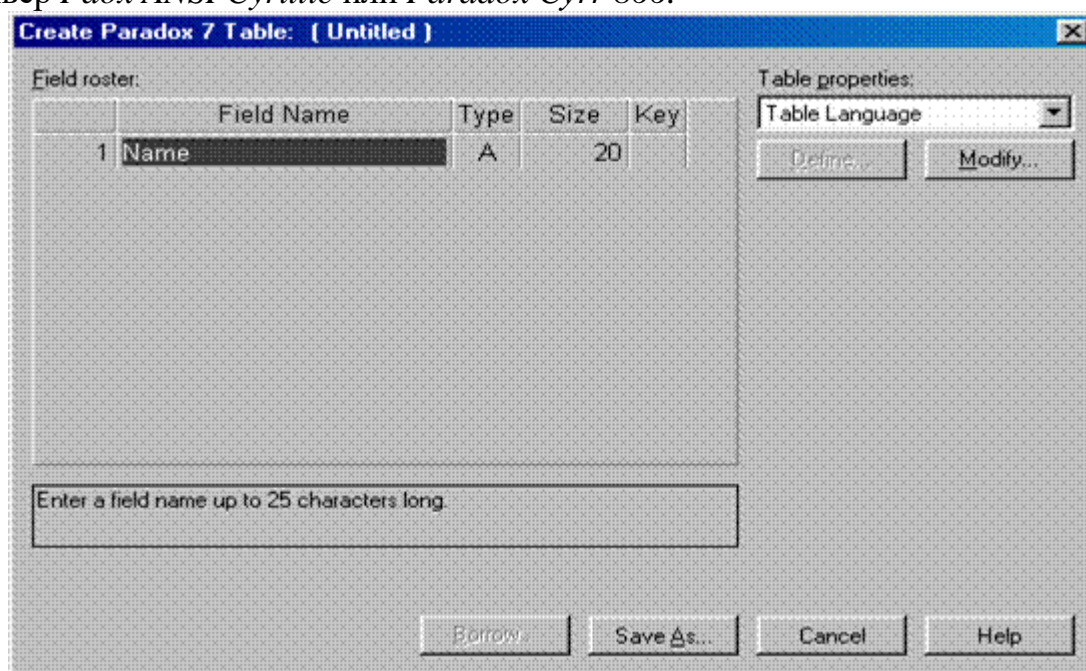


Рис. 4.4

## 6. Определение индексов

*Индексы* предназначены для обеспечения быстрого доступа к определенным записям таблицы и сортировки записей таблицы по значениям выбранных полей. Суть индексов состоит в том, что они хранят отсортированные значения индексных полей (т.е. полей, по которым построен индекс) и указатель на запись в таблице. Для каждой таблицы может быть определено несколько индексов. Каждый индекс хранится в отдельном файле. Индекс создается и по первичному ключу, если ключ задан. В этом случае индекс называется *первичным*.

Для создания дополнительных, *вторичных*, индексов в комбинированном списке *Table Properties* необходимо выбрать элемент *Secondary Indexes*. Для того чтобы определить новый индекс, необходимо нажать кнопку *Define*. В появившемся диалоговом окне (рис. 4.5) в

поле *Fields* содержится список всех полей таблицы. Окно *Indexed Fields* предназначено для хранения полей, входящих в создаваемый индекс. Нужные поля необходимо перенести в это окно с помощью стрелки «⇒».

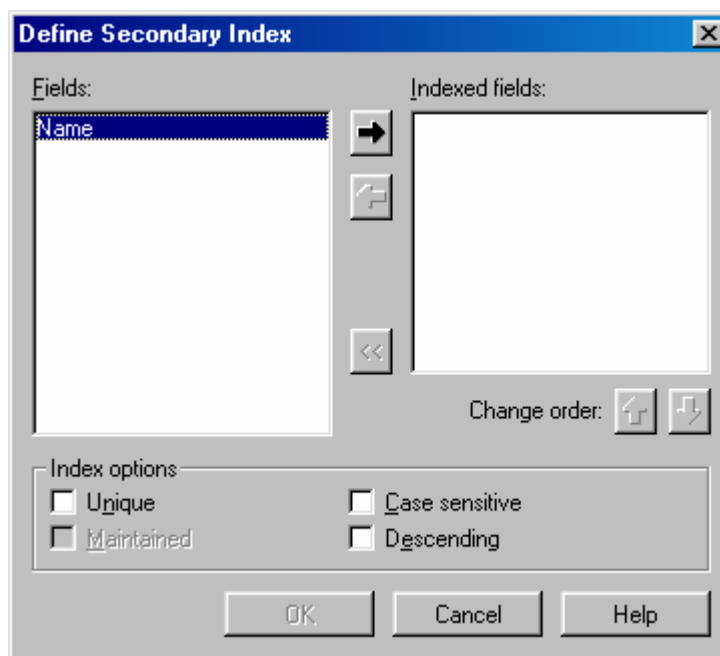


Рис. 4.5

В появившемся диалоговом окне задается имя индекса. Не рекомендуется составлять название индекса только из имен полей.

## 7. Определение ссылочной целостности между таблицами

Ссылочная целостность определяет связь между таблицами и вид каскадных взаимодействий между ними.

Для установки ссылочной целостности в списке *Table Properties* выбирают элемент *Referential Integrity* и нажимают *Define*. В появившемся диалоговом окне (рис. 4.6) в списке *Fields* показаны поля выбранной таблицы, а в списке *Tables* – остальные таблицы базы данных.

Сначала указывают поле связи для выбранной таблицы. Для этого в списке *Fields* выбирают поле и нажимают кнопку с изображением стрелки вправо. Название поля будет записано в поле *Child Fields*. Затем в поле *Tables* выбирают другую таблицу и нажимают кнопку с изображением стрелки влево.

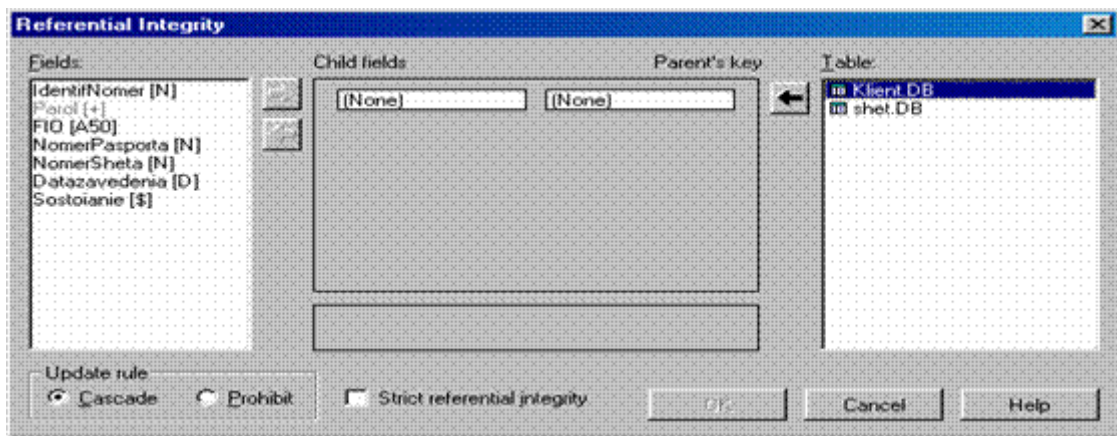


Рис. 4.6

В поле *Parents Key* будут показаны поля из первичного ключа таблицы. Переключатели *Update rules* определяют вид каскадных воздействий на первую таблицу при изменении значения поля связи во второй таблице или при удалении в ней записи:

- *Cascade* – разрешены каскадные изменения (изменения или удаления в записях дочерней таблицы при одновременном изменении (удалении) записи родительской таблицы) и удаления подчиненных записей в дочерней таблице;
- *Prohibit* - запрещены изменения полей связи или удаление записи в родительской таблице, если для данной записи есть связанные записи в дочерней таблице.

Отмечают вид каскадного взаимодействия. Появится запрос об имени ссылочной целостности. Вводят имя и нажимают ОК. Имя созданной ссылочной целостности будет помещено в список.

## 8. Изменение структуры таблицы

Структуру созданной таблицы можно изменить. Для этого необходимо выбрать элемент меню *Tools / Utilities / Restructure*, выбрать таблицу и произвести необходимые изменения в структуре таблицы. Можно добавлять, удалять поля, изменять их типы и имена. В некоторых случаях система попросит подтвердить выполнение действий.

## 9. Открытие и сохранение таблицы

Для изменения данных, хранящихся в таблице, таблицу необходимо «открыть». Открытие таблицы происходит с помощью подменю *File / Open*. Внесение изменений (добавление/удаление записей, изменение значений полей конкретных записей и т.д.) происходит только в режиме *редактирования*. Для перехода в режим редактирования необходимо нажать клавишу F9. Сохранение изменений в таблице происходит по выбору



пунктов меню *File / Save* или *File / SaveAs*. Перед сохранением изменений необходимо выйти из режима редактирования, повторно нажав клавишу F9.

## 10. Дополнительные утилиты

Рассмотрим некоторые дополнительные возможности, которые среда Database Desktop предоставляет для работы с таблицами. Эти возможности доступны через пункт меню *Tools / Utilities*:

*Add* – добавить записи из одной таблицы в другую (структуры должны совпадать);

*Copy* – копировать одну таблицу в другую;

*Delete* – удалить таблицу;

*Empty* – опустошить таблицу (удалить все записи);

*Info Structure* – показать структуру таблицы;

*Rename* – переименовать таблицу;

*Sort* – отсортировать таблицу по значениям каких-либо полей;

*Restructure* – изменить структуру таблицы;

*Subtract* – удалить из таблицы записи, совпадающие с записями другой таблицы (записи должны совпадать).