

## Контрольная работа № 2

### Создание клиентского приложения к СУБД Oracle 10g

Цель работы: создать клиентское приложение для работы оператора ЭВМ с базой Oracle при помощи языка программирования Delphi.

Задание. На основе созданной базы данных по варианту работы №1 разработать работающее клиентское приложение. Приложение должно включать удобный интерфейс пользователя для работы с БД, а также ряд выгружаемых отчетов, выгружаемых в MS Excel и сгенерированных при помощи средства FastReport 4.7.

### Пример выполнения работы

Для работы с СУБД Oracle в связке со средой программирования Delphi наиболее удобно использовать не стандартные компоненты, включаемые в дистрибутив Delphi, а сторонние, например, библиотеку компонентов Direct Oracle Access, которые существуют для всех версий среды Delphi. Эти компоненты обеспечивают большее удобство разработки и большую скорость работы интерфейса, чем стандартные работающие через odbc драйвер доступа к СУБД. Библиотека Direct Oracle Access устанавливается при закрытой среде Delphi простым запуском установочного exe файла. После установки в палитре компонентов Delphi появляется вкладка Oracle, рис. 1.

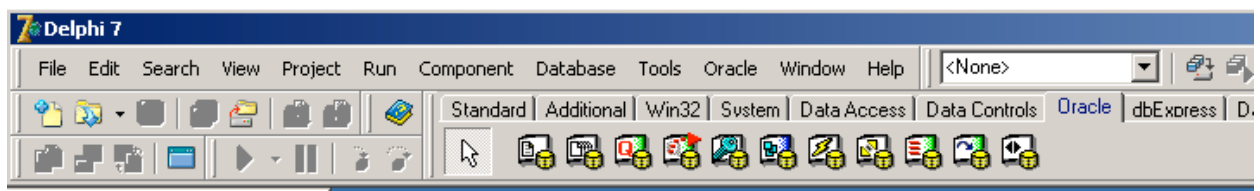


Рис. 1

В ходе выполнения работы будем использовать 3 основных компонента из этой вкладки: OracleSession – компонент для коннекта к БД, OracleDataSet –компонент по своему функционалу напоминающий стандартный компонент Table контейнер для отображения запросов из БД, OracleQuery -компонент для создания транзакций к СУБД, т.е. удаление, редактирование и вставку записей.

Разберем ряд примеров для работы с СУБД (файлы примеров находятся по пути “С:\Примеры” виртуальной машины).

Пример 1. Работа с компонентам OracleDataSet (файлы примера в папке С:\Примеры\test 1)

Создадим новую форму в среде Delphi 7. На вновь созданную форму поместим компоненты OracleSession, OracleDataSet, DataSource, DBNavigator, Button и DBEdit в количестве как на рис.2.

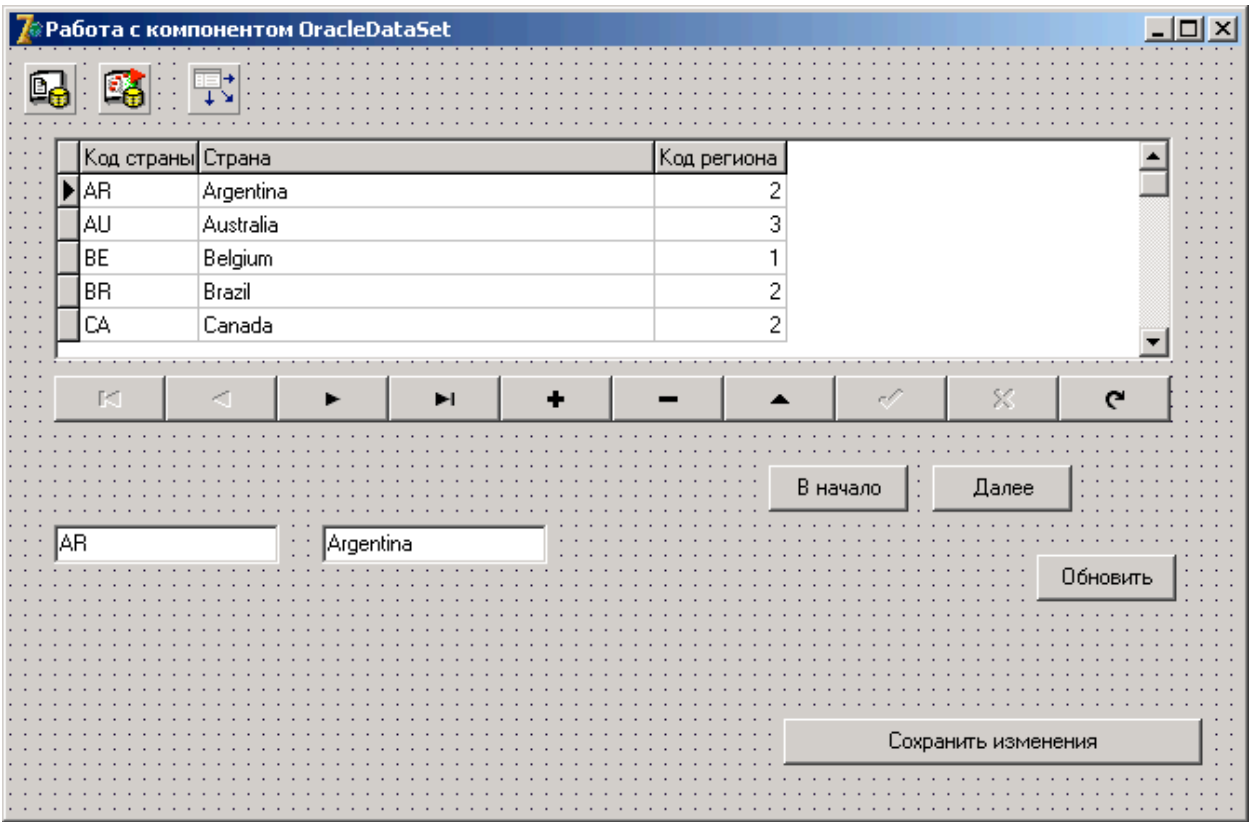


Рис. 2

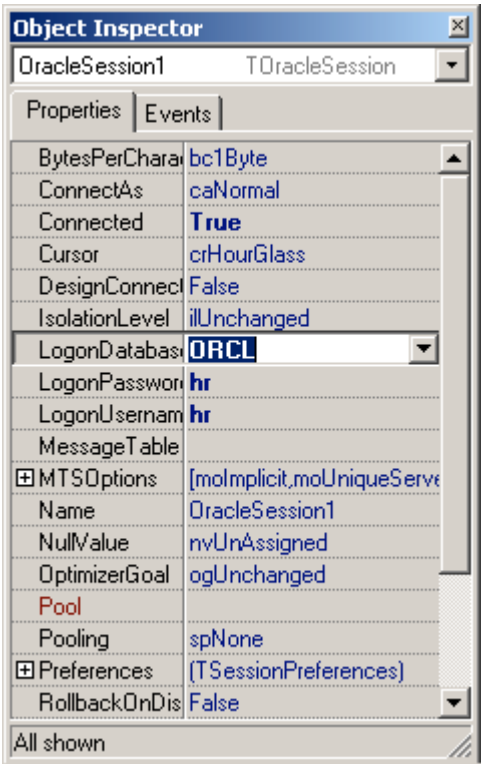


Рис. 3

В компоненте OracleSession пропишем название экземпляра базы Oracle, название схемы и пароль, как на рис.3.

В компоненте DataSource1 записываем OracleDataSet1 рис.4

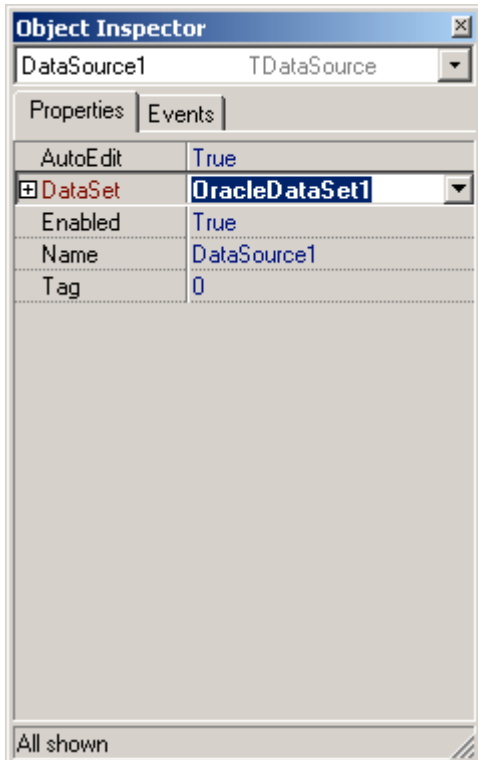


Рис.4

Аналогично записываем ссылку на OracleDataSet во всех DB компонентах.

В компоненте DBGrid1 в свойстве datasource указываем DataSource1.

Для передвижения по записям используем метод OracleDataSet1.Next т.к. dbGrid выполняет лишь функцию визуализации OracleDataSet.

Чтобы ввести необходимые названия полей, необходимо 2 раза кликнуть мышью по компоненту dbgrid рис.5

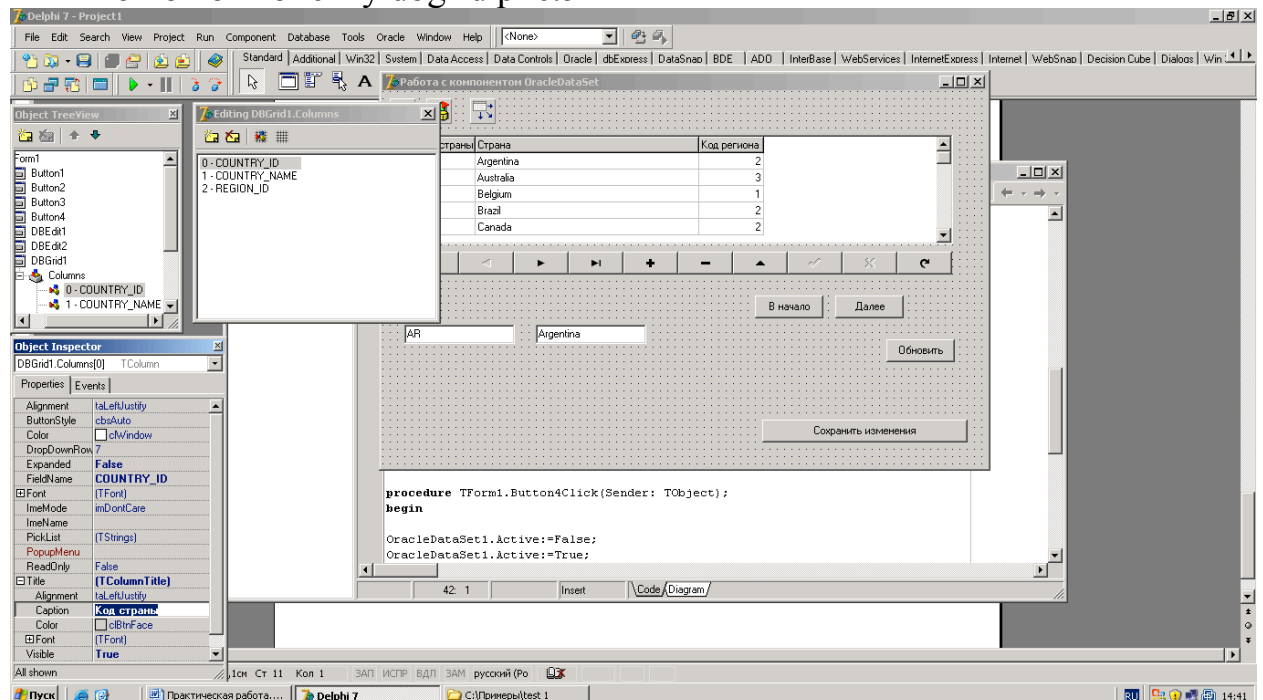


Рис. 5

Затем кликнуть правой клавишей мыши по окну Editing Dbgrids.Columns и добавить все записи рис.6.

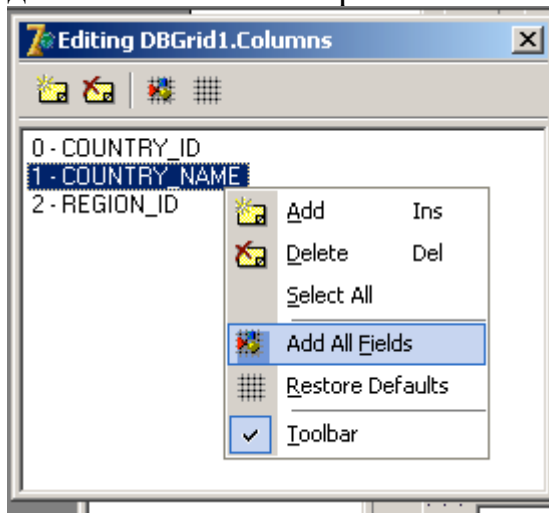


Рис.6

Затем в свойствах каждого столбца можно ввести его новое название.

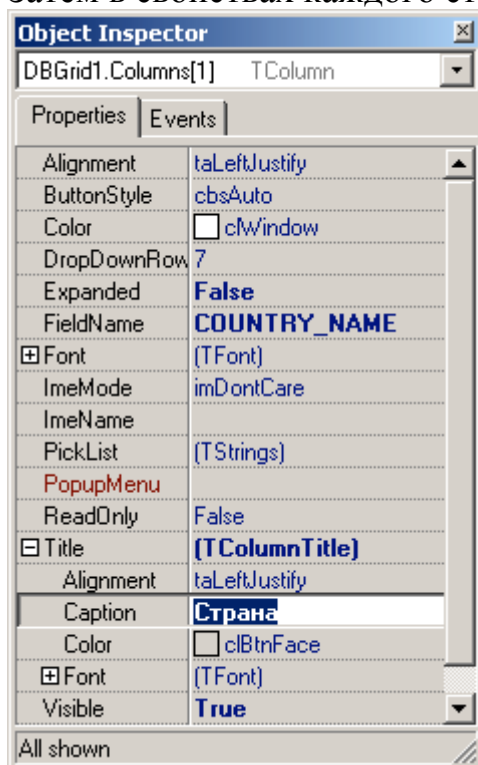


Рис.7

Для обновления данных при удалении или корректировании используем операторы:

```
OracleDataSet1.Active:=False;
```

```
OracleDataSet1.Active:=True;
```

Для перехода на первую запись используем метод OracleDataSet1.First.

Пример 2. Работа с полями OracleDataSet в коде (файлы примера в папке C:\Примеры\test 2)

Размещаем компоненты, как на рисунке 8

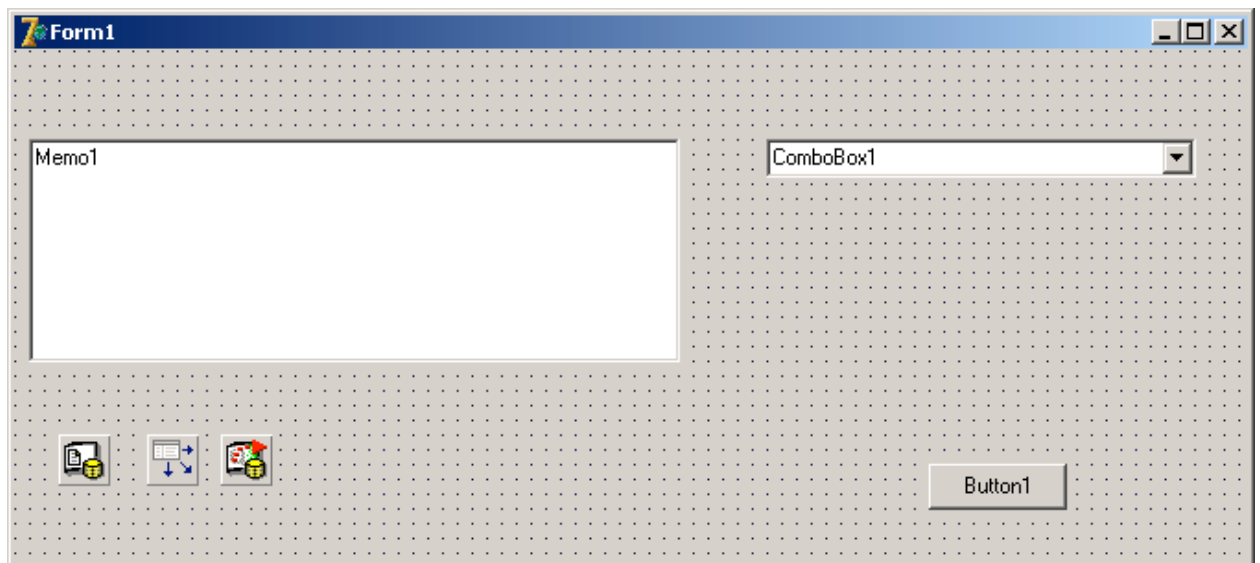


Рис.8

Связь с базой настраивается аналогично примеру 1. В данном примере для удобства записываем SQL текст запроса прямо внутри кода Delphi, объявляя его строковой константой.

```
const SQL_Text=
'select s.job_title,' + #13#10 +
'    e.first_name,' + #13#10 +
'    e.last_name,' + #13#10 +
'    lower(e.email)||"@mail.ru" as email,' + #13#10 +
'    e.salary' + #13#10 +
'    from JOB_HISTORY j, employees e, departments d, jobs s' + #13#10 +
'where j.employee_id=e.employee_id' + #13#10 +
'    and j.department_id=d.department_id' + #13#10 +
'    and j.job_id=s.job_id';
```

Такой код получен копированием запроса из средства PL/SQL Developer см. рис. 9. Такой подход очень удобен т.к. автоматически генерируются необходимые специальные символы оформления SQL запроса.

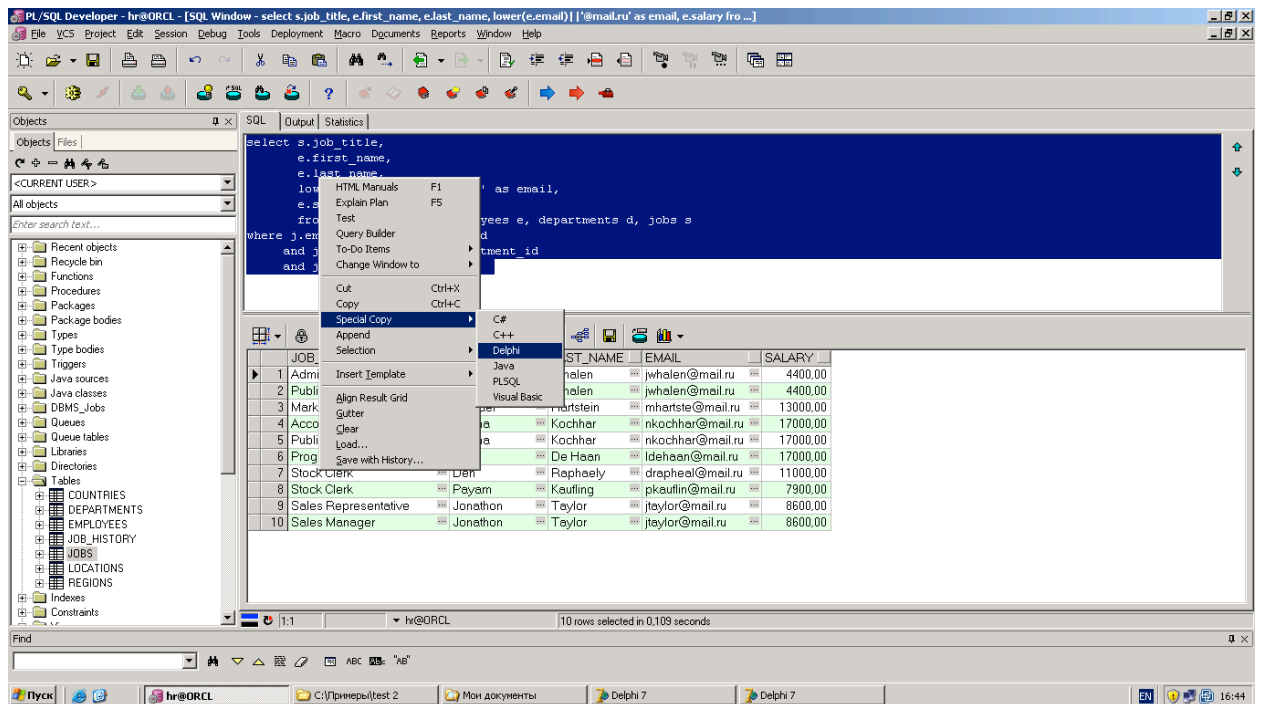


Рис.9

По событию onClick кнопки записываем код

```
OracleDataSet1.Active:=false;
OracleDataSet1.SQL.Add(SQL_Text);
OracleDataSet1.Active:=true;
```

```
OracleDataSet1.First;
```

```
While not OracleDataSet1.Eof do
```

```
begin
```

```
s:=OracleDataSet1.Fields[1].AsString; //Получить данные можно
```

```
по индексу ...
```

```
// А можно по имени поля...
```

```
s1:= OracleDataSet1.FieldValues['last_name']; //А можно по
```

```
имени поля
```

```
Memo1.Lines.Add(s+' '+s1);
```

```
ComboBox1.Items.Add(s);
```

```
OracleDataSet1.Next; //Не забываем про "Next" иначе беск. цикл
```

```
end;
```

В коде показан доступ к значению полей, как по имени так и по индексу поля.

Пример 3. Интерфейс оператора для ввода транзакций (файлы примера в папке C:\Примеры\test 3)

The screenshot shows a Windows-style application window titled "Обновление, вставка, удаление". Inside, there is a table with the following data:

JOB_TITLE	FIRST_NAME	LAST_NAME	EMAIL	SALARY	ID
Administration Assistant	Jennifer	Whalen	jwhalen@mail.ru	4405	1
Public Accountant	Jennifer	Whalen	jwhalen@mail.ru	4400	2
Marketing Representative	Michael	Hartstein	mhartste@mail.ru	13000	3
Accounting Manager	Neena	Kochhar	nkochhar@mail.ru	17000	4
Public Accountant	Neena	Kochhar	nkochhar@mail.ru	17000	5
Programmer	Lex	De Haan	ldehaan@mail.ru	17000	6

Below the table, there is a form with the following fields and buttons:

- Buttons: "<--", "-->" (for navigation), "Обновить" (Update), "Вставить" (Insert), "Удалить" (Delete).
- Form fields:
  - № п/п: Edit6
  - Профессия: Edit1
  - Имя: Edit2
  - Фамилия: Edit3
  - E-mail: Edit4
  - Зарплата: Edit5

Рис. 10

Связь с базой настраивается аналогично примеру 1. При нажатии кнопок со стрелками происходит перемещение по строкам dataset, одновременно компоненты Edit заполняются значениями столбцов из базы при помощи процедуры TForm1.UpdateEdit();

Для генерирования необходимого SQL запроса "склеиваем" его внутри кода: SQL\_up:=

```
'update OFFICE_EMPLOYEE t' + #13#10 +
'SET t.job_title='+#39+Edit1.Text+#39+', ' + #13#10 +
' t.first_name='+#39+Edit2.Text+#39+', ' + #13#10 +
' t.last_name='+#39+Edit3.Text+#39+', ' + #13#10 +
' t.email='+#39+Edit4.Text+#39+', ' + #13#10 +
' t.salary='+#39+Edit5.Text+#39+', ' + #13#10 +
'where t.id='+Edit6.Text;
```

Для внесения изменений в базу сначала записываем запрос в компонент, при помощи метода Execute выполняем его и фиксируем изменения оператором Commit.

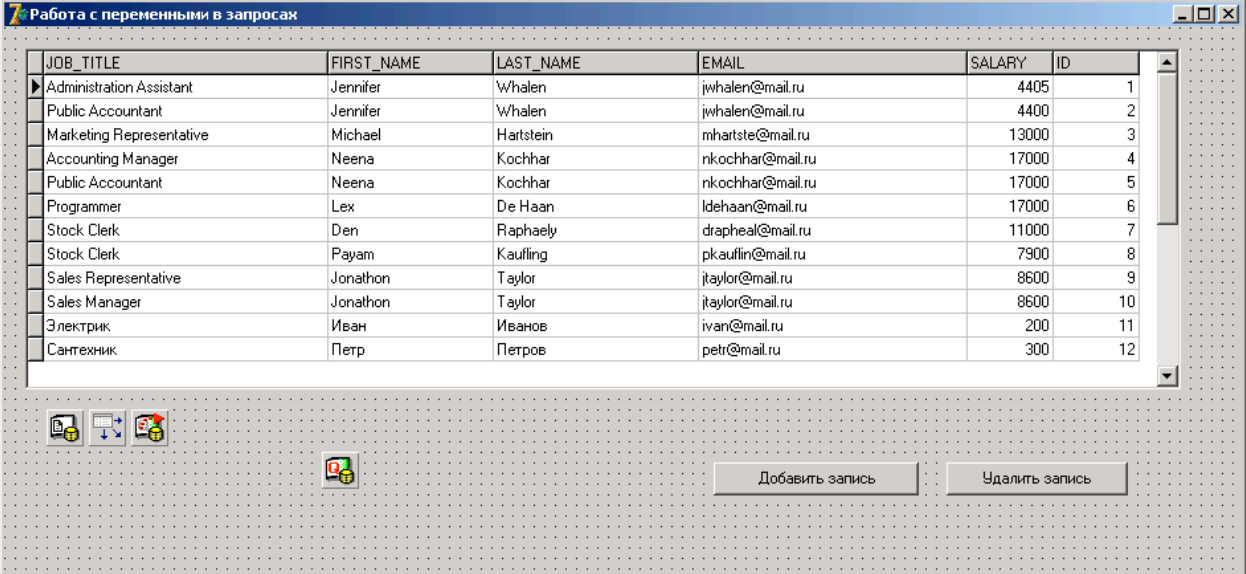
```
OracleQuery1.SQL.Text:=SQL_up;
OracleQuery1.Execute;
OracleQuery1.Session.Commit;
```

Чтобы проверить правильно ли работает наш запрос можно скопировать его в буфер обмена:

Clipboard.asText:=SQL\_del.

Пример 4. Работа с переменными в запросах (файлы примера в папке C:\Примеры\test 4)

В примере используются 2 формы и 1 модуль для хранения глобальных переменных.



JOB_TITLE	FIRST_NAME	LAST_NAME	EMAIL	SALARY	ID
Administration Assistant	Jennifer	Whalen	jwhalen@mail.ru	4405	1
Public Accountant	Jennifer	Whalen	jwhalen@mail.ru	4400	2
Marketing Representative	Michael	Hartstein	mhartste@mail.ru	13000	3
Accounting Manager	Neena	Kochhar	nkochhar@mail.ru	17000	4
Public Accountant	Neena	Kochhar	nkochhar@mail.ru	17000	5
Programmer	Lex	De Haan	ldehaan@mail.ru	17000	6
Stock Clerk	Den	Raphaely	drapheal@mail.ru	11000	7
Stock Clerk	Payam	Kauffman	pkauffin@mail.ru	7900	8
Sales Representative	Jonathon	Taylor	jtaylor@mail.ru	8600	9
Sales Manager	Jonathon	Taylor	jtaylor@mail.ru	8600	10
Электрик	Иван	Иванов	ivan@mail.ru	200	11
Сантехник	Петр	Петров	petr@mail.ru	300	12

Рис.11

В этом примере мы храним sql запрос не в тексте программы, а в самом компоненте, при этом нет необходимости обрамлять текст запроса специальными символами, а новый запрос генерируется при помощи переменных рис. 12. Запись значения переменной в компонент осуществляется оператором .SetVariable('V\_JOB\_TITLE',VJOB\_TITLE); 1 параметр имя переменной в компоненте, 2 параметр имя переменной в тексте программы.





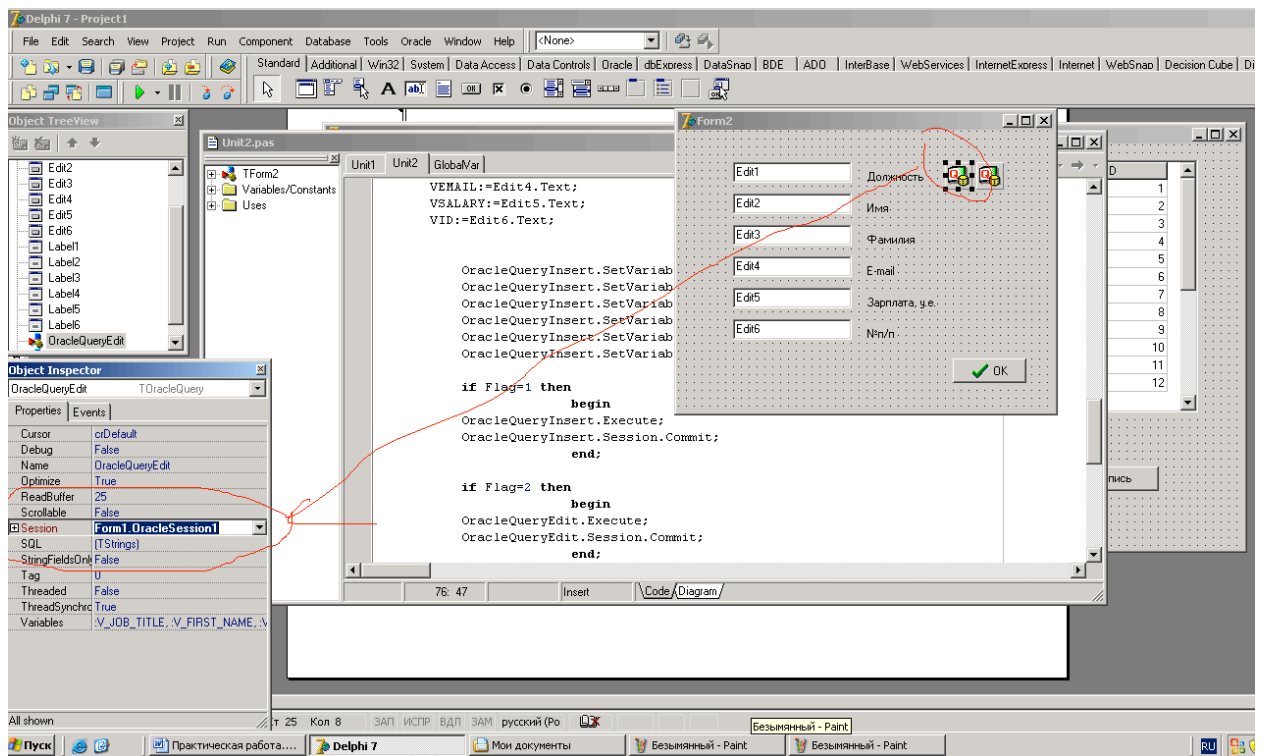


Рис.13

OracleDataSet на 2-й форме примера ссылается на OracleSession 1-й формы, для этого необходимо добавить Unit1 в секцию Uses 2-й формы.

Пример 5. Выгрузка данных в MS Excel (файлы примера в папке C:\Примеры\test 5)

В примере используются возможности стандартной библиотеки ComObj.

```
Var SQL_Excel: AnsiString;
    vPeriod: String;
    XL, Colum: variant;
    SH, Cell : olevariant;
    i,j,k: integer;
```

begin

```
    XL:= createOLEObject('Excel.application'); //Создаем объект типа Excel
    XL.Visible:=true; //Говорим, что он будет виден пользователю
    XL.Workbooks.Add(-4167); //Добавляем новую раб. книгу
    XL.Workbooks[1].WorkSheets[1].Name:='Отчёт по сотрудникам';
//Называем рабочую книгу
    Cell:=XL.Workbooks[1].WorkSheets['Отчёт по сотрудникам']; //Получаем
    привязку к ячейкам
    SH:= XL.WorkSheets['Отчёт по сотрудникам']; //Привязываем к рабочей
    книге
```

```

//Можем работать через диапазоны Range
SH.Range['C1:D1']:='Выгружаем через диапазон... ';

//А можем через ячейки Cells сначала указываем номер строки потом
столбца
Cell.Cells[2,1]:='Выгружаем через ячейку';
Cell.Cells[3,3]:='Дата выгрузки:';
Cell.Cells[3,4]:=DateToStr(date);

//Привязываемся к столбцам
Colum:=XL.Workbooks[1].Worksheets['Отчёт по сотрудникам'].Columns;

//Устанавливаем ширины
Colum.Columns[1].ColumnWidth:=5;
Colum.Columns[2].ColumnWidth:=45;
Colum.Columns[3].ColumnWidth:=25;
Colum.Columns[4].ColumnWidth:=25;
Colum.Columns[5].ColumnWidth:=25;


//Рисуем шапку
//1 столбец
SH.Range['A6'].Borders.LineStyle := 1;
SH.Range['A6'].Borders.Weight:= -4138;
SH.Range['A6'].VerticalAlignment:=-4108;
SH.Range['A6'].HorizontalAlignment:=-4108;
SH.Range['A6']:='№ п.п.';
SH.Range['A6'].Font.Size:=11;
SH.Range['A6'].Font.Color:=clBlue;


//2 столбец
SH.Range['B6'].Borders.LineStyle := 1;
SH.Range['B6'].Borders.Weight:= -4138;
SH.Range['B6'].VerticalAlignment:=-4108;
SH.Range['B6'].HorizontalAlignment:=-4108;
SH.Range['B6']:='Ф.И.';
SH.Range['B6'].Font.Size:=11.5;
SH.Range['B6'].Font.Color:=clRed;


//3 столбец
SH.Range['C6'].Borders.LineStyle := 1;
SH.Range['C6'].Borders.Weight:= -4138;

```

```
SH.Range['C6'].VerticalAlignment:=-4108;  
SH.Range['C6'].HorizontalAlignment:=-4108;  
SH.Range['C6']:='Должность';
```

```
//4 столбец
```

```
SH.Range['D6'].Borders.LineStyle := 1;  
SH.Range['D6'].Borders.Weight:= -4138;  
SH.Range['D6'].VerticalAlignment:=-4108;  
SH.Range['D6'].HorizontalAlignment:=-4108;  
SH.Range['D6']:='E-mail';
```

```
//5 столбец
```

```
SH.Range['E6'].Borders.LineStyle := 1;  
SH.Range['E6'].Borders.Weight:= -4138;  
SH.Range['E6'].VerticalAlignment:=-4108;  
SH.Range['E6'].HorizontalAlignment:=-4108;  
SH.Range['E6']:='Зарплата, в у.е.';
```

```
i:=6; //Грузим из базы с 5 строки
```

```
j:=0; //Просто счетчик
```

```
OracleDataSet1.First;
```

```
While not OracleDataSet1.Eof do
```

```
begin
```

```
inc(i);
```

```
inc(j);
```

```
///Рисуем рамки
```

```
for k:=1 to 5 do
```

```
begin
```

```
Cell.Cells[i,1].Font.Bold:=true; //Жирный цвет
```

```
Cell.Cells[i,k].Borders.LineStyle:= 1; //Рисуем тонкую рамку
```

```
Cell.Cells[i,k].Borders[10].LineStyle:= 1;
```

```
Cell.Cells[i,k].Borders[10].Weight:= -4138;
```

```
end;
```

```
//////////
```

```
Cell.Cells[i,1]:=intToStr(j);
```

```
Cell.Cells[i,2]:=OracleDataSet1.FieldValues['LAST_NAME']+' '+  
OracleDataSet1.FieldValues['FIRST_NAME'];
```

```
Cell.Cells[i,3]:= OracleDataSet1.FieldValues['JOB_TITLE'];
```

```
Cell.Cells[i,4]:= OracleDataSet1.FieldValues['EMAIL'];
```

```
Cell.Cells[i,5]:= OracleDataSet1.FieldValues['SALARY'];
```

```
OracleDataSet1.Next;
end;
```

Используя цикл по строкам OracleDataSet ( While not OracleDataSet1.Eof do) можем выгрузить их в MS Excel.

№ п.п.	Ф.И.	Должность	E-mail	Зарплата, в у.е.
1	Whalen Jennifer	Administration Assistant	jwhalen@mail.ru	4405
2	Whalen Jennifer	Public Accountant	jwhalen@mail.ru	4400
3	Hartstein Michael	Marketing Representative	mhartste@mail.ru	13000
4	Kochhar Neena	Accounting Manager	nkochhar@mail.ru	17000
5	Kochhar Neena	Public Accountant	nkochhar@mail.ru	17000
6	De Haan Lex	Programmer	ldehaan@mail.ru	17000
7	Raphaely Den	Stock Clerk	drapheal@mail.ru	11000
8	Kaufling Payam	Stock Clerk	pkaufli@mail.ru	7900
9	Taylor Jonathon	Sales Representative	jtaylor@mail.ru	8600
10	Taylor Jonathon	Sales Manager	jtaylor@mail.ru	8600
11	Иванов Иван	Электрик	ivan@mail.ru	200
12	Петров Петр	Сантехник	petr@mail.ru	300

Рис. 14

Пример 6. Работа с генератором отчетов FastReport (файлы примера в папке C:\Примеры\test 6)

FastReport является мощным средством для генерирования различных отчетов для разных сред программирования. Позволяет создавать отчеты древовидной структуры, кросс таблицы и т.п. Включая растровые изображения и диаграммы.

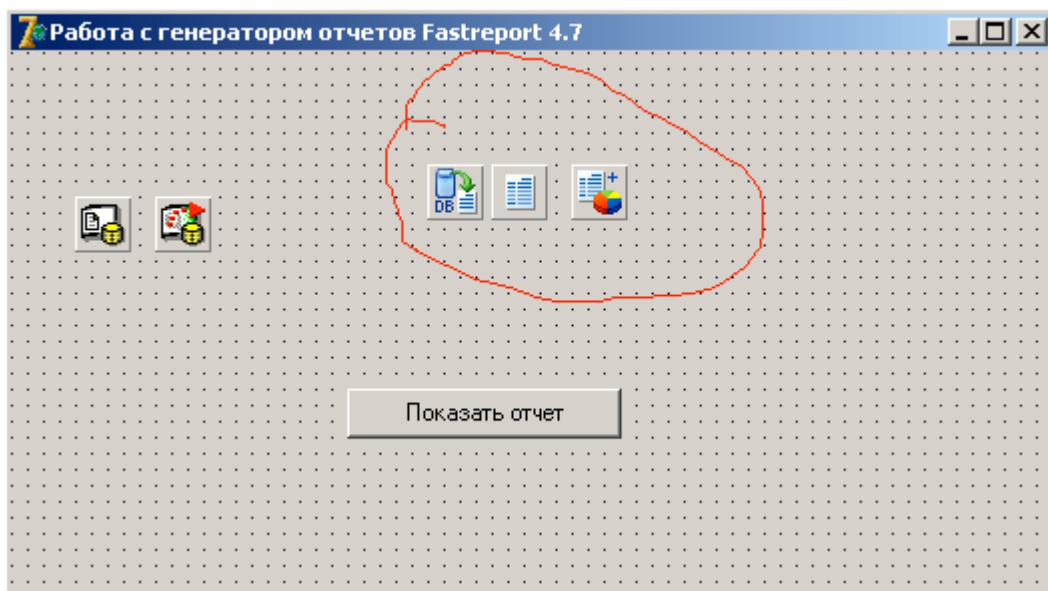


Рис.15

Для работы разместим компоненты `frxReport1`, `frxDBDataset1`, `frxChartObject1`. `frxDBDataset1`-является контейнером данных в свойствах данных которого прописывается `OracleDataSet` компонент. `frxReport1` –это шаблон отчета, `frxDBDataset1` –компонент для построения диаграмм.

Щелкнем 2 раза по компоненту `frxReport1`, и в открывшемся окне выберем `File->New Report`

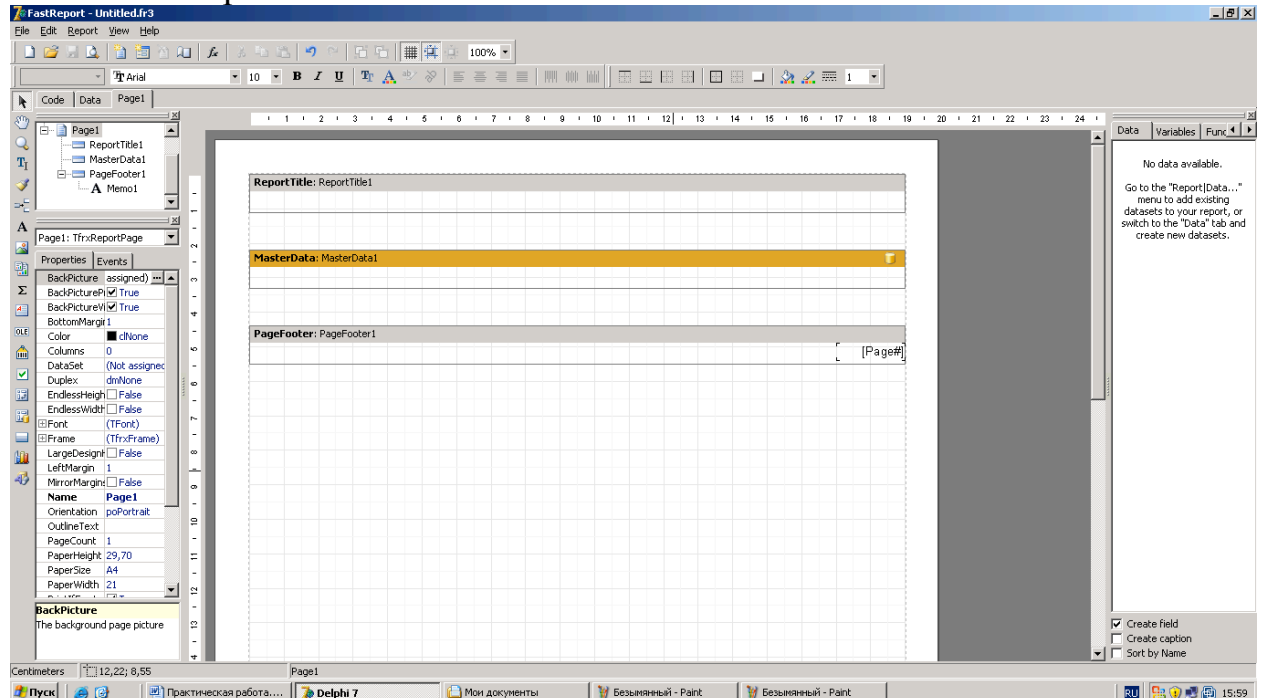


Рис. 16.

Простейший отчет состоит из 3-х бэндов `ReportTitle1`-здесь размещают заголовок отчета, `MasterData1`- это непосредственно данные из базы Oracle и `PageFooter1` –это нижняя часть нашего отчета, где обычно размещают подписи руководителей.

Для подключения данных из OracleDataSet необходимо указать источник данных рис. 17

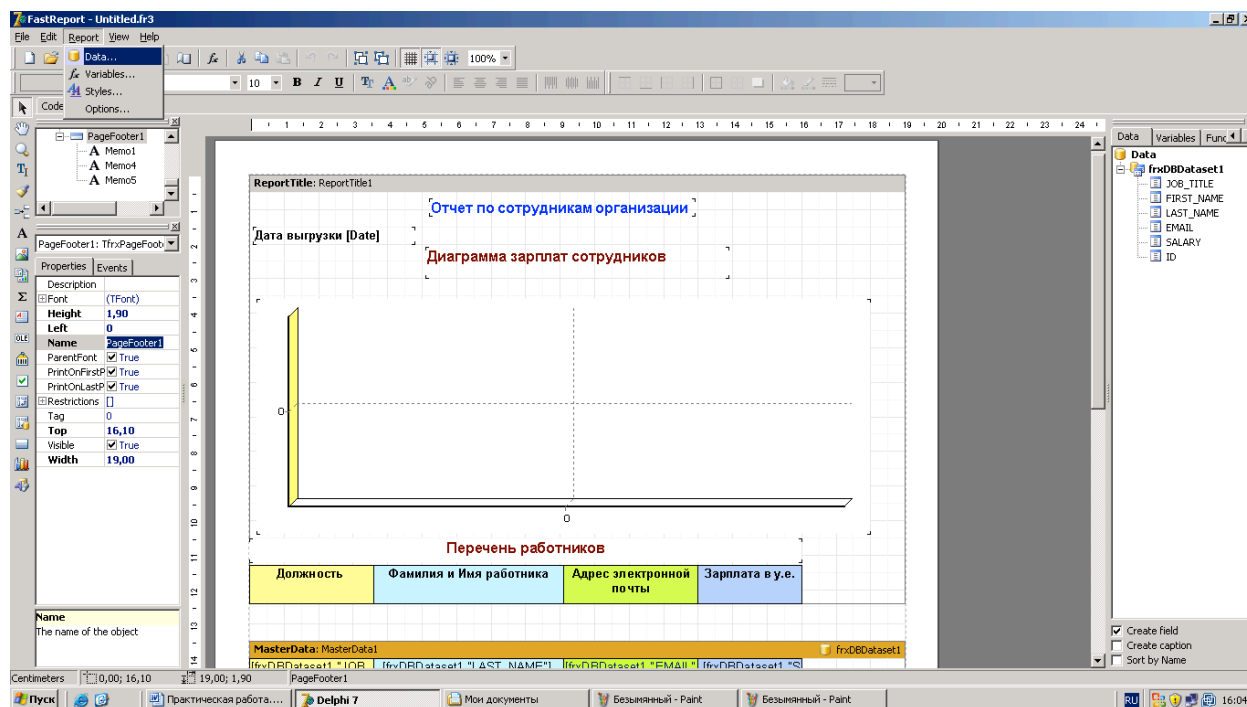


Рис. 17

Для отображения данных в MasterData так же необходимо указать источник данных рис. 18 щелкнув 2 раза по этому бэнду.

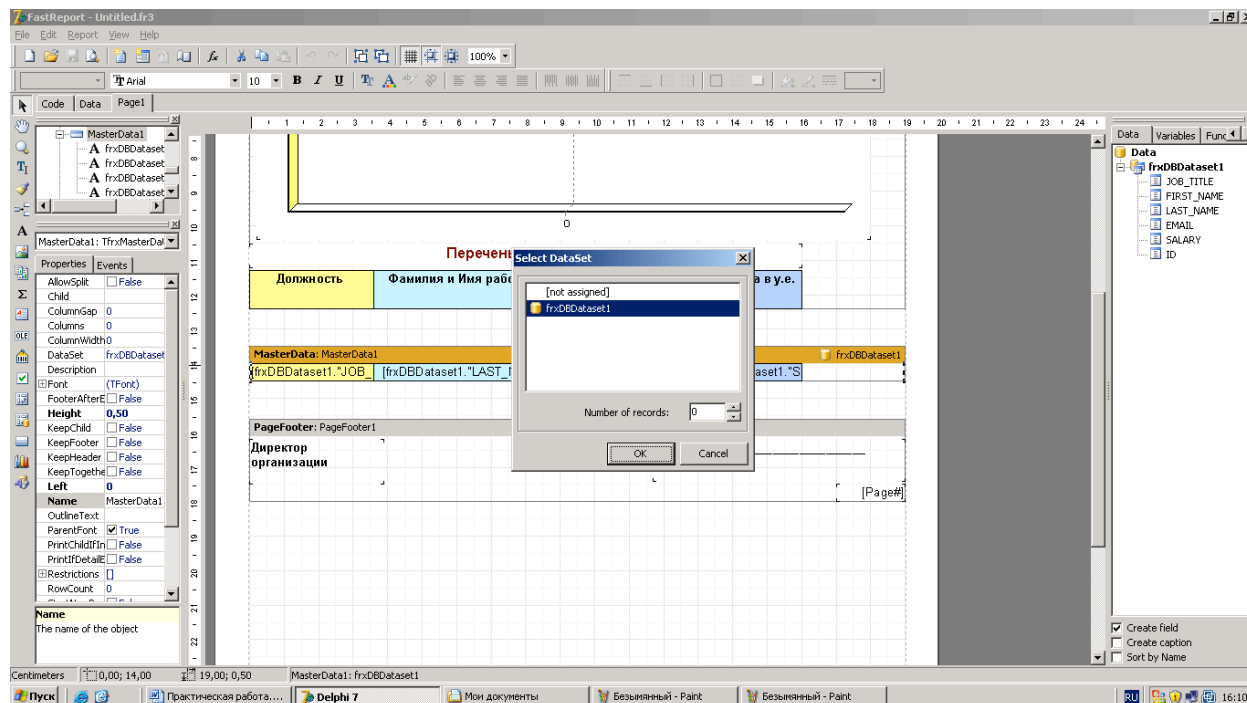


Рис. 18

Отображение данных из полей БД осуществляется простым перетаскиванием мышью необходимых полей на бэнд MasterData. Для оформления документа существует ряд компонентов в палитре слева от документа. Более подробную

информацию о средстве FastReport можно получить, скачав документацию с сайта [www.fast-report.com](http://www.fast-report.com) .

Пример 7. Работа с MS Word(файлы примера в папке C:\Примеры\test 7)

В ряде случаев (например, автоматическое генерирование писем из базы данных) возникает необходимость выгрузки информации из СУБД в редактор MS Word. В самом простом случае для этого можно воспользоваться шаблоном заготовкой в формате MS Word и средствами Delphi заменять фразы в шаблоне строками из БД.

// Функция находит текстовую константу и подставляет на ее место нужный текст

```
function FindAndInsert(FindText,ReplacementText:string):boolean;  
const wdReplaceAll=2;
```

```
begin
```

```
W.Selection.Find.Text:=FindText;
```

```
W.Selection.Find.Replacement.Text:=ReplacementText;
```

```
FindAndInsert:=W.Selection.Find.Execute(Replace:=wdReplaceAll);
```

```
End;
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
W:=CreateOleObject('Word.Application');
```

```
W.Visible:=true;
```

```
W.documents.Add(ExtractFileDir(Application.ExeName)+'\Doc1.doc');
```

```
//открываем файл шаблона
```

```
FindAndInsert('##Edit1', Edit1.Text); // Меняем слово ##Edit1 на шаблоне на  
нужную нам фразу
```

```
FindAndInsert('##Edit2', Edit2.Text);
```

```
FindAndInsert('##Edit3', Edit3.Text);
```

```
end;
```