

#### 4. Создание проекта, базы данных и таблиц

При создании приложения используется проект, объединяющий элементы приложения Visual FoxPro и группирует их по типам. Информация о проекте хранится в специальной таблице, что, в отличие от обычных таблиц Visual FoxPro, имеет расширение PJX. Использование проекта упрощает разработку приложения, так как в проекте базы данных, программы, формы, отчеты, запросы и другие элементы приложения располагаются в соответствующих разделах, а также запоминается расположение каждого включенного в проект элемента.

Создав проект и определив входящие в него элементы, можно использовать его для сборки приложения, построив файл с расширением APP, или для создания файла, выполняющего, с расширением EXE. При построении приложения из проекта осуществляется поиск и сборка файлов, на которые ссылаются элементы приложения, отслеживаются версии файлов, входящих в проект.

База данных является частью проекта, поэтому ее целесообразно создавать в окне проекта. Файл базы данных и файлы таблиц располагаются в папке DATA соответствующего проекта. В окне конструктора базы данных в графическом виде отображается структура базы данных. В нем можно создавать новые таблицы, модифицировать существующие, создавать для них индексы, устанавливать связи между таблицами, задавать правила для обеспечения ссылочной целостности в соответствии с диаграммой «таблица-связь».

Создание и модификация структуры таблиц осуществляется с помощью конструктора таблиц.

На вкладке Fields делается описание полей таблицы в соответствии с диаграммой «таблица-связь» - имя, тип, размер, возможность размещения в поле неопределенного значения (NULL). В области Field validation - задается

правило для проверки достоверности данных уровня поля (ограничения, которые налагает пользователь).

На вкладке Indexes определяются индексные выражения простых и сложных индексов, назначается имя тега и тип индекса.

На вкладке Table задаются правила проверки достоверности данных уровня записи, триггеры ссылочной целостности, текст которых генерируется автоматически.

Правила проверки достоверности данных уровня поля и уровня записи отличаются по времени возникновения события:

- Событие проверки правила уровня поля возникает при попытке покинуть указанное поле;

- Событие проверки правила уровня записи возникает при попытке переместиться на другую строку. При перемещении по полям строки это событие не возникает. Таким образом, в этих правилах можно сравнивать значения нескольких полей. Например, для задачи отдела кадров можно установить правило, чтобы значение в поле «Дата окончания работы» было больше значения в поле «Дата приема на работу».

В Visual FoxPro нет режима «установления» в правилах для обеспечения ссылочной целостности данных. С одной стороны, этот режим предусматривает установление значений внешнего ключа в неопределенное значение. Кроме того, разработчик таблиц имеет возможность запретить появление значений NULL в определенных полях. Если это будет запрещено в поле внешнего ключа, то перед СУБД будет поставлена неразрешимая задача. Поэтому при настройке ссылочной целостности, этот тип правил интерпретируется, как Ignore. Visual FoxPro в этом случае за целостность данных по ссылкам не следит, вся ответственность возлагается на программиста.

## 5. Экранные формы

Экранные формы используются для ввода данных в таблице – это визуальный класс типа контейнер. В контейнере можно размещать любое количество объектов. Форма имеет более 100 свойств и 60 методов с возможностью добавления пользователем дополнительных свойств и методов.

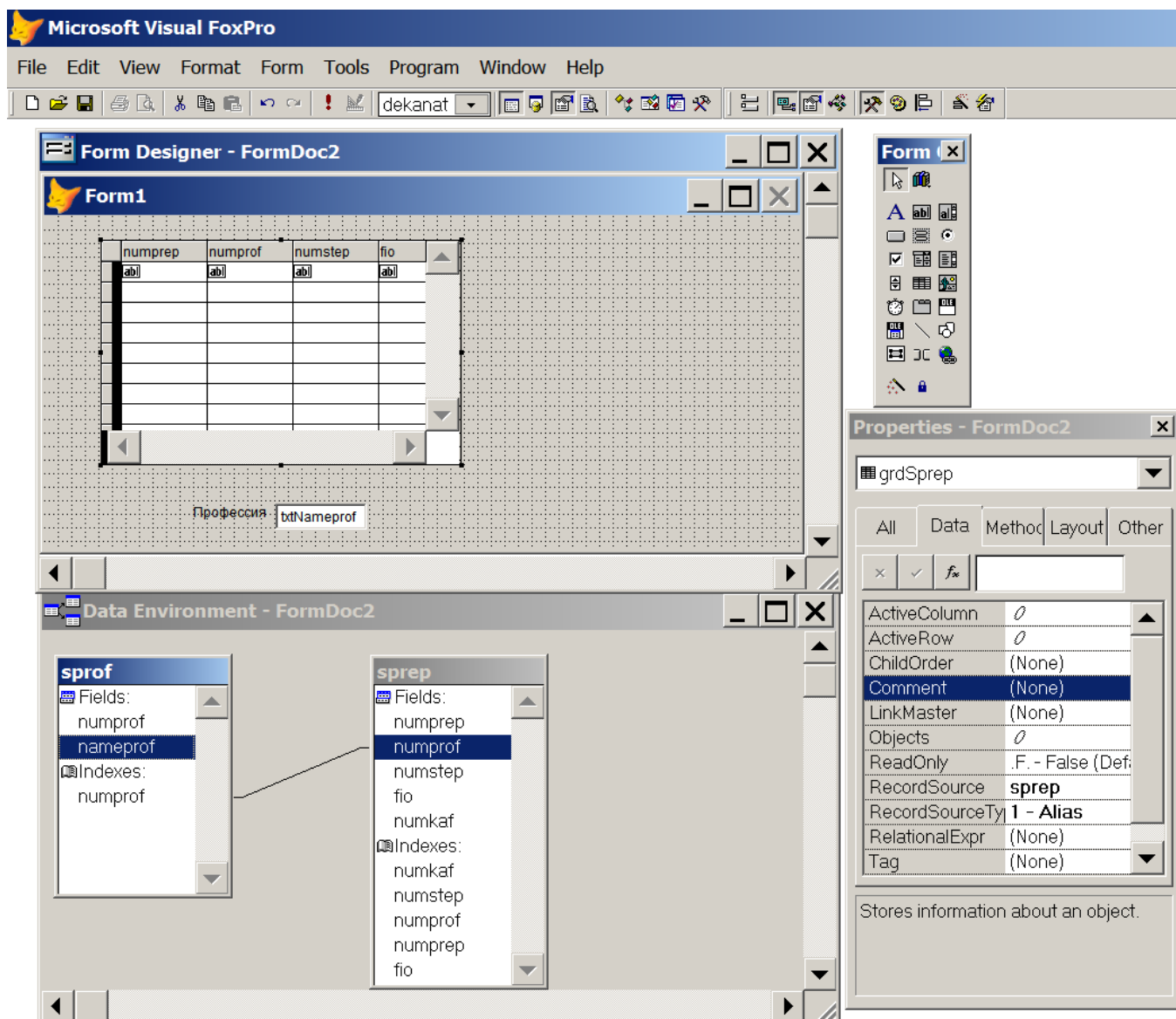


Рис.5.1

Новые формы для ввода данных создаются в Менеджере проекта на вкладке DOCUMENTS.

**Источником данных** для формы обычно являются таблицы. Из меню **Forms** или контекстного меню вызывается дочерний объект **DATA ENVIRONMENT**, в котором определяется графическая схема среды данных. В этом объекте можно:

- добавлять таблицы, представления, курсоры;
- устанавливать отношения между таблицами.

Все источники данных добавляются в статусе объекта Cursor, причем, он имеет свой набор свойств, событий и методов. Для курсора можно определить управляющий индекс, фильтр, права доступа, способ буферизации и т.п.

Для организации отношения между двумя таблицами необходимо выделить поле отношения в родительской таблице и перетащить его на соответствующий индекс или поле в дочерней таблице. Если индекса по этому полю нет, он будет создан. Если при создании базы данных **между таблицами определены постоянные отношения** для определения ограничений по ссылочной целостности, все эти отношения будут автоматически установлены при добавлении таблиц. Для правильного отображения данных на форме рекомендуется все эти отношения разрывать, и устанавливать их в направлении от родительской таблицы к дочерней. В предлагаемых постоянных отношениях зачастую родительские таблицы другие.

На рис. 5.1 приведен вид окна FoxPro проектирования формы. При проектировании из объекта DATA ENVIRONMENT можно вставить в форму отдельное поле таблицы. Для этого необходимо выделить щелчком мыши необходимое поле таблицы, перетащить его на макет формы и опустить. В результате будет создан редактор для поля и текстовая метка с именем поля. Если выделить заголовок таблицы и перетащить его на макет формы, будет создан элемент управления Grid для этой таблицы. Для настройки вида формы и отдельных элементов управления используется BUILDER, вызываемый одним нажатием правой кнопки по объекту на форме, или ручная настройка свойств в окне PROPERTIES.

#### **В окне PROPERTIES:**

Если в свойстве элемента управления **ControlSource** указать имя поля таблицы источника данных, такой элемент управления будет называться

**связанным**, все изменения в нем будут синхронизироваться с данными в таблице.

**Свойство Bound** - Определяет, привязывается ли элемент управления в объекте Column к источнику объекта. Доступно на этапе разработки и на этапе выполнения.

Если свойство Bound объекта Column имеет значение "истина" (.T.), установка ControlSource данного объекта будет применена к столбцу и ко всем содержащимся в нем элементам управления. Если попытаться установить свойство ControlSource для какого-либо элемента управления, содержащегося в столбце, произойдет ошибка. Если свойство Bound объекта Column имеет значение "ложь" (.F.), можно установить свойство ControlSource для элемента управления, содержащегося в столбце, непосредственно. Если после этого установить значение свойства ControlSource для Column, оно отменит действие установки ControlSource для данного элемента управления. В следующей таблице показано, к каким типам данных обычно привязываются те или иные элементы управления.

Базовый класс элемента управления

объекта Column	Распространенные типы данных Column
CheckBox	Logical, Numeric
ComboBox	Character, Numeric
CommandButton	Character, Numeric
EditBox	Character
ListBox	Character, Numeric
OptionButton	Character, Numeric
Spinner	Currency, Numeric
TextBox	Любой тип данных, кроме General и Memo

Замечание Нельзя привязать элемент управления к объекту Column, тип данных которого General или Memo.

**Свойство DataSession** Определяет, может ли объект Form, FormSet или ToolBar выполняться в собственном сеансе и иметь отдельную среду данных.

Принимаемые значения:

**DEFAULT** – общие настройки, включая положение указателя в таблице.

**PRIVATE** – для каждой формы сохраняются свои настройки.

Установленные свойства в окне **PROPERTIES** выделяются различными шрифтами:

- **курсив** – свойства базового класса, только для просмотра;
- **простой** – свойства, унаследованные от базового класса, устанавливаются по умолчанию;
- **полужирный** – свойства, измененные разработчиком. Если после изменения свойства, вернуться к старому значению (по умолчанию), шрифт не изменится. Это свидетельствует о том, что связь свойства с базовым классом разорвана. Изменение этого свойства в базовом классе не будет синхронизировано со свойством в форме. Для восстановления связи с базовым классом необходимо использовать пункт **Reset to Default** из контекстного меню.

По щелчку правой клавиши мыши на свободном поле окна **PROPERTIES** вызывается меню, в котором по пункту **Non\_Default Properties Only** скрываются все свойства, установленные по умолчанию. Это очень удобно, если при изучении методов конструирования форм необходимо разобраться, каким образом был настроен элемент управления.

Форма сохраняется в файле с расширением **FRM**. Созданные формы надо сохранять в свою отдельную папку. Вызов формы на выполнение осуществляется командой **DO FORM**.

**DO FORM my\_form.**

Окно формы может открываться в двух режимах:

- **Modeless** –немодальный режим – можно переключаться между различными окнами;
- **Modal** – модальный режим – переключиться на другое окно возможно только после закрытия модального окна. Такое окно применяется в том случае, если

от пользователя необходимо немедленно получить ответ на какой-либо вопрос, или ввести данные, например ввод пароля.

Тип окна задается в свойстве формы **WindowType**.

С помощью формы можно вернуть значение в переменную. Для этого форма вызывается командой **DO FORM my\_form to cNAME**

При закрытии формы командой **Release** передача значения выполняется в операторе **RETURN** события **UNLOAD**. В RETURN надо указать выражение, которое должно возвратиться. Чаще всего это значение добавленного программистом свойства формы.

### **Формы с поддержкой справочника**

В справочнике преподавателей заносятся код и ФИО преподавателя. Для должности, степени и кафедры введены внешние ключи, по которым из соответствующих таблиц (справочников) подбираются наименования.

В таблицу информация вводится в следующем виде:



Numprep	Numprof	Numstep	Fio	Numkaf
1	5	1	Баркалов А.А.	1
2	5	1	Лапко В.В.	1
3	3	1	Баркалов А.А.	1
4	2	2	Краснокутский В.А.	1
5	2	2	Гусев Б.С.	1
6	2	2	Федяев О.И.	2
7	9	2	Дорожко Л.И.	1
8	2	2	Павлыш В.Н.	3
9	3	3	Зеленева И.Я.	1
10	9	3	Молдаванов А.В.	1

Рис. 5.2

В форме ввода информация представляется в следующем виде:

Код	Фамилия И.О.	Должность	Степень	Кафедра
1	Баркалов А.А.	Профессор	д.т.н.	Эвм
2	Лапко В.В.	Ассистент	д.т.н.	Эвм
3	Баркалов А.А.	Ассистент	д.т.н.	Эвм
4	Краснокутский В.А.	Доцент	к.т.н.	Эвм
5	Гусев Б.С.	Доцент	к.т.н.	Эвм
6	Федяев О.И.	Доцент	к.т.н.	ПО
7	Дорожко Л.И.	Ст.преп.	к.т.н.	Эвм
8	Павлыш В.Н.	Доцент	к.т.н.	ПМИ
9	Зеленева И.Я.	Ассистент	нет	Эвм
10	Медведева А.В.	Ст.преп.	нет	Эвм

Рис. 5.3

Должность, степень и кафедра представляются на форме не кодами, а наименованиями и должны выбираться из списка, сформированного из соответствующих справочников.

Для создания такой формы в объект Data Environment переносятся все таблицы, являющиеся источниками данных для нее. Постоянные связи, предложенные FoxPro, удаляются и устанавливаются новые от родительской таблицы SPREP к дочерним (справочникам).

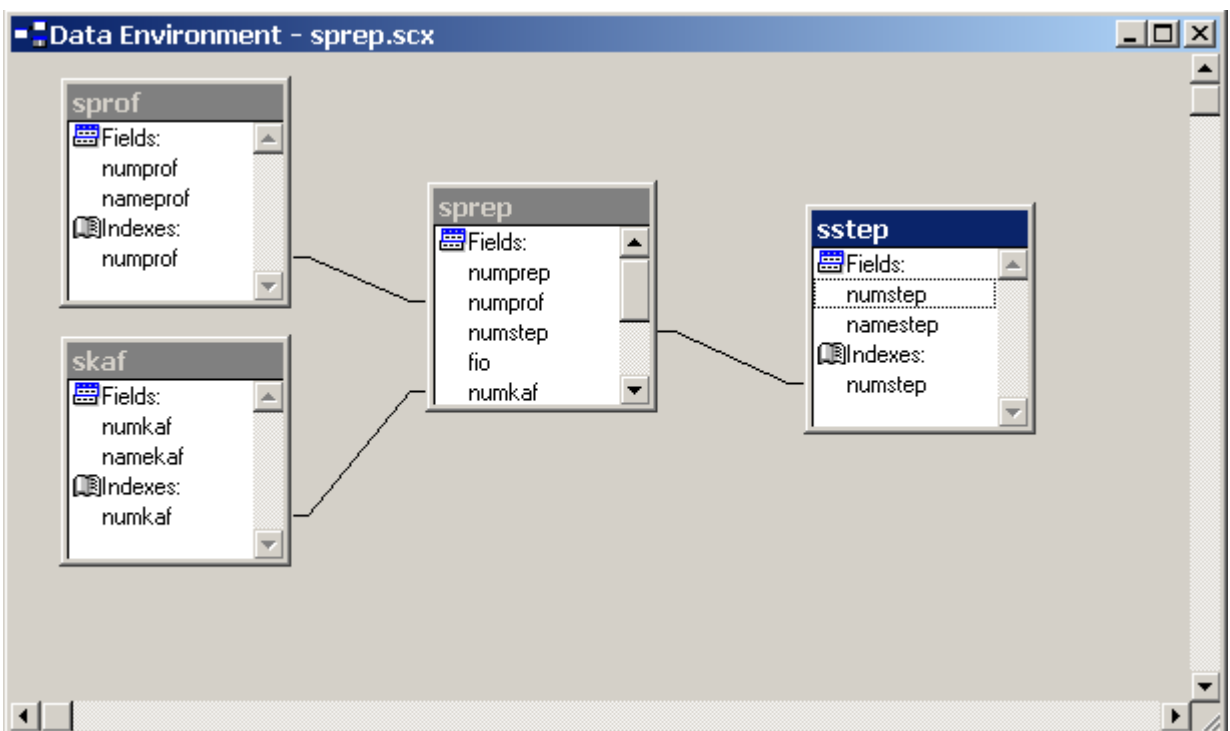


Рис. 5.4



Далее на поле формы создается GRID, в которой в столбцах должность, кафедра и степень в свойстве DataSource вместо кодов из родительской таблицы, размещаются наименования из дочерних таблиц справочников. После чего в этих столбцах размещаются элементы управления ComboBox, которые настраиваются на формирование списка из соответствующей таблицы справочника, и осуществляется их связывание с полями родительской таблицы, в которые будут записываться коды выбранных из справочника наименований.