Тема16. Работа с базами данных (практика)

Прежде чем начать строить приложения, работающие с базами данных, надо иметь сами базы данных. Для их создания не обязательно использовать стандартные СУБД - Delphi содержит все необходимые для этого средства. В частности, в Delphi имеется утилита универсального назначения **Database Desktop,** которая позволяет создавать таблицы баз данных некоторых, самых распространенных форматов, задавать и изменять структуру таблиц.

Обычно вызов **Database Desktop** включен в главное меню Delphi в раздел Tools. Ее так же можно запустить через **Пуск - Программы - Borland Delphi - Database Desktop.**

Задание 1. Создание двухтабличной базы КЛИЕНТЫ-ДОГОВОРА

Создадим с помощью **Database Desktop** таблицы базы данных СУБД Paradox. В Paradox база данных - это каталог, в котором лежат таблицы - файлы с расширением **.db**.

Последовательность действий:

1. Создать каталог, в котором будет размещаться база данных.

2. Запустить Database Desktop.

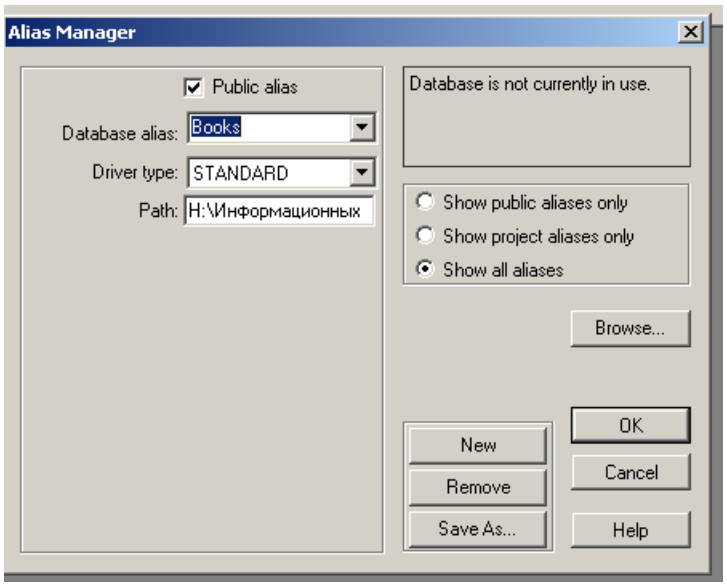
3. Создать псевдоним (alias). Как правило, в приложениях используют не прямое указание на базу данных, а ее псевдоним. Псевдоним содержит всю информацию, необходимую для доступа к базе данных. Использование псевдонимов обеспечивает физическую независимость приложений - изменение каталогов, сервера и т.п. не влечет изменение текста приложения. Один из способов создания псевдонима - выполнить команду меню **Tools/Alias Manager** утилиты **Database Desktop**. В появившемся окне нажмите кнопку **New**, и задайте следующие параметры:

• в поле **Database alias** введите имя псевдонима вашей базы данных (придумайте сами).

• **Driver Type** - выберете значение **Standard (**для обработки баз данных Paradox используется стандартный драйвер).

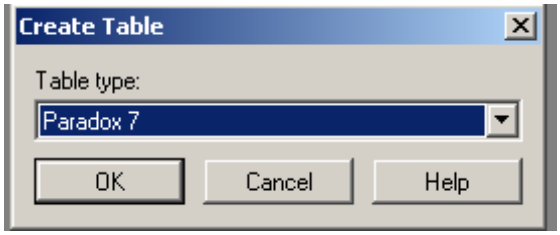
• нажмите кнопку **Browse** и укажите путь к каталогу, где будет размещаться ваша база данных.

• Затем ОК - псевдоним создан.

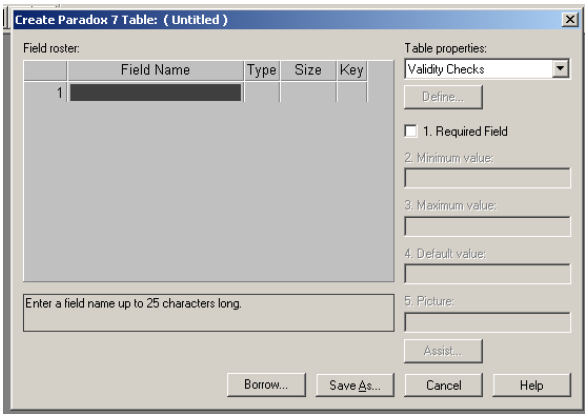


4. Создание таблицы. Создадим таблицу **КЛИЕНТЫ**. Это делается по команде меню **File/New/Table**.

5. Выбор типа таблицы. Перед Вами появится диалоговое окно выбора типа таблицы. Выберете **Paradox 7.**



6. Задать структуру таблицы - порядок, имена и типы полей. В следующем окне, специфичном для каждого формата базы данных, вы сможете задать структуру таблицы, создать вторичные индексы, ввести диапазоны допустимых значений полей, значения по умолчанию и многое другое.



Назначение колонок:

• **Field Name** - имя поля,

• **Type** – тип поля,

• **Size** – размер поля,

• **Key** – признак ключевого поля (первичный индекс).

Имя поля может включать до 25 символов и должно начинаться с символа отличного от пробела (но внутри пробелы допускаются).

Для задания типа поля перейдите к столбцу **Type**, а затем нажмите пробел или щелкните правой кнопкой мышки. Появится список доступных типов (табл.), из которых выбирается нужный вам.

Ключевые поля должны быть отмечены символом «*» в последней колонке. Для того чтобы поставить или удалить этот символ, нужно сделать двойной щелчок в графе Key соответствующего поля или нажать пробел.

Таблица **КЛИЕНТЫ** должна содержать следующие поля:

• **ClientNum** - номер клиента, тип Autoincrement;

• **OrgName** - название организации, тип Alpha, размер 20;

• **Addr**- адрес, тип Alpha, размер 20;

• **Chief**- руководитель, тип Alpha, размер 20.

Таблица 2. Типы полей формата Paradox.

Alpha	Строка длиной 1-255 байт, содержащая печатаемые символы.
Number	Числовое поле длиной 8 байт, значение которого может

	быть положительным и отрицательным. Диапазон чисел – от 10^{-308} до 10^{308} с 15 значащими цифрами.
$ (Money)	Числовое поле, значение которого может быть положительным и отрицательным. По умолчанию, является форматированным для отображения десятичной точки и денежного знака.
Short	Числовое поле длиной 2 байта, которое может содержать только целые числа в диапазоне от –32768 до 32767.
Long Integer	Числовое поле длиной 4 байта, которое может содержать целые числа в диапазоне от –2147483648 до 2147483648.
# (BCD)	Числовое поле, содержащее данные в формате BCD (Binary Coded Decimal). Скорость вычислений немного меньше, чем во многих числовых форматах, однако точность – гораздо выше. Может иметь 0-32 цифр после десятичной точки.
Date	Поле даты длиной 4 байта, которое может содержать дату от 1 января 9999 г. до нашей эры – до 31 декабря 9999 г. нашей эры. Корректно обрабатывает високосные года и имеет встроенный механизм проверки правильности даты.
Time	Поле длиной 4 байта, содержит время в миллисекундах от полуночи. Ограничено 24 часами.
(Timestamp)	Обобщенное поле длиной 8 байт – содержит и дату и время.
Memo	Поле для хранения символов, суммарная длина которых более 255 байт. Может иметь любую длину. При этом размер, указываемый при создании таблицы, означает количество символов, сохраняемых в таблице (1-240) – остальные символы сохраняются в отдельном файле с

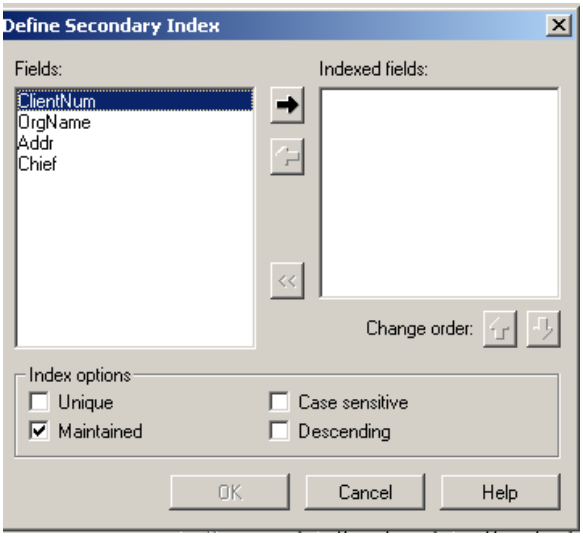
	расширением MB.
Formatted Memo	Поле, аналогичное Memo, с добавлением возможности задавать шрифт текста. Также может иметь любую длину. При этом размер, указываемый при создании таблицы, означает количество символов, сохраняемых в таблице (0-240) - остальные символы сохраняются в отдельном файле с расширением .MB. Delphi в стандартной поставке не обладает возможностью работать с полями типа Formatted Memo.
Graphic	Поле, содержащее графическую информацию. Может иметь любую длину. Смысл размера - такой же, как и в Formatted Memo. Database Desktop «умеет» создавать поля типа Graphic, однако наполнять их можно только в приложении.
OLE	Поле, содержащее OLE-данные (Object Linking and Embedding) - образы, звук, видео, документы - которые для своей обработки вызывают создавшее их приложение. Может иметь любую длину. Смысл размера - такой же, как и в Formatted Memo. Database Desktop «умеет» создавать поля типа OLE, однако наполнять их можно только в приложении. Delphi «напрямую» не умеет работать с OLE-полями, но это легко обходится путем использования потоков.
Logical	Поле длиной 1 байт, которое может содержать только два значения - **T** (true, истина) или **F** (false, ложь). Допускаются и строчные и прописные буквы.

+ - (Autoincrement)	Поле длиной 4 байта, содержащее нередактируемое (read only) значение типа *long integer*. Значение этого поля автоматически увеличивается (начиная с 1) с шагом 1, что очень удобно для создания уникального идентификатора записи. Заметим, что физический номер записи не может служить ее идентификатором, т.к. в Paradox он не используется. В InterBase отсутствуют как физические номера записей, так и поля типа Autoincrement. Для создания идентификатора здесь используется встроенная функция *Gen_jd.*
Binary	Поле, содержащее любую двоичную информацию. Может иметь любую длину. При этом размер, указываемый при создании таблицы, означает количество символов, сохраняемых в таблице (0-240) - остальные символы сохраняются в отдельном файле с расширением .MB. Это полнейший аналог поля BLOb в InterBase.
Bytes	Строка цифр длиной 1-255 байт, содержащая любые данные.

7. Задать простой или составной первичный индекс (primary index) двойным щелчком мыши в колонке Key. Ключевое поле - **ClientNum.**

8. Для Paradox'а обязательно наличие вторичных индексов для полей, которые предполагается использовать при сортировке, поиске данных или связывании таблиц. Вторичный индекс создается командой **Table Properties/ Secondary indexes/ Define.** Появится окно, в котором выбираются поля, образующие ключ индекса, и нажатием кнопки со стрелкой перемещаются в окно Indexed fields. После нажатия **ОК** введите имя индекса. Оно будет использоваться в дальнейшем при обращении к индексу. Имя должно быть уникальным, не совпадающим с именами полей и других

индексов. Создайте вторичные индексы для каждого неключевого поля таблицы.



9. Завершение создания. Сохраните созданную таблицу в каталоге базы данных: команда **Save As** внизу окна, задайте имя таблицы (**Clients).** Если вы отметите опцию **Display Table** внизу окна, то таблица автоматически откроется после сохранения, и вы сможете просматривать ее содержимое (команда **Table/View data** включается по умолчанию) и редактировать (команда **Table/Edit data).** К сожалению, Database Desktop не настраивается на русский язык, и все вводимое русскими буквами выглядит абракадаброй. Но при использовании таблицы все надписи показываются правильно.

Введите в таблицу несколько записей.

10. Повторите пункты 1 – 9 для построения таблицы ДОГОВОРА.

Имя таблицы – **Contracts.**

Поля:

- **ContractNum** – номер договора, тип Autoincrement, ключевое поле;
- **ClientNum** - номер клиента, тип Long Integer;
- **Date** - дата заключения договора, тип Date;
- **Sum** - сумма, тип Number.

11. После создания таблиц просмотрите каталог БД и основные типы поддерживающих файлов**.**