

Тема 20.Создание приложения со связанными таблицами

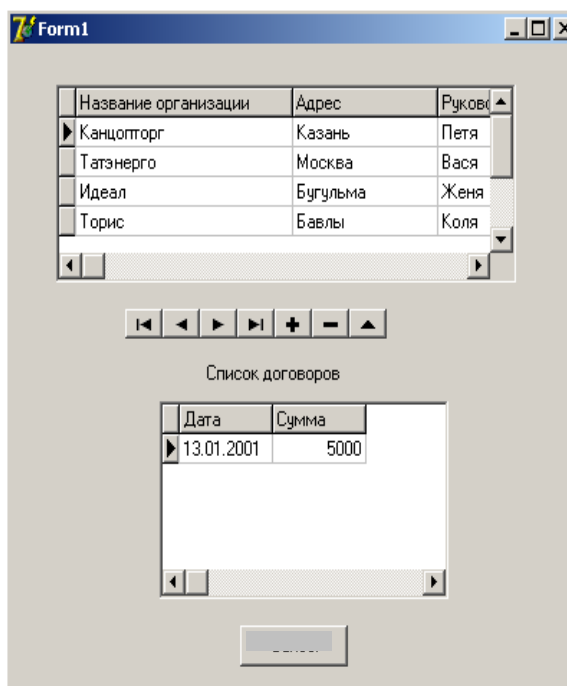
Практические задания

Задание 1. Создание двухтабличного приложения со связанными таблицами

Теперь посмотрим, как строить приложения со связанными друг с другом таблицами.

Две таблицы могут быть связаны друг с другом по ключу. Одна из этих связанных таблиц является головной (master), а другая - подчиненной, детализирующей (detail). Построим приложение, которое при перемещении по таблице **КЛИЕНТЫ** показывало бы список договоров, заключенных с выбранной организацией. В этом случае головной таблицей является таблица **КЛИЕНТЫ**, а подчиненной - таблица **ДОГОВОРА**.

1. Создайте новый проект, разместите по два компонента **Table** и **DataSource** и настройте их соответственно на таблицы **КЛИЕНТЫ** (Clients) и **ДОГОВОРА** (Contracts). Настройте имена компонентов, чтобы в дальнейшем было удобнее ссылаться на них.



2. Разместите два компонента **DBGrid**, настройте их на соответствующие таблицы и настройте выводимые имена полей в сетке с

помощью редактора колонок. Сделайте невидимыми поля **ContractNum** и **ClientNum** в таблице **КЛИЕНТЫ**.

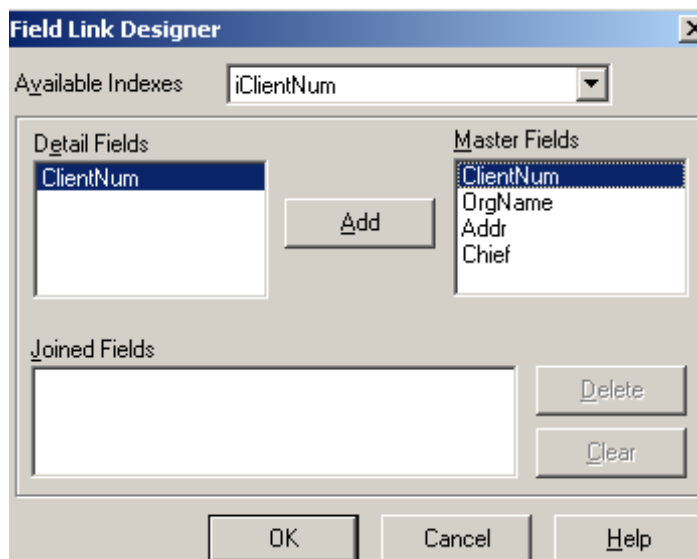
3. Добавьте подписи к таблицам при помощи компонента **Label**.

4. Разместите компонент **DBNavigator**, свяжите его с таблицей **КЛИЕНТЫ** и настройте как в предыдущем задании.

5. Сделаем таблицу **ДОГОВОРА** подчиненной по отношению к таблице **КЛИЕНТЫ**. Для этого свойство **MasterSource** компонента **Table** для таблицы **ДОГОВОРА** установить в **DataSource**, соответствующий таблице **КЛИЕНТЫ**.

6. Теперь надо задать, по какому полю связать таблицы. Щелкните на кнопку в правой части свойства **MasterFields** компонента **Table** для таблицы **ДОГОВОРА**. Запустится конструктор связей **Field Link Designer**. В нем слева в окне **DetailFields** расположены имена полей подчиненной таблицы, но только тех, по которым таблица индексирована. Слева в окне **MasterFields** расположены поля головной таблицы. В списке **AvailableIndexes** выберете вторичный индекс для поля-связки **ClientNum** и свяжите поля в master и detail таблицах по кнопке **Add**.

7. Запустите приложение. Проверьте его работу при добавлении и удалении записей в главной и подчиненной таблицах. Вы увидите, что в зависимости от того, какую запись вы выделяете в списке организаций, отображается список договоров с этой организацией. Подчиненная таблица отражает только те записи, в которых ключевые поля совпадают с ключевыми полями головной таблицы.



Задание 2. Поиск и сортировка данных

Добавим в приложение возможность сортировки и поиска. Для индексированных полей существует возможность инкрементального поиска с помощью метода **FindNearest** и обычного компонента-поля ввода **Edit**.

Таблица должна быть предварительно индексирована по тому полю, по которому будет производиться поиск (помните вторичные индексы?).

1. Настроим поиск по имени организации. Для ввода искомых значений разместите на форме компонент **Edit**.

2. Создайте обработчик события **OnChange**:

```
Table.IndexName:='Org';
```

```
Table .FindNearest([Edit1 .Text]);
```

Table - это компонент **Table**, связанный с таблицей **КЛИЕНТЫ** (у вас может быть другое название). Свойство **IndexName** задает индекс, который будет использоваться для поиска. В этом примере индекс для поля с названием организации (**OrgName**) называется **Org** (у вас же может быть другое название). Оператор **FindNearest** обеспечивает переход к строке с наиболее совпадающим значением (по первым буквам), если полного совпадения нет.

The image shows two screenshots of a Windows application interface. The top screenshot displays a table with the following data:

Название организации	Адрес	Руководитель
Татэнерго	Москва	Вася
Идеал	Бугульма	Женя
Торис	Бавлы	Коля
Канцопторг	Казань	Петя

The bottom screenshot shows the same table, but with search input fields and a 'Поиск' button. The search results are displayed in a separate table:

Дата	Сумма
04.05.1999	6000

1. Добавьте еще два компонента **Edit** и создайте соответствующие обработчики, для поиска по полям с адресом и фамилией руководителя.

4. Добавьте подписи к компонентам-редакторам.

5. Запустите приложение и проверьте поиск по разным полям. Обратите внимание: то поле, по которому ведется поиск (чей индекс выставлен в свойстве **IndexName**), автоматически сортируется.

Теперь остановимся на методе **Locate**. В простейшем случае применение этого метода отличается только отсутствием необходимости индексировать набор данных (вторичных индексов).

6. Добавим в приложение возможность поиска договоров по дате. Разместите на форме компонент **Edit**, подпись к нему и кнопку. Для кнопки создайте обработчик нажатия и введите туда строку:

```
Table2.Locate('Date',StrToDate(Edit4.Text),[loCaseInsensitive]);
```

Первый параметр - строка, содержащая список полей, по которым производится поиск. Второй параметр - массив искомых значений. Третий параметр может принимать два значения: **loCaseInsensitive** - нечувствительность поиска к регистру, в котором введены символы, и **loPartialKey** -допустимость частичного совпадения.