

ЛАБОРАТОРНАЯ РАБОТА №3

РАБОТА С ПОЛЯМИ – КОМПОНЕНТ TFIELD

Цель:

- 1) Провести ознакомление с работой с полями – компонентом TField.
- 2) Показать работу со свойствами полей в визуальных компонентах.
- 3) Создать вычисляемое поле.
- 4) Создать поле выбора данных (lookup поле).
- 5) Установить связь между набором данных в приложении.

Работа выполняется в приложении для работы с базой данных для своей предметной области (на примере базы данных для кафе).

Краткие теоретические сведения

Компонент TField позволяет обращаться к полям наборов данных.

Каждый набор данных состоит из записей, а те, в свою очередь, состоят из полей. Таким образом, в составе записи имеется минимум одно поле.

В C++Builder имеется возможность использовать при работе с НД или все поля, определенные в данной таблице БД на текущий момент, или использовать только часть существующих полей, или проводить добавление вычисляемых полей, полей из других наборов данных.

Использование только части имеющихся полей дает то преимущество, что значения неиспользуемых полей не могут быть изменены ни вследствие алгоритмической ошибки, ни вследствие злонамеренного умысла, поскольку неиспользуемые поля считаются неизвестными.

Существует два способа задания состава полей для НД.

Первый способ состоит в том, что после создания НД не предпринимается никаких дополнительных действий по уточнению состава полей. Тогда будет разрешен доступ ко всем полям, определенным в данный момент в наборе данных.

К полю в этом случае можно обращаться с помощью метода (функции) FieldByName компонентов НД

function FieldByName(const имя поля AnsiString);

или через свойство указанных компонентов **Fields [Index]**, которое возвращает указатель на тип TField. Подробнее об этом свойстве будет сказано ниже.

Второй способ определения состава полей заключается в том, что для данного НД поля как компоненты TField добавляются в форму с помощью редактора полей. О нем будет сказано чуть ниже.

К таким полям можно обращаться через его имя, определяемое в свойстве Name компонента TField, соответствующего данному полю.

Компонент TField, как будет показано ниже, обладает рядом свойств, методов и событий, обращаться к которым следует через указание имени компонента TField и имени свойства, метода или обработчика события.

Когда поле набора данных определено в форме в качестве экземпляра компонента TField к нему можно обращаться, например, по имени (содержащемуся в свойстве Name), а также через метод (функцию) НД FieldByName и свойство Fields [Index].

К "несуществующим" полям обратиться из данного НД нельзя никак. Такие попытки будут возбуждать исключительные ситуации с сообщением «Field <имя> not found».

Таким образом, обратиться к полю можно:

1. Если полю соответствует компонент TField через редактор полей - через имя данного компонента, которое определяется свойством Name.
2. Используя метод (функцию) FieldByName (“ИмяПоля”) набора данных.
3. Используя свойство Fields[индекс] набора данных,
Индекс является порядковым номером поля в определении таблицы БД. Отсчет идет от 0.

Компоненты TField различаются по типу данных.

Иерархия компонентов - полей такова:

TFeld

TVlobField	большой двоичный объект
------------	-------------------------

TGraphicField	графическое поле (работает с содержимым BLOB-поля как с графическим изображением)
---------------	---

ТMemoField мемо-поле (интерпретирует BLOB-поле как большой текст)

TBooleanField логическое поле

TBinaryField не типизированное двоичное поле

TByteField поле для хранения байтовых значений фиксированной длины

TVarBytesField поле для хранения байтовых значений переменной длины

DateTimeField поле для хранения даты и времени

TDateField поле для хранения только даты

TTimeField поле для хранения только времени

TNumericField поле для хранения числовых значений

TBCDField ВCD-значений

TFloatField значений с плавающей точкой

TCurrencyField в том числе в денежном формате

TIntegerField	целочисленных значений
TAutoIncField	в том числе автоинкрементных
TSmallIntField	в том числе коротких целых
TWordField	в том числе в формате беззнакового длинного целого
TStringField	поле для хранения строковых значений.

Обращение к значению поля

К значению поля можно обратиться, например, при помощи свойств:

1) свойство (property) **Value** (тип Variant);

2) свойство **AsNNN** (свойство приведения типов полей):

- property **AsBoolean** - числовые значения приводятся к типу Boolean, если содержат 0 (false) или 1 (true). Символьные значения - если содержат в качестве первого символа "Y", "y", "T" или "t" (или "Yes" или "True"), и false во всех иных случаях.

- property **AsCurrency** - для приведения к денежному типу (к типу Currency).

- property **AsDateTime** - для приведения к типу TDateTime значений TDateField, TDateTimeField и TTimeField, а также для приведения к типу TDateTime строковых значений, находящихся в соответствующем формате.

- property **AsFloat** - служит для приведения к типу Double значений полей TFloatField, TBCDField и TCurrencyField.

- property **AsInteger** - служит для приведения к типу Longint полей типа TIntegerField, TSmallIntField и TWordField. Для полей типа TStringField преобразование к Longint выполняется, если оно возможно.

- property **AsString** - служит для приведения к строковому типу.

- property **AsVariant** - служит для приведения к типу Variant.

В таблице 1 показана совместимость значений полей разных типов.

Используемые обозначения:

= типы равнозначны;

+ преобразование возможно;

+RI преобразование возможно, округление до ближайшего целого;

? преобразование происходит, если возможно; часто зависит от формата показа (свойство DisplayFormat);

x преобразование не разрешено;

мето имеет значение для мето-поля.

Таблица 1 - Совместимость значений полей разных типов

Тип поля	AsString	AsInteger	AsFloat	AsDateTime	AsBoolean
TStringField	=	?	?	?	?
TIntegerField	+	=	+	x	x
TSmallIntField	+	=	+	x	x
TWordField	+	=	+	x	x
TFloatField	+	+RI	=	x	x
TCurrencyField	+	+RI	=	x	x
TBCDField	+	+RI	=	x	x
TDateTimeField	+	x	Преобразование даты к числу дней с 01.01.0001	=	x
TDateField	?	x	Преобразование даты к числу дней с 01.01.0001	=	x
TTimeField	?	x	Преобразование времени делением на 24 часа	=	x
TBooleanField	В строку «True» или «False»	x	x	x	=
TBytesField	+	x	x	x	x
TVarBytesField	+	x	x	x	x
TBlobField	memo				
TMemoField	memo				
TGraphicField	memo				

Особенности показа полей в визуальных компонентах во время выполнения

Свойства property DisplayLabel и DisplayName

DisplayLabel используется, чтобы назначать заголовки столбцов, например, в компоненте TDBGrid. Заголовки TDBGrid используют свойство DisplayName полей, чьи величины они представляют. Установка DisplayLabel изменяет только для чтения свойство DisplayName от FieldName до строки определенной как DisplayLabel.

Свойство property Visible

Свойство Visible применяется для того, чтобы скрывать столбцы, например, в компоненте TDBGrid.

Если свойство Visible установлено в false, то поле становится невидимым, например, в компоненте TDBGrid.

Свойство property DisplayWidth

DisplayWidth определяет ширину видимой части поля, например, в компоненте TDBGrid.

Свойство property ReadOnly

Если свойство ReadOnly установлено в значение true, то значения в поле доступны «только для чтения».

Свойство property Required

Если свойство Required установлено в значение true, то значения в поле должны быть обязательно заданы.

Другие свойства, события будут рассмотрены в ходе дальнейшего выполнения лабораторных работ

Проверка введенного в поле значения

Свойство IsNull

Свойство **IsNull** во время выполнения возвращает true, если поле содержит пустое значение.

Значение поля по умолчанию и ограничения на значение поля

Значение поля по умолчанию можно установить при помощи свойства **property DefaultExpression**.

В случае указания значений, отличных от целочисленного, они должны заключаться в кавычки.

property CustomConstraint – позволяет задавать ограничение на значения в поле. Является строковым типом и должно заключаться в кавычки.

property ConstraintErrorMessage позволяет указать сообщение об ошибке, выдаваемое пользователю в случае, если введенное значение поля не удовлетворяет ограничению, указанному в свойстве CustomConstraint.

И другие.

Задание 1. Использование редактора полей для организации компонентов TField

Для того, чтобы определить один или несколько компонентов TField нужно:

1. Выбрать необходимый НД.
2. Нажать правую кнопку мыши.
3. Во всплывающем меню выбрать режим Fields Editor (запуская тем самым редактор полей).
4. Или вместо пунктов 2 и 3 два раза щелкнуть мышью по выбранному набору данных (запуская тем самым редактор полей).
5. Вновь нажать правую кнопку мыши и во всплывающем меню выбрать Add Fields.

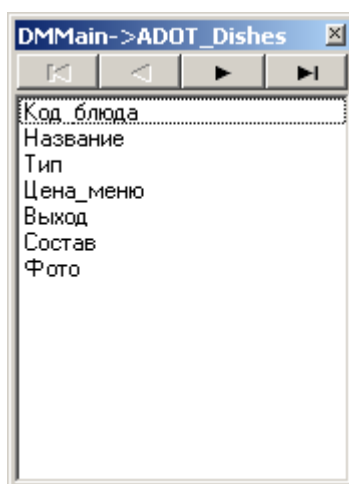
6. В появившемся списке полей таблицы БД или полей, участвующие запросе, выбрать необходимые.

7. Если требуется добавить в редактор полей все поля таблицы БД, то нажать Add all Fields.

8. Нажать ОК. Для каждого из указанных полей будет создан компонент TField.

9. Если необходимо изменить свойства конкретного поля или написать обработчик для какого-либо события, необходимо в редакторе полей выбрать нужное поле и, используя инспектор объектов, установить значение в свойство или определить обработчик события.

- *Перейдите на модуль данных DMMain.*
- *Добавьте в редактор полей набора данных ADOT_Dishes все поля из таблицы Блюда.*



- *Выделите поле Состав в редакторе полей. Укажите для него в Object Inspector свойство Visible (видимо) в false, так как это поле показывается отдельно в компоненте DBМетод.*

- *Выделите поле Фото в редакторе полей. Укажите для него в Object Inspector свойство Visible (видимо) в false, так как это поле показывается отдельно в компоненте DBImage.*

- *Добавьте в редактор полей набора данных ADOT_Sotrudniki все поля из таблицы Сотрудники.*

- *Добавьте в редактор полей набора данных ADOT_Sales все поля из таблицы Продажи.*

- *Добавьте в редактор полей набора данных ADOT_DisheSale все поля из таблицы Продано.*

Задание 2. Обращение к значениям в полях. Создание вычисляемых полей

Если необходимо создать вычисляемое поле, значение которого вычисляется по значениям других полей, поступают так:

1. В редакторе полей необходимо создать новое поле, пометив его как поле Calculated. Для этого нужно сделать текущим (при помощи мыши) необходимый НД, нажать правую кнопку мыши, выбрать в меню Fields Editor (или двойное нажатие мышью) и снова нажать правую кнопку мыши и выбрать в меню New Field. Затем в окне диалога необходимо указать имя поля, его тип и для строковых полей длину.

Для нового поля будет создан компонент TField, доступ к которому отныне можно осуществлять в редакторе полей.

2. Для компонента НД, к которому принадлежит вычисляемое поле, необходимо определить обработчик события OnCalcFields.

Событие OnCalcFields возникает всякий раз, когда курсор (указатель записи) перемещается в НД от записи к записи. Это событие возникает и при инициализации НД (после открытия), а также после фильтрации записей в НД, что, впрочем, также связано с изменением положения указателя записи.

Процедура - обработчик события OnCalcFields содержит реализацию алгоритма вычисления значения вычисляемого поля или группы полей.

Необходимо помнить, что в этом обработчике значение может быть присвоено только вычисляемому полю и не может - полю, определенному в структуре таблицы БД.

- *Создайте вычисляемое поле ФИО в наборе данных ADOT_Sotrudniki.*

- *Значения в поле ФИО будем формировать по значению кода сотрудника, пробел, фамилия, пробел, имя, пробел, отчество, типе, должность.*

- *Выделите компонент ADOT_Sotrudniki на модуле данных DMMain.*

- *Установите у компонента ADOT_Sotrudniki свойство Active в false, так как сейчас будем менять набор данных.*

- *Два раза щелкните мышью по набору данных ADOT_Sotrudniki (запуская тем самым редактор полей).*

- *В редакторе полей ADOT_Sotrudniki нажмите правую кнопку мыши и выберите New Field. Затем в окне диалога укажите для нового поля:*

- *в Field Properties:*

- *Name в ФИО;*

- *Type укажите в string;*

- *Size укажите в 50;*

- *в Field type вычисляемое поле Calculated.*

- *Нажмите Ок.*

- *Закройте редактор полей ADOT_Sotrudniki.*

- *Выделите компонент ADOT_Sotrudniki. Для него через Object Inspector перейдите на закладку Events и нажмите два раза мышью на событии OnCalcFields, для которого запишите:*

```
ADOT_Sotrudniki->FieldByName("ФИО")->Value=  
ADOT_Sotrudniki->FieldByName("Код_сотрудника")->AsString+  
" "+ADOT_Sotrudniki->FieldByName("Фамилия")->AsString+" "+
```

```
ADOT_Sotrudniki->FieldByName("Имя")->AsString+" "+  
ADOT_Sotrudniki->FieldByName("Отчество")->AsString+"-"  
+ADOT_Sotrudniki->FieldByName("Должность")->AsString;
```

- Установите у компонента *ADOT_Sotrudniki* свойство *Active* в *true* для подключения таблицы БД Сотрудники.

- Нажмите F9.

- Посмотрите в *DBGrid*, как вычисляется поле ФИО при работе с сотрудниками.

- Создайте вычисляемое поле *Сумма* в наборе данных *ADOT_DisheSale*.

- Будем считать сумму с учетом скидки: $\text{цена} * \text{количество} * (1 - \text{скидка})$.

- Выделите компонент *ADOT_DisheSale* на модуле данных *DMMain*.

- Установите у компонента *ADOT_DisheSale* свойство *Active* в *false*, так как сейчас будем менять набор данных.

- Два раза щелкните мышью по набору данных *ADOT_DisheSale* (запуская тем самым редактор полей).

- В редакторе полей *ADOT_DisheSale* нажмите правую кнопку мыши и выберите *New Field*. Затем в окне диалога укажите для нового поля:

- в *Field Properties*:

- *Name* в *Сумма*;

- *Type* укажите в *Currency*;

- в *Field type* вычисляемое поле *Calculated*.

Нажмите *Ок*.

- Закройте редактор полей *ADOT_DisheSale*.

- Выделите компонент *ADOT_DisheSale*. Для него через *Object Inspector* перейдите на закладку *Events* и нажмите два раза мышью на событии *OnCalcFields*, для которого запишите:

```
ADOT_DisheSale->FieldByName("Сумма")-> AsCurrency =  
ADOT_DisheSale->FieldByName("Цена")->AsFloat *  
ADOT_DisheSale->FieldByName("Количество")->AsFloat *  
(1 - ADOT_DisheSale->FieldByName("Скидка")->AsFloat);
```

- Установите у компонента *ADOT_DisheSale* свойство *Active* в *true* для подключения таблицы БД Продано.

- Нажмите F9.

- Посмотрите в *DBGrid*, как вычисляется поле *Сумма* при работе с продано.

Можно предложить другие варианты вычисляемых полей к рассматриваемой БД.

Задание 3. Создание полей выбора данных (lookup-полей)

Кроме обычных полей, связанных с полями таблицы БД и вычисляемых полей, в НД имеется возможность создавать поля выбора данных (lookup поля).

Поля выбора данных одного набора данных содержат значения их другого набора данных, связанного по ключу с НД, к которому принадлежит поле выбора данных. Поле выбора данных всегда доступно только для чтения и не может быть одновременно полем выбора данных и вычисляемым полем. Реляционное отношение НД, служащего источником значений для поля выбора данных и НД, к которому оно принадлежит, есть "один-ко-многим" и реже "один-к-одному". Это означает, что на один вариант значения в наборе данных - источнике должно приходиться одно или несколько связанных значений в НД, к которому принадлежит поле выбора.

Для определения поля набора данных необходимо создать новое поле в редакторе полей, сразу же установив радио-группу Field Type в значение Lookup.

Затем устанавливаются значения свойств:

- DataSet - property LookupDataSet - имя НД - источника значений для поля выбора данных;

- Key Fields - property KeyFields - индексные поля набора данных - владельца поля выбора данных. По этим полям НД - владелец соединяется с НД - источником значений поля выбора данных. Если в индексе имеется несколько полей, они перечисляются через точку с запятой;

- Lookup Keys - property LookupKeyFields - индексные поля НД - источника значений для поля выбора. По значениям этих индексных полей устанавливается связь набора - источника со значениями индексных полей НД - владельца поля выбора. Если в индексе имеется несколько полей, они перечисляются через точку с запятой;

Result Field - property LookupResultField - поле набора данных - источника, возвращаемое в качестве результата. Необходимо следить, чтобы тип вновь создаваемого поля и поля результата совпадали.

Заметим, что свойство поля property Lookup должно быть установлено в True.

Таким образом, поля выбора данных можно использовать для автоматического занесения информации в одну таблицу БД из другой таблицы БД.

- *Создайте в наборе данных ADOT_Sales поле выбора данных из набора данных ADOT_Sotrudniki, так как при работе с продажи не удобно указывать коды сотрудников. Для этого выделите на модуле данных DMMain компонент ADOT_Sales.*

- *Установите у компонента ADOT_Sales свойство Active в false, так как сейчас будем менять набор данных.*

- *Два раза щелкните мышью по набору данных ADOT_Sales (запуская тем самым редактор полей).*

- В редакторе полей *ADOT_Sales* нажмите правую кнопку мыши и выберите *New Field*. Затем в окне диалога укажите для нового поля:

- в *Field Properties*:
 - *Name* в *Сотрудник*;
 - *Type* укажите в *String*;
 - *Size* укажите в *50*;
- в *Field type* поле выбора *Lookup*;
- в *Lookup Definition* укажите:
 - *DataSet* в *ADOT_Sotrudniki*;
 - *Key Fields* в *Код_сотрудника*;
 - *Lookup Keys* в *Код_сотрудника*;
 - *Result Field* в *ФИО*;
- Нажмите *Ок*.

- После создания поля выбора *Сотрудник* можно скрыть поле *Код_сотрудника* в *ADOT_Sales*. Для этого выделите поле *Код_сотрудника* в редакторе полей. Укажите для него в *Object Inspector* свойство *Visible* (видимо) в *false*.

- Закройте редактор полей *ADOT_Sales*.
- Установите у компонента *ADOT_Sales* свойство *Active* в *true* для подключения таблицы БД Продажи.
- Нажмите *F9*.
- Посмотрите в *DBGrid*, как отрабатывает поле выбора данных *Сотрудник* при работе с продажами.

- Создайте в наборе данных *ADOT_DisheSale* поле выбора данных из набора данных *ADOT_Dishes*, так как при работе с продано не удобно указывать коды блюд. Для этого выделите на модуле данных *DMMain* компонент *ADOT_DisheSale*.

- Установите у компонента *ADOT_DisheSale* свойство *Active* в *false*, так как сейчас будем менять набор данных.

- Два раза щелкните мышью по набору данных *ADOT_DisheSale* (запуская тем самым редактор полей).

- В редакторе полей *ADOT_DisheSale* нажмите правую кнопку мыши и выберите *New Field*. Затем в окне диалога укажите для нового поля:

- в *Field Properties*:
 - *Name* в *Блюдо*;
 - *Type* укажите в *String*;
 - *Size* укажите в *40*;
- в *Field type* поле выбора *Lookup*;
- в *Lookup Definition* укажите:
 - *DataSet* в *ADOT_Dishes*;
 - *Key Fields* в *Код_блюда*;
 - *Lookup Keys* в *Код_блюда*;
 - *Result Field* в *Название*;
- Нажмите *Ок*.

- *Закройте редактор полей ADOT_DisheSale.*
- *Установите у компонента ADOT_DisheSale свойство Active в true для подключения таблицы БД Продано.*
- *Нажмите F9.*
- *Посмотрите в DBGrid, как отрабатывает поле выбора данных Продано при работе с продано.*

Задание 4. Установление связей между наборами данных в приложении

Внимание:

Предварительно таблицы БД необходимо связать в СУБД с обеспечением ссылочной целостности.

Связь между наборами данных в приложении можно устанавливать различными способами. Наиболее распространены:

- использование запросов;
- установка связи Master-Detail (родитель-потомок);
- создание подстановочных полей (Lookup).

Подстановочные поля были нами созданы ранее (см. задание 3).
Запросы будут рассмотрены позднее.

Покажем связь между всеми таблицами БД для кафе: Сотрудники, Продажи, Продано, Блюда. Для БД имеем связи:

- *Сотрудники 1:М Продажи;*
- *Продажи 1:М Продано;*
- *Блюда 1:М Продано.*

Покажем связь между всеми таблицами БД для кафе: Сотрудники, Продажи, Продано, Блюда в отдельной форме:

- *связь между Блюда и Продано через подстановочное поле (задание 3);*
- *связь между Сотрудники и Продажи через подстановочное поле (задание 3);*
- *связь между Продажи и Продано через Master-Detail.*

- *Добавьте в проект новую форму File/New/Form.*
- *Свойство Name у новой форму установите в fmSales.*
- *Сохраните форму в проект. Для этого нажмите File/Save all и в диалоговом окне сохраните новую форму в той же папке, что и другие модули проекта. При сохранении укажите новому модулю имя Sales.cpp.*
- *Перейдите на форму fmSales.*

- Перейдите на вкладку *DataAccess*. Найдите на данной вкладке компонент *DataSource* и разместите его объект (*DataSource1*) на форму *fmSales*.

- Для *DataSource1* установите свойство *Name* в *dsSales* – так мы обозначим его привязку к набору данных *DMMain->ADOT_Sales* (для Продаж).

- Наши наборы данных находятся на модуле данных в модуле *DM*. Данный файл *DM.h* необходимо подключить в форме *fmSales*.

Для этого, находясь на форме *fmSotrudniki*, выберите в меню *File/Include Unit Hdr...* (или нажмите *Alt+F11*). В диалоговом окне *Use Unit* выберите модуль *DM*.

После этого в модуле *Sales.cpp* появится запись *#include "DM.h"*

- Установите у *dsSales* свойство *DataSet* в набор данных *DMMain->ADOT_Sales*.

- Перейдите на вкладку *DataAccess*. Найдите на данной вкладке компонент *DataSource* и разместите его объект (*DataSource1*) на форму *fmSales*.

- Для *DataSource1* установите свойство *Name* в *dsDisheSale* – так мы обозначим его привязку к набору данных для Продано.

- Установите у *dsDisheSale* свойство *DataSet* в набор данных *DMMain->ADOT_DisheSale*.

- С закладки *Win32* разместите на форму *fmSales* компонент *ToolBar1*.

- С закладки *Data Controls* разместите на компонент *ToolBar1* компонент *DBNavigator1*. Для него установите:

- свойство *Name* в *DBNavigatorSales*;

- свойство *DataSource* в *dsSales*;

- С закладки *Data Controls* разместите на форму *fmSales* компонент *DBGrid1*. Для него установите свойства:

- свойство *Name* в *DBGridSales*;

- свойство *DataSource* в *dsSales*;

- свойство *Align* в *alTop*.

- С закладки *Standart* разместите компонент *GroupBox1*. Установите для него:

- свойство *Name* в *GroupBoxDiheSale*;

- свойство *Caption* в Блюда в продаже;

- свойство *Align* в *alClient*.

- С закладки *Data Controls* на компонент *GroupBoxDiheSale* разместите компонент *DBNavigator1*. Для него установите:

- свойство *Name* в *DBNavigatorDisheSale*;

- свойство *DataSource* в *dsDisheSale*;

- свойство *Align* установите для него в *alTop*.

- С закладки *Data Controls* на компонент *GroupBoxDisheSale* разместите компонент *DBGrid1*. Для него установите свойства:

- свойство *Name* в *DBGridDisheSale*;

- свойство *DataSource* в *dsDisheSale*;

- свойство *Align* в *alClient*.

Далее включим связь *Master-Detail* между *Продажи* и *Продано*. Для этого перейдите на форму *DMMain*.

- Подключите на форме *DMMain* форму *fmSales*.

Для этого, находясь на форме *DMMain*, выберите в меню *File/Include Unit Hdr...* (или нажмите *Alt+F11*). В диалоговом окне *Use Unit* выберите модуль *Sales*.

После этого в модуле *DM.cpp* появится запись *#include "Sales.h"*.

- На форме *DMMain* выделите компонент *ADOT_DisheSale*. Для него установите:

- свойство *MasterSource* (родительский источник данных) в *fmSales->dsSales*;

- нажмите на свойстве *MasterFields* (установка поля связи с родительским источником данных). Откроется диалог для связывания родительского (*Master*) и дочернего (*Detail*) НД. В нашем случае:

Продажи

Номер продажи	Дата	Код сотрудника (FK)
1П	12.02.2008	1
2П	12.02.2008	1
3П	12.02.2008	2
4П	20.02.2008	3
5П	21.02.2008	1

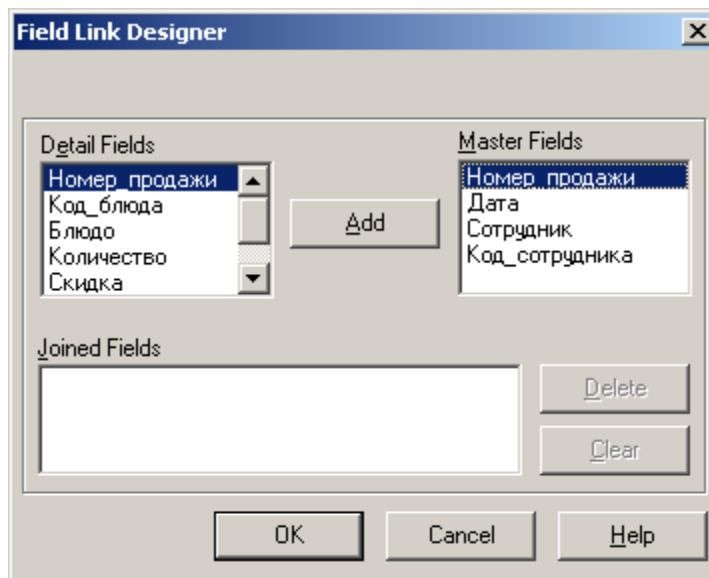
Блюда

Код блюда	Название	Тип	Цена меню	Выход
1Б	Оливье	Салаты	100 руб.	250 гр.
2Б	Борщ	Супы	90 руб.	300 гр.
3Б	Цезарь	Салаты	120 руб.	270 гр.

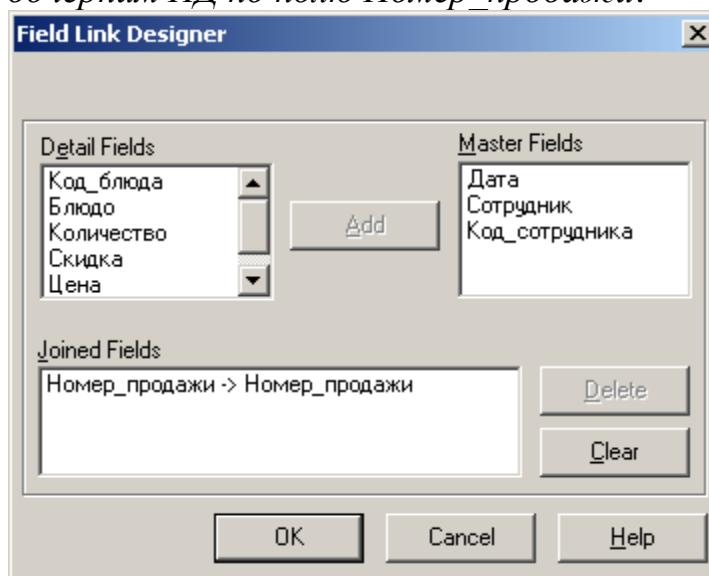
Продано

Номер продажи (FK)	Код блюда (FK)	Количество	Цена	Скидка
1П	1Б	2	100 руб.	0%
1П	3Б	1	120 руб.	0%
2П	2Б	10	90 руб.	10%
3П	1Б	3	100 руб.	0%
3П	2Б	1	90 руб.	0%
3П	3Б	1	120 руб.	5%

В диалоге при этом:



После нажатия кнопки Add произойдет установка связи между родительским и дочерним НД по полю Номер_продажи.



- Свойство MasterFields при этом установится в Номер_продажи.

Внимание:

После установки связи Master-Detail в работающем приложении для выбранной пользователем строки родителя (в нашем случае это Продажи) будут отражаться только связанные с выбранным родителем потомки в дочерней наборе данных (в нашем случае Продано).

- Перейдите на форму *fmMain*.
- Для акции *actSales* в *ActionList1* пропишите в событии *OnExecute*:

```
fmSales->Caption="Продажи блюд";  
fmSales->ShowModal();
```

- Наша акция *actSales* уже выбрана для пунктов меню и панели инструментов.
- Нажмите F9.
- В приложении посмотрите отработку установленных связей.

Задание 5. Оформление отчета к лабораторной работе

Оформить отчёт со следующим содержанием:

1. Титульный лист.
2. Цель работы.
3. Постановка задачи.
4. Краткая теория и ход выполнения заданий.
5. Описание результатов.

Опишите работу с полями, вычисляемые поля, поля выбора данных, наборы данных с установленной связью. Приведите скриншоты полученной программы и листинг программы.

6. Заключение (выводы).