

Работа с полями

Объект поля.

Редактор полей.

Операции с полями.

Объект поля.

Объект поля Field имеет тип TField. Тип TField – абстрактный класс, который может использовать след. классы:

- ◆ TMemoField - поле типа Memo;
- ◆ TGraphicField - графическое поле;
- ◆ TStringField - поле строкового значения;
- ◆ TDateTimeField - поле даты и времени;
- ◆ TFloatField - поле вещественного числа;
- ◆ TCurrencyField - поле денежной суммы;
- ◆ TIntegerField - поле целого числа;
- ◆ TAutoIncField - поле автоинкрементного значения;
- ◆ TBlobField - поле BLOB-значения;
- ◆ TBlobField - поле BLOB-объекта;

Способы задания состава полей набора данных

- ◆ по умолчанию (динамические поля);
Динамические поля создаются автоматически при разработке приложения
- ◆ с помощью редактора полей (статические поля).

Свойство **DefaultFields** (Boolean) определяет вид поля НД.

Редактор полей

Статические поля создаются с помощью *Редактора полей*, который позволяет:

- ◆ определять вычисляемые поля, значения которых рассчитываются с помощью выражений, в которых можно использовать значения других полей;
- ◆ ограничить состав полей НДС;
- ◆ изменить порядок полей НДС;
- ◆ скрыть или показать отдельные поля при выполнении приложения;
- ◆ задать формат отображения или редактирования данных поля на этапе разработки приложения.

Создание нового статического поля

New Field [X]

Field properties

Name: Component:

Type: Size:

Field type

Data Calculated Lookup

Lookup definition

Key Fields: Dataset:

Lookup Keys: Result Field:

OK Cancel Help

Группа управляющих элементов **Field properties** (Свойства поля):

- ◆ В поле **Name** (Имя) задается значение свойства **FieldName**, то есть имя поля
- ◆ в поле **Component** – название компонента (объекта типа TField) поля. Значение поля **Component** формируется Delphi автоматически
- ◆ В полях **Type** и **Size** указываются тип и размер поля. Тип должен быть задан в обязательном порядке для любого поля.

Группа переключателей Field type (Тип поля)

- ◆ **Data** – поле данных, заменяющее соответствующее физическое поле таблицы;
 - ◆ **Calculated** – вычисляемое поле, значение которого рассчитывается в обработчике события OnCalcFields во время выполнения приложения;
 - ◆ **LookUp** – поле выбора, значение которого можно выбирать из списка, формируемого на основе заданных критериев и правил.
- 

Создание вычисляемого поля

- ◆ 1 шаг. Создать статическое вычисляемое поле.

New Field

Field properties

Name: Component:

Type: Size:

Field type

Data Calculated Lookup

Lookup definition

Key Fields: Dataset:

Lookup Keys: Result Field:

OK Cancel Help

2 шаг. Написать обработчик события **OnCalcFields** компонента **Table**.

```
procedure  
  TForm1.Table2CalcFields(DataSet:  
    TDataSet);  
begin  
  Table2Summa.Value :=  
  Table2Kolvo.Value*Table2Zena.Value;  
end;
```

Создание полей выбора данных

Поле выбора данных одного НД содержит значения из другого набора данных, связанных по ключу с НД, к которому принадлежит это поле.

Первый НД - родительский, а второй - дочерний.

Создадим новое поле в НД **Table2** с именем **NSklad** типа **Integer**. Установим переключатель в положение **Lookup**.

После выбора переключателя становятся доступными элементы группы **Lookup definition**, с помощью которых устанавливаются параметры связи наборов данных

New Field

Field properties

Name: NSklad Component: Table2NSklad

Type: Integer Size: 0

Field type

Data Calculated Lookup

Lookup definition

Key Fields: Dataset:

Lookup Keys: Result Field:

OK Cancel Help

Key Fields - Определяет список ключевых полей родительского НД. По этим полям построен индекс для связи родительского НД с дочерним. Если в индексе несколько полей, они разделяются точками с запятыми

DataSet - Определяет имя родительского НД

LookupKeys - Определяет список ключевых полей дочернего НД. По этим полям построен индекс для связи дочернего НД с родительским.

Result Fields - Поле родительского НД, возвращаемое в качестве результата. Необходимо следить, чтобы тип вновь создаваемого поля совпадал с типом указанного здесь поля

Операции с полями.

Свойства и методы для доступа к полям:

- ◆ Свойство **FieldCount**: integer указывает количество полей НД. Количество полей НД может отличаться от физического числа полей таблицы БД, поскольку в НД не обязательно включаются все поля таблицы.
- ◆ Свойство **FieldNo** : integer – **порядковый номер поля в НД**

Операции с полями

- ◆ Свойство **Fields[Index:Integer]: TField** – возвращает содержимое указанного в параметре Index поля НД.

К отдельному полю можно обратиться, указав его номер **Index** в массиве Fields. Номера полей изменяются от 0 до FieldCount-1.

Примеры.

Edit1.Text:=Table1.Fields[5].AsString; – в компонент Edit1 появится та информация, которая находится в шестом поле НД.

Table1.Fields[0].Value:='Монитор';

Операции с полями.

- ◆ Свойства объекта Field – **AsString**, **AsInteger**, **AsFloat**, **AsBoolean** позволяют обращаться к значению поля текущей записи как к строковому, целочисленному, вещественному или логическому значению.

```
procedure
  TForm1.Table2CalcFields(DataSet:
    TDataSet);
begin
  Table2Summa.asFloat:=
  Table2Kolvo.asInteger*Table2Zena.asFloat;
end;
```

Операции с полями.

- ◆ Чтобы обращаться к какому-либо полю HD не по его номеру в массиве полей, а по его имени существуют методы **FindField** и **FieldByName**.

```
Table1.FieldByName('Naim').Value:='Монитор';
```

Операции с полями

```
procedure
  TForm1.Table2CalcFields(DataSet:
    TDataSet);
begin
  Table2.FieldByName('Summa').asFloat:=
  Table2. FieldByName('Kolvo').asInteger*
  Table2. FieldByName('Zena').asFloat;
end;
```

Операции с полями.

Доступ к полям

Товар	Kod	DataVipusk	SrokGodnos	Kolvo	Cena
Хлеб	123	02.06.03	05.06.03	200	7,50р.
Сметана	109	03.06.03	07.06.03	38	14,20р.
Масло сливочное	431	12.05.03	12.06.03	324	56,80р.
Сахар	543	15.04.03	15.04.06	753	17,00р.
Колбаса "Докторская"	834	14.06.03	20.06.03	128	97,00р.

Σ Общее количество товара Label1

Σ Общая стоимость единиц товара Label2

Общая стоимость товара для текущей записи Label3

Для кнопок необходимо написать соответствующие процедуры:

```
procedure TForm1.BitBtn1Click(Sender: TObject);  
var s:integer;  
      i:longint;  
begin  
s:=0;  
table1.First;  
for i:=1 to table1.RecordCount do  
begin  
s:=s+table1.fieldbyname('kolvo').asinteger;  
table1.Next;  
end;  
label1.Caption:=inttostr(s);  
end;
```

```
procedure TForm1.BitBtn2Click(Sender:  
    TObject);  
    var s:real;  
        i:longint;  
begin  
    s:=0;  
    table1.First;  
    While not table1.Eof do  
    begin  
    s:=s+table1.fieldbyname('cena').asfloat;  
    table1.Next;  
    end;  
    label2.Caption:=floattostr(s);  
end;
```

procedure

```
TForm1.BitBtn3Click(Sender:  
TObject);
```

```
var s:real;
```

```
begin
```

```
s:=table1.fieldbyname('kolvo').asfloat  
*table1.fieldbyname('cena').asfloat;
```

```
label3.Caption:=floattostr(s);
```

```
end;
```

