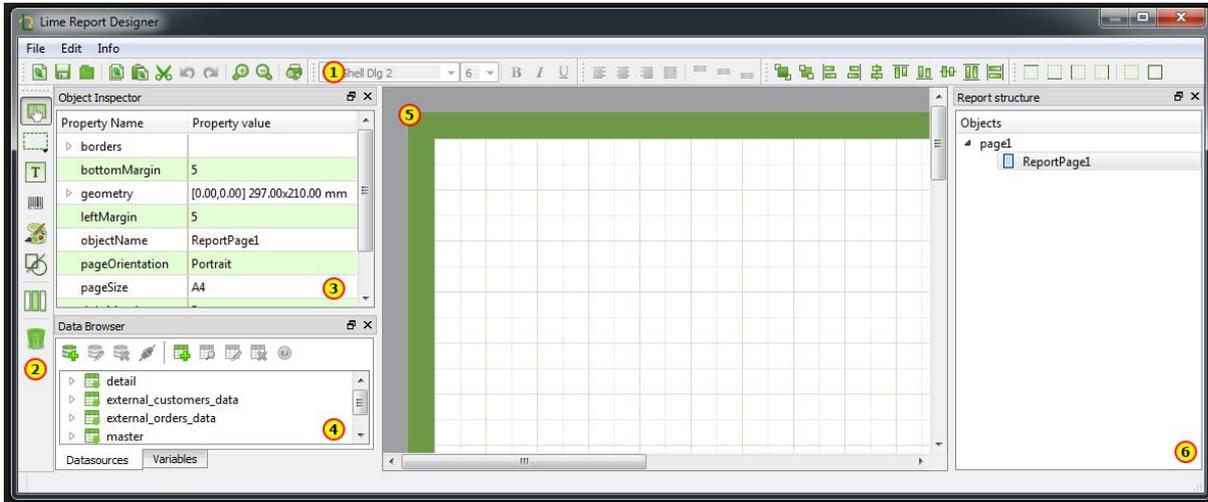


# ДИЗАЙНЕР



1. Панели инструментов
2. Панель объектов отчета
3. Обзорщик объектов
4. Обзорщик данных
5. Страница отчета
6. Обзорщик структуры отчета

## БЫСТРЫЙ ДОСТУП

Ctrl+N	Новый отчет
Ctrl+O	Загрузить отчет
Ctrl+S	Сохранить
Ctrl+Shift+S	Сохранить как
Ctrl+P	Предварительный просмотр
Ctrl+Z	Отмена изменений
Ctrl+Shift+Z	Возврат отмененных изменений
Ctrl+C	Копировать
Ctrl+X	Вырезать
Ctrl+V	Вставить
Ctrl+Arrows	Переместить выделенный объект
Shift+Arrows	Изменить размер выделенных объектов
Del	Удалить выделенные объекты
Shift+Left mouse button	Создать область выделения

## ПАНЕЛИ ИНСТРУМЕНТОВ

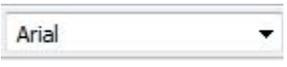
### Главная панель



	Новый отчет
	Сохранить отчет
	Загрузить отчет
	Копировать выделенное
	Вставить
	Вырезать
	Отмена изменений
	Возврат ранее отмененных изменений
	Увеличить масштаб просмотра страницы
	Уменьшить масштаб просмотра страницы
	Предварительный просмотр отчета

### Панель редактирования шрифта выделенных объектов



	Шрифт
	Размер шрифта
	Жирность
	Курсив

	Подчеркивание
---	---------------

### Панель форматирования текста



	Выравнивать текст по левому краю
	Располагать текст по центру
	Выравнивать текст по правому краю
	Выравнивать текст по ширине
	Прижимать текст к верхнему краю
	Располагать текст по центру
	Прижимать текст к нижнему краю

### Панель редактирования расположения объектов



	Переместить выделенные объекты на передний план
	Переместить выделенные объекты на задний план
	Выровнять выделенные объекты по левому краю
	Выровнять выделенные объекты по правому краю
	Выровнять выделенные объекты вертикально по центру
	Выровнять выделенные объекты по верхнему краю
	Выровнять выделенные объекты по нижнему краю
	Выровнять выделенные объекты горизонтально по центру
	Сделать выделенные объекты одной высоты

	Сделать выделенные объекты одной ширины
---	---

### Панель редактирования границ



	Обвести выделенные объекты сверху
	Обвести выделенные объекты снизу
	Обвести выделенные объекты слева
	Обвести выделенные объекты справа
	Снять границы
	Обвести объект целиком

### Панель объектов отчета



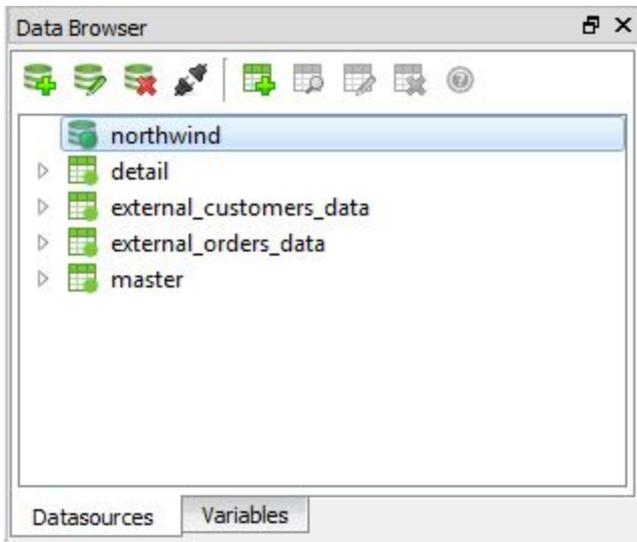
	Выбрать объект
	Вставить бэнд
	Вставить текстовую надпись
	Вставить штрих код (Требуется библиотека Zint)
	Вставить картинку
	Вставить фигуру
	Объединить выделенные объекты в горизонтальный лайаут
	Удалить объект

## ИСТОЧНИКИ ДАННЫХ

В LimeReport предусмотрено несколько источников данных:

- Переменные, объявленные в отчете и доступные внешнему приложению.
- Наборы данных на основе SQL-запросов, использующие соединение с БД. Соединение с БД может быть инициализировано:
  - непосредственно из генератора отчетов
  - внешним приложением.
- Подключение внешних наборов данных через передачу генератору отчетов объекта, реализующего QAbstractItemModel.
- Реализация на стороне приложения методов передачи данных и подключение их к генератору отчетов посредством механизма SIGNAL-SLOT.

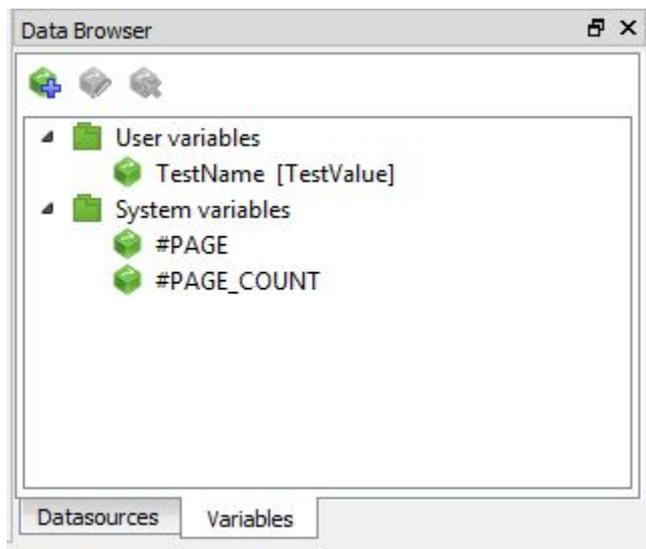
### Инструментальное окно “Data Browser”



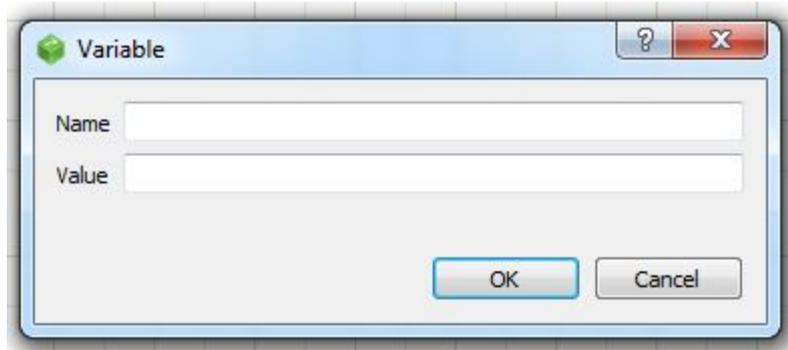
Данное окно предназначено для управления источниками данных. С его помощью задаются параметры соединений, переменных и наборов данных, которые впоследствии будут использованы в процессе построения отчета.

## Объявление переменных

Параметры переменных можно установить на закладке “Variables” в инструментальном окне “Data Browser”.



Добавление переменной производится кнопкой , далее посредством диалога “Variable” можно задать имя и значение переменной (значение переменной может быть изменено внешним приложением)

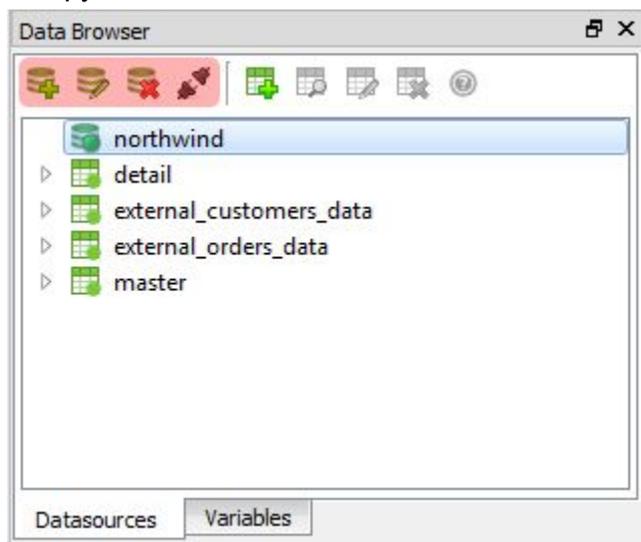


Используя кнопку  можно отредактировать объявление переменной.

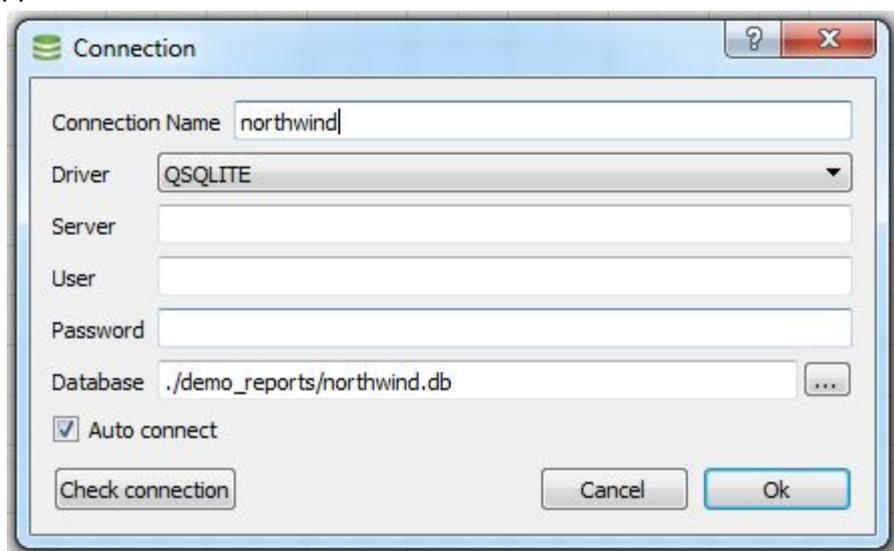
Кнопка  служит для удаления переменной.

## Соединение с базой данных непосредственно из генератора отчетов

Параметры соединения с БД устанавливаются с помощью диалогового окна "Connection", вызываемого посредством кнопки  (Add database connection), расположенной на панели инструментального окна "Data Browser".



### Диалог "Connection"



С помощью данного диалога настраиваются параметры соединения, такие как:

Connection Name - Название соединения для последующей идентификации в наборах данных, построенных на основе SQL запросов

Driver - Драйвер, посредством которого будет установлено соединения с базой данных

Server - Адрес сервера, с которым будет установлено соединение

User - Имя пользователя

Password - Пароль

Database - Имя базы данных, к которой будет осуществляться подключение

Auto connect - Параметр, влияющий на то, будет ли автоматически устанавливаться соединение с БД непосредственно после загрузки отчета (создания описателя соединения). В

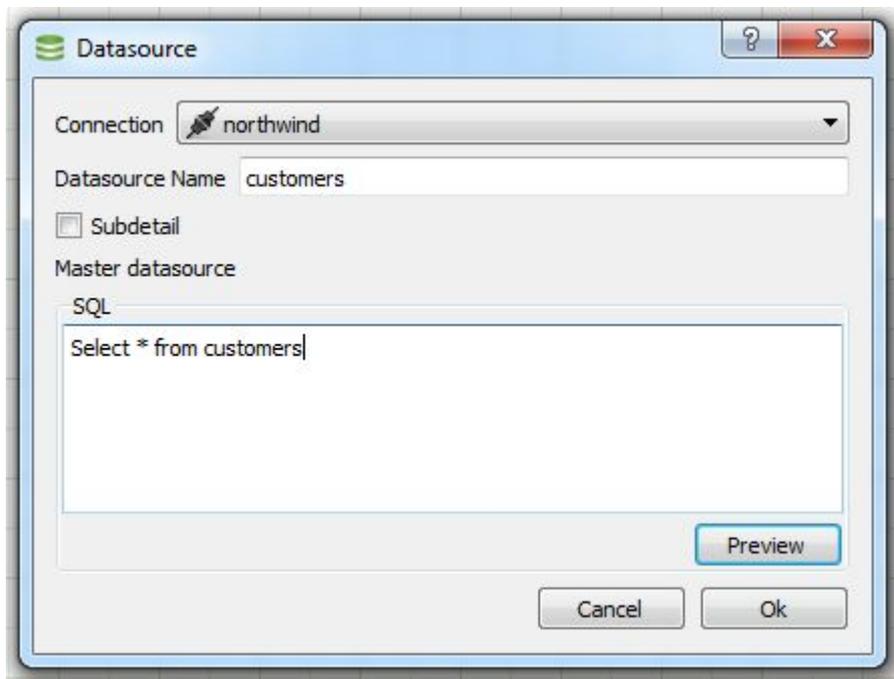
случае, если параметр не отмечен, соединение будет устанавливаться в процессе генерации отчета или может быть принудительно активировано кнопкой  . Кнопка “Check connection” позволяет проверить правильность заполнения параметров путем попытки установления соединения с указанными параметрами и вывода сообщения о результатах этой попытки.

После завершения конфигурации соединения параметры могут быть изменены с помощью кнопки  . Для удаления соединения служит кнопка  .

### Создание наборов данных в отчете

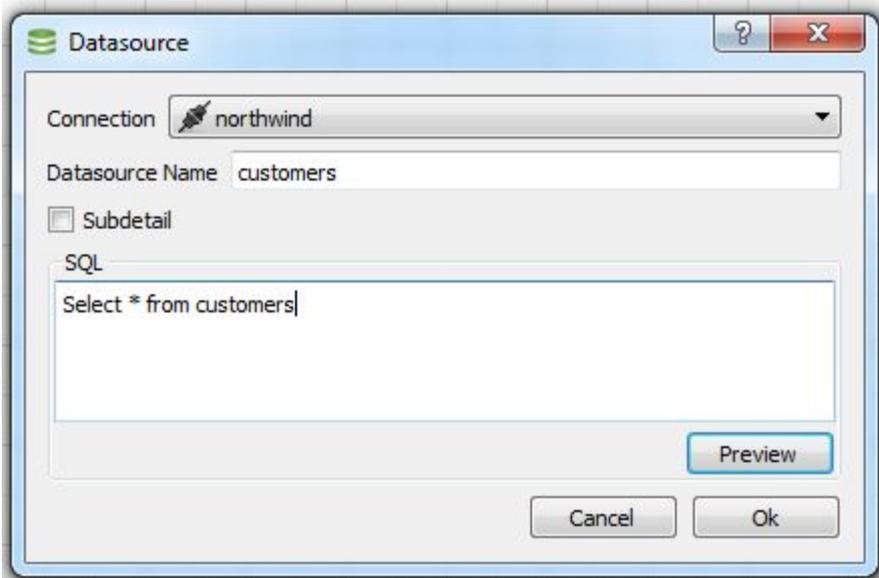
Параметры набора данных могут быть установлены с помощью диалогового окна “Datasource”, вызываемого кнопкой  , расположенной на панели инструментального окна “Data Browser”.

Диалог “Datasource”



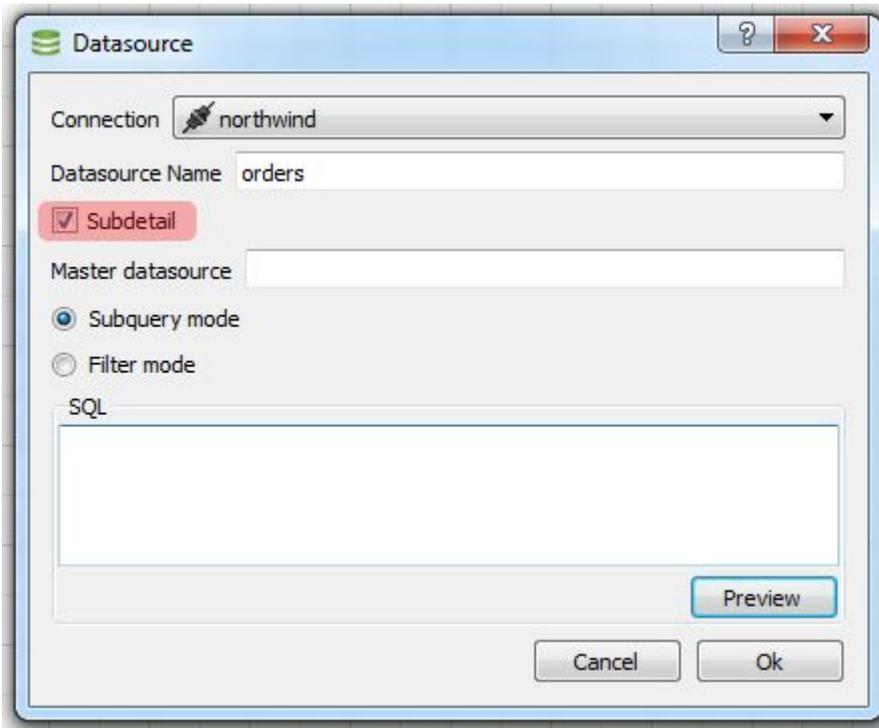
С помощью данного диалога возможно настроить несколько типов источников данных:

## 1. Источник данных, базирующийся на обычном SQL запросе к базе данных



Обычный запрос к базе данных может быть как независимым SQL запросом, так и запросом с параметрами, в качестве значений которых будут выступать переменные отчета. Обращение к переменным осуществляется посредством синтаксиса  $\$V\{\text{ИмяПеременной}\}$

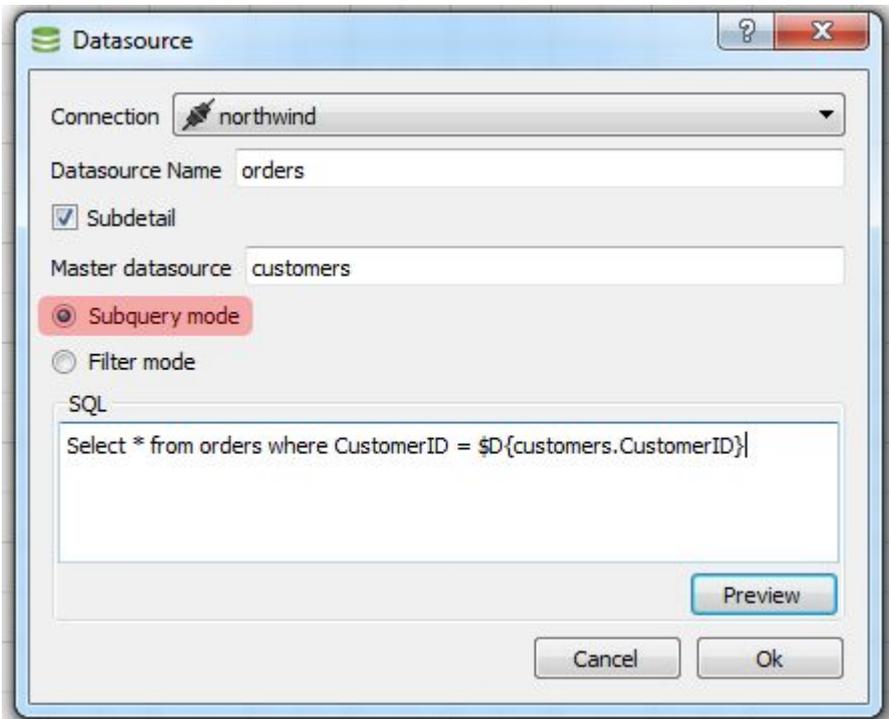
## 2. Подчиненный (связанный) набор данных



Подчиненный набор данных может использовать в качестве своих параметров другой источник данных (главный набор данных). *Любой другой источник данных может выступить в качестве главного. Главный источник данных задается параметром "Master datasource".* Во время генерации отчета перемещение по главному набору данных сопровождается одновременным обновлением связанного набора. Для того, чтобы включить режим связанного набора, необходимо отметить чекбокс "Subdetail" в окне "Datasource"

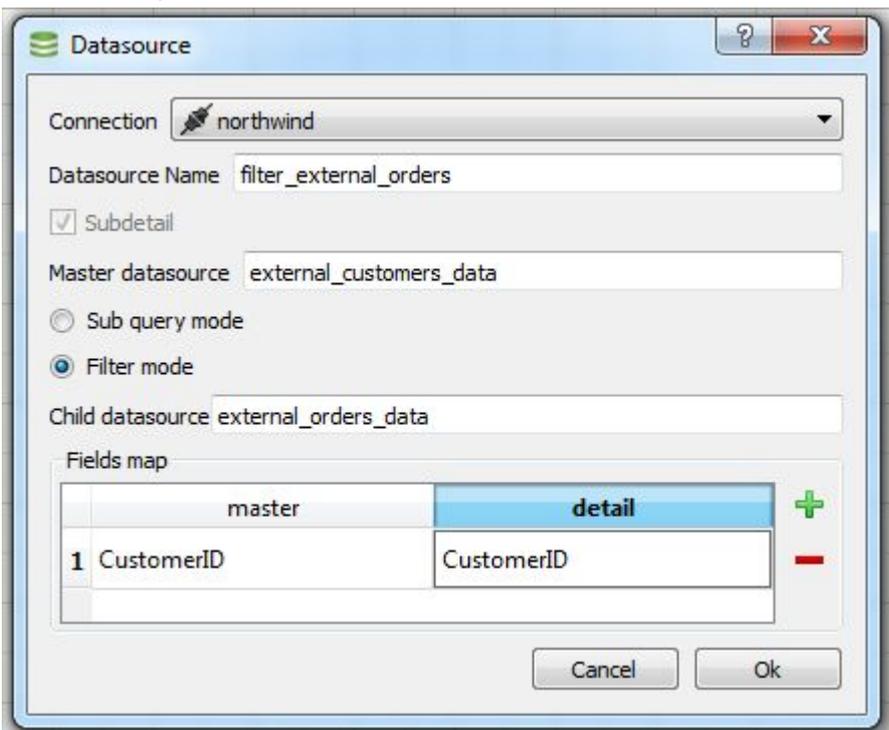
Подчиненные наборы могут быть двух видов:

## 2.1. Подчиненный SQL запрос



Для включения режима подчиненного SQL запроса необходимо выбрать "Sub query mode". Для доступа к полям главного набора данных используется синтаксис `$D{ИмяНабораДанных.ИмяПоля}`

## 2.2. Фильтрующий набор данных



Фильтрующий набор данных предназначен для случая, когда во время генерации отчета перемещение по главному набору данных должно приводить к фильтрации значений дочернего набора данных.

## Внешние источники данных

Внешние источники данных могут быть двух видов:

1. Источник, реализующий интерфейс QAbstractDataModel. Например, QStringListModel. Подключение источника данного типа производится при помощи метода `report->dataManager()->addModel(QString name, QAbstractDataModel* model, bool owned):`

### Пример:

```
QStringList simpleData;
simpleData << "value1" << "value2" << "value3";
QStringListModel* stringListModel = new QStringListModel();
stringListModel->setStringList(simpleData);
report->dataManager()->addModel("string_list",stringListModel,true);
```

2. Источник, подключенный с помощью механизма SIGNAL-SLOT.

Для использования данного метода подключения необходимо:

1. Реализовать метод (слот) получения данных с параметрами (`LimeReport::CallbackInfo info, QVariant &data`), где параметр `info` в поле `info.dataType` (`info.dataType`) содержит информацию о том, какие данные должен вернуть этот метод через изменяемый параметр `data`:
  - a. `LimeReport::CallbackInfo::IsEmpty` - `data (bool)` - есть ли данные в источнике
  - b. `LimeReport::CallbackInfo::HasNext` - `data(bool)` - есть ли следующая строка в источнике
  - c. `LimeReport::CallbackInfo::ColumnHeaderData` - `data(QString)` - наименование для колонки с индексом, указанным в `info.index` - это значение будет в дальнейшем использоваться для идентификации этой колонки в отчете и в методе получения значений
  - d. `LimeReport::CallbackInfo::ColumnData` - `data (QVariant)` - значение для колонки с наименованием, указанным в `info.columnName` (это значение ранее было установлено через вызов метода с параметром `LimeReport::CallbackInfo::ColumnHeaderData`)
2. Реализовать метод (слот) перемещения по набору данных с параметрами (`const LimeReport::CallbackInfo::ChangePosType &type, bool &result`). Параметр `type` указывает, какого типа должно быть осуществлено перемещение, а изменяемый параметр `result` должен быть установлен в значение `true`, если перемещение было успешным, и `false` в обратном случае. Параметр `type` может принимать следующие значения:
  - a. `First` - перейти к первой позиции набора данных
  - b. `Next` - перейти к следующей позиции
3. Создать новый (`ICallbackDatasource`) источник данных с помощью метода `report->dataManager()->createCallbackDatasouce()` и подключить ранее созданные методы к сигналам: `getCallbackData` - первый метод, `changePos` - второй.

### Пример:

```
class MainWindow : public QMainWindow
{
...
 QSqlQuery* m_customers;
```

```

.....
}

void MainWindow::prepareData(QSqlQuery* ds, LimeReport::CallbackInfo info, QVariant &data)
{
    switch (info.dataType) {
        case LimeReport::CallbackInfo::IsEmpty:
            data = !ds->first();
            break;
        case LimeReport::CallbackInfo::HasNext:
            data = ds->next();
            if (data.toBool()) ds->previous();
            break;
        case LimeReport::CallbackInfo::ColumnHeaderData:
            if (info.index < ds->record().count())
                data = ds->record().fieldName(info.index);
            break;
        case LimeReport::CallbackInfo::ColumnData:
            data = ds->value(info.columnName);
            break;
    }
}

void MainWindow::slotGetCallbackData(LimeReport::CallbackInfo info, QVariant &data)
{
    if (!m_customers) return;
    prepareData(m_customers, info, data);
}

void MainWindow::slotChangePos(const LimeReport::CallbackInfo::ChangePosType &type, bool &result)
{
    QSqlQuery* ds = m_customers;
    if (!ds) return;
    if (type == LimeReport::CallbackInfo::First) result = ds->first();
    else result = ds->next();
    // В этом методе может быть реализовано обновление зависимого источника данных
    m_orders->bindValue("id",m_customers->value("CustomerID"));
    m_orders->exec();
}

{
    ...
    LimeReport::ICallbackDatasource * callbackDatasource = report->dataManager()->createCallbackDatasouce();
    connect(callbackDatasource, SIGNAL(getCallbackData(LimeReport::CallbackInfo,QVariant&)),
            this, SLOT(slotGetCallbackData(LimeReport::CallbackInfo,QVariant&)));
    connect(callbackDatasource, SIGNAL(changePos(const LimeReport::CallbackInfo::ChangePosType&,bool&)),
            this, SLOT(slotChangePos(const LimeReport::CallbackInfo::ChangePosType&,bool&)));
    report->dataManager()->addCallbackDatasource(callbackDatasource,"master");
    ...
}

```

## ЭЛЕМЕНТЫ ОТЧЕТА

### 1. - БЭНД

Элемент “контейнер” предназначен для размещения других элементов отчета. Бэнд может быть нескольких типов:

1. Report Header - заголовок отчета
2. Report Footer - завершение отчета
3. Page Header - верхний колонтитул страницы отчета
4. Page Footer - нижний колонтитул страницы отчета
5. Data - данные отчета
6. SubDetail - подчиненные данные отчета
7. SubDetailHeader - заголовок подчиненных данных
8. SubDetailFooter - завершение подчиненных данных
9. GroupHeader - заголовок группы
10. GroupFooter - завершение группы

Общие для всех бэндов свойства:

autoHeight	Автоматический подбор высоты
backgroundColor	Цвет заливки бэнда
borders	Границы
geometry	Размер
keepBottomSpace	Сохранять отступ от нижней границы бэнда
objectName	Имя объекта
printIfEmpty	Печатать, если на бэнде нет данных
splittable	Разделить бэнд, если он не влезает на страницу

### Report Footer

maxScalePercent	Максимальный процент, на который можно уменьшить бэнд, если он не влезает на страницу. Если бэнд даже после сжатия не влезает на страницу, он будет перенесен полностью или частично, в зависимости от настроек бэнда
-----------------	---

### Data

datasource	Источник данных. Data бэнд будет сформирован для каждой строки в источнике данных
keepFooterTogether	Если Report Footer не влезает на страницу, то он будет перенесен на следующую страницу совместно с последним экземпляром Data бэнда
sliceLastRow	Указывает генератору отчетов на то, можно ли разрезать последний экземпляр Data бэнда или его нужно перенести целиком

**SubDetail**

datasource	Источник данных. SubDetail бэнд будет сформирован для каждой строки в источнике данных
------------	--

**SubDetailHeader, SubDetailFooter**

printAlways	Печатать, даже если SubDetail пустой
-------------	--------------------------------------

**GroupHeader**

groupFieldName	Поле, по которому осуществляется группировка. Экземпляр GroupHeader будет формироваться при смене значения в этом поле
----------------	--

2. **T** - ТЕКСТ

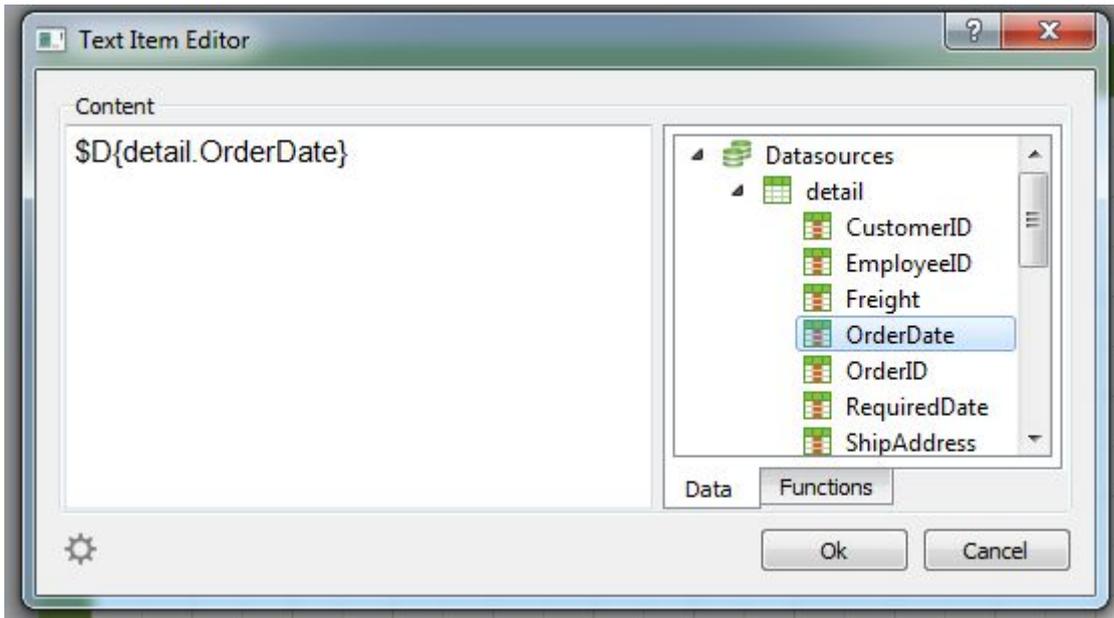
Элемент Текст служит для вывода надписей или содержимого полей источников данных.

## Параметры элемента Текст

alignment	Выравнивание текста по вертикали и горизонтали
angle	Поворот надписи
autoHeight	Автоматический подбор высоты
autoWidth	Автоматический подбор ширины
backgroundColor	Цвет заливки
backgroundMode	Режим заливки
backgroundOpacity	Прозрачность заливки
borders	Границы
content	Содержимое объекта текст
font	Шрифт
fontColor	Цвет шрифта
foregroundOpacity	Прозрачность шрифта
geometry	Размер и расположение объекта
itemLocation	Расположение объекта (Страница или Бэнд)
margin	Отступы
objectName	Имя объекта
stretchToMaxHeight	Устанавливать значение высоты самого высокого объекта на бэнде

trimValue	Обрезать пробелы в начале и в конце надписи.
-----------	--

Для редактирования параметра “content” используется “Text Item Editor” вызов, которого осуществляется двойным кликом на элементе Текст.



“Content” может содержать: Текст, Значения переменных, Значения полей из набора данных, а также значения, формируемые посредством исполнения скрипта. Для вывода значений переменных используется синтаксис  $\$V\{\text{имяПеременной}\}$ , для вывода значения поля данных  $\$D\{\text{имяНабораДанных.ИмяПеременной}\}$ , для исполнения скрипта  $\$S\{\text{телоСкрипта}\}$

### 3. - ШТРИХ КОД

Элемент для отображения штрих кодов

angle	Угол поворота штрих кода
backgroundColor	Цвет заливки
barcodeType	Тип штрих кода
barcodeWidth	параметр для QZint
borders	Границы
content	Значение для отображения штрих кодом
foregroundColor	Цвет штрих кода
geometry	Размер и расположение
itemLocation	Расположение объекта (Страница или Бэнд)
objectName	Имя объекта
pdf417CodeWords	параметр для QZint

securityLevel	параметр для QZint
stretchToMaxHeight	Устанавливать значение высоты самого высокого объекта на бэнде
testValue	Значение которое отображает штрих код в режиме разработки
whitespace	Свободное пространство

#### 4. - ИЗОБРАЖЕНИЕ

Элемент для вывода изображений

autoSize	Подгонять размер под размер изображения
borders	Границы
content	
datasource	Источник данных
field	Поле данных
geometry	Размер и расположение
image	Изображение
itemLocation	Расположение объекта (Страница или Бэнд)
objectName	Имя объекта
stretchToMaxHeight	Устанавливать значение высоты самого высокого объекта на бэнде

#### 5. - ФИГУРА

Элемент для вывода фигур

borders	Границы
geometry	Размер и расположение
itemLocation	Расположение объекта (Страница или Бэнд)
lineWidth	Толщина линии
objectName	Имя объекта
opacity	Прозрачность
penStyle	Стиль линии
shape	Фигура
shapeBrush	Стиль заливки

shapeBrushColor	Цвет заливки
shapeColor	Цвет фигуры
stretchToMaxHeight	Устанавливать значение высоты самого высокого объекта на бэнде

6.  - ГОРИЗОНТАЛЬНЫЙ ЛАЙОУТ

Элемент позволяющий объединять несколько элементов в группу

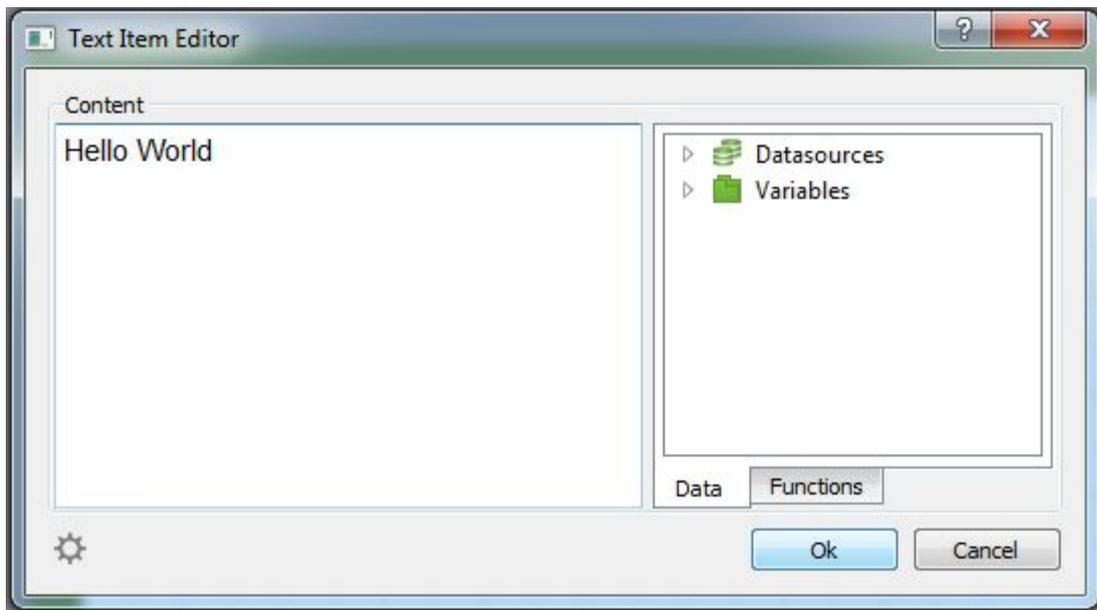
borders	Границы
geometry	Размер и расположение
itemLocation	Расположение объекта (Страница или Бэнд)
objectName	Имя объекта
stretchToMaxHeight	Устанавливать значение высоты самого высокого объекта на бэнде

## ПОСТРОЕНИЕ ОТЧЕТОВ

### 2.1. Hello World

По традиции самый первый и простой отчет "Hello World"

Создаем новый отчет . Кликаем на объект "Текст"  на панели объектов. Перемещаем указатель мыши на нужное место на листе, и опять нажимаем клавишу мыши. Дважды кликаем на только что созданном объекте, тем самым активировав окно редактора.



Набираем текст "Hello World" и нажимаем кнопку ОК



Отчет готов. Для его просмотра можно воспользоваться кнопкой предварительного просмотра на панели инструментов или пунктом меню File | Render Report, также можно воспользоваться сочетанием клавиш Ctrl+P. Вы увидите окно предварительного просмотра с единственной страницей отчета, которая содержит надпись "Hello World!". 

## 2.2. Объект текст

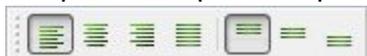
Объект "Текст" обладает очень широкими возможностями. Он умеет отображать текст, рамку, заливку. Текст может быть отображен любым шрифтом, любого размера, цвета и стиля. Большинство настроек делаются визуально с помощью панелей инструментов.

Приметы оформления текста

Hello World	Hello World	Hello World
Hello World	<b>Hello World</b>	<i>Hello World</i>
Hello World	Hello World	World Hello

Для знакомства с объектом "Текст" создадим его и разместим в нем некоторый текст:  
*Это первая строка демонстрационного текста*  
*А это вторая*

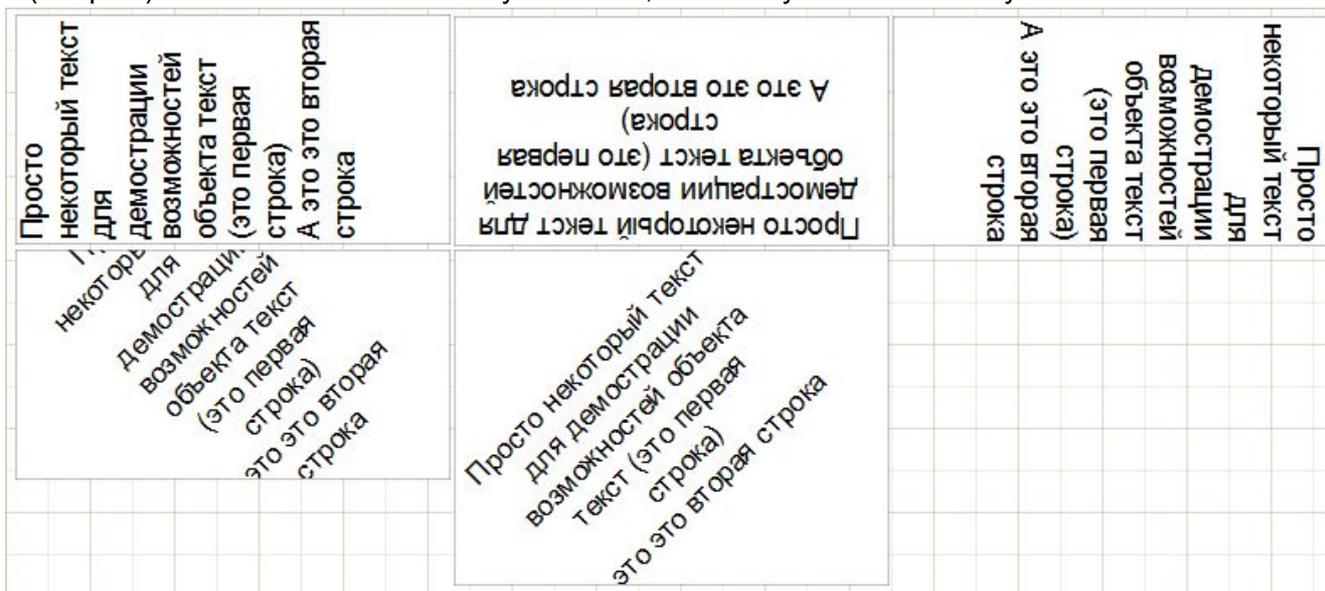
Теперь посмотрим, как работает выравнивание текста внутри объекта.



Кнопки выравнивания расположены на панели инструментов и позволяют независимо задать выравнивание текста по горизонтали и по вертикали. Обратите внимание на кнопку "Выравнивание по ширине"  - она позволяет выровнять параграф по обоим краям объекта.

Просто некоторый текст для демонстрации возможностей объекта текст (это первая строка) А это это вторая строка	Просто некоторый текст для демонстрации возможностей объекта текст (это первая строка) А это это вторая строка	Просто некоторый текст для демонстрации возможностей объекта текст (это первая строка) А это это вторая строка
Просто некоторый текст для демонстрации возможностей объекта текст (это первая строка) А это это вторая строка	Просто некоторый текст для демонстрации возможностей объекта текст (это первая строка) А это это вторая строка	Просто некоторый текст для демонстрации возможностей объекта текст (это первая строка) А это это вторая строка

Весь текст может быть повернут на 45, 90, 180, 270 и 315 градусов. При повороте на значения, отличные от 90, 180, 270, текст может вылезти за пределы объекта, как в нашем случае (см. рис.). Чтобы текст полностью уместился, немного увеличим высоту объекта.



Объект “Текст” понимает HTML теги

Например:

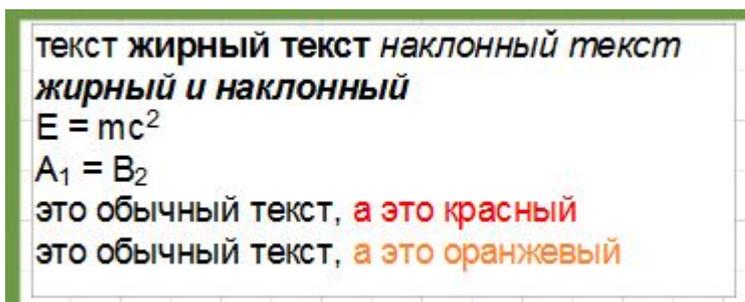
текст **жирный текст** *наклонный текст* **жирный и наклонный**

$E = mc^2$

$A_1 = B_2$

это обычный текст, **а это красный**

это обычный текст, **а это оранжевый**



Отображение выражений с помощью объекта "Текст"

Одна из самых главных особенностей этого универсального объекта - это возможность отображения не только статического текста, но и выражений. Причем, выражения могут располагаться в объекте вперемешку с текстом. Рассмотрим простой пример - поместим в объект "Текст" следующую строку:

*Привет! Сегодня  $\$S\{now()\}$*

Если запустить отчет на построение, мы увидим приблизительно следующее:

*Привет! Сегодня 2015-08-03*

Что произошло? В процессе построения отчета LimeReport встретил в тексте выражение, заключенное  $\$S\{\}$ , вычислил его и вставил полученное значение обратно в текст, убрав, разумеется, скобки. Объект "Текст" может содержать любое количество выражений, смешанных с обычным текстом. В скобки можно заключать и одиночные переменные, и выражения, например,  $\$S\{1+2*(3+4)\}$ . В выражениях можно использовать константы, переменные ( $\$V\{\}$ ), функции, поля БД ( $\$D\{\}$ ).

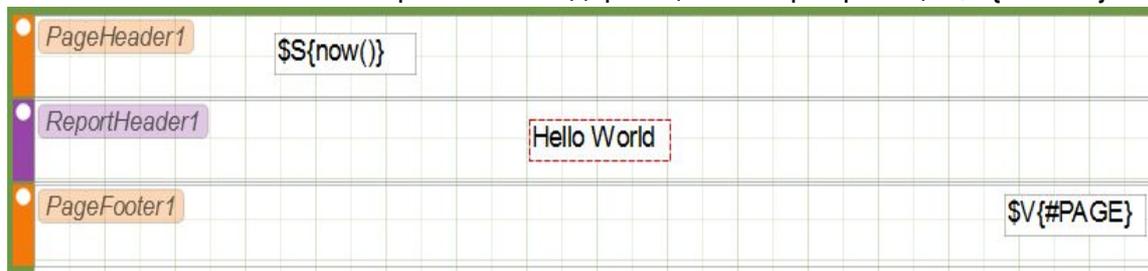
### 2.3. Использование бэндов

Бэнды применяются для логической группировки объектов. Так, разместив объект на бэнде типа "Page Header", мы тем самым говорим LimeReport, что данный объект надо вывести на каждой странице готового отчета вверху. Аналогичным образом бэнд "Page Footer" выводится внизу каждой страницы, со всеми лежащими на нем объектами. Продемонстрируем это небольшим примером. Сделаем отчет, который содержит надпись "Hello!" вверху страницы, текущую дату вверху справа и номер страницы внизу справа.

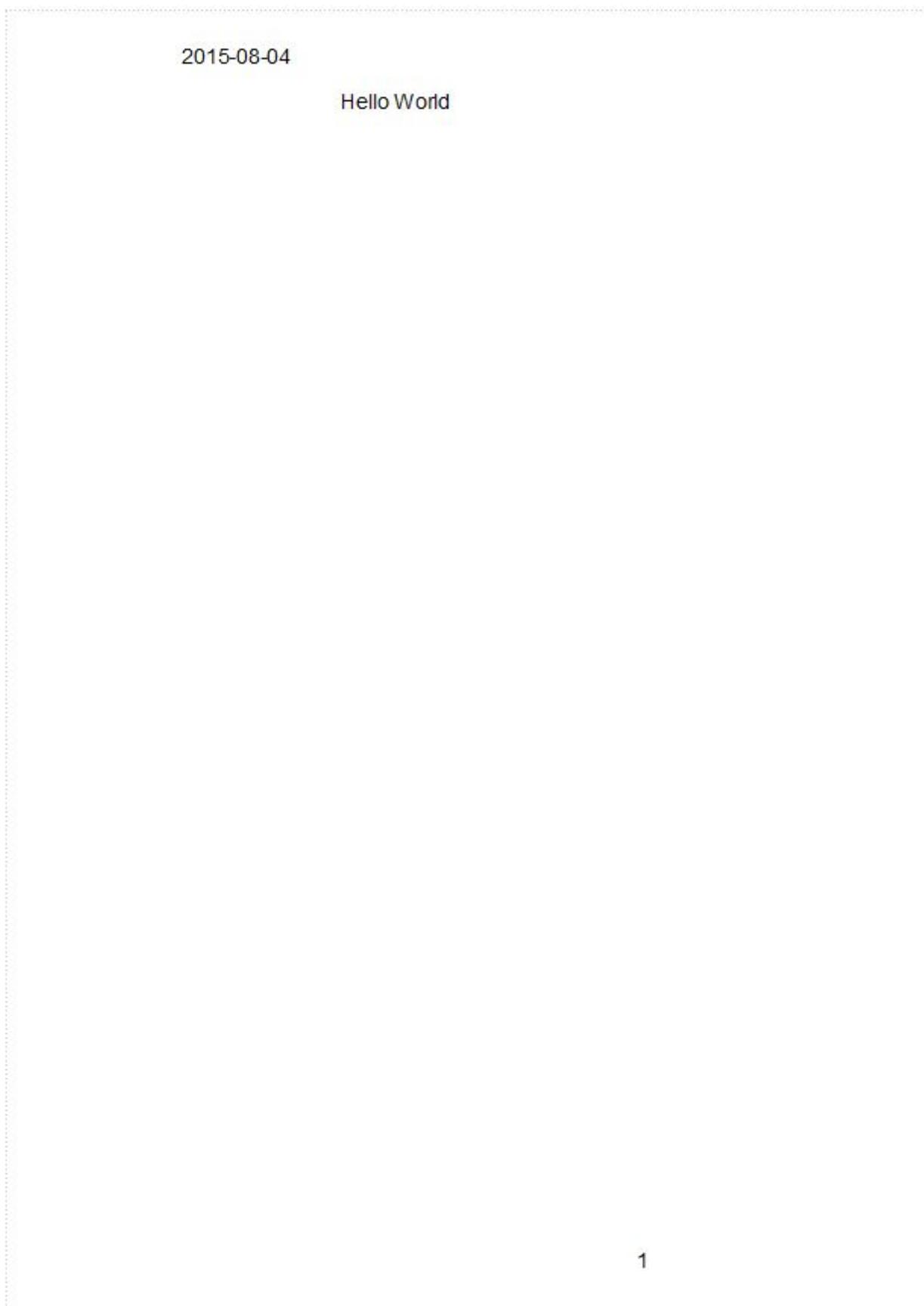
Для этого на панели объектов щелкните кнопку "Вставить бэнд"  и из открывшегося списка выберите "Page Header". Мы видим, что на страницу добавился новый бэнд. Дизайнер LimeReport автоматически размещает бэнды на странице таким образом, чтобы вверху находились бэнды-заголовки, после них - бэнды-данные, и ниже всех - бэнды-завершения.

Далее мы вставляем бэнды "Report Header" и "Page Footer"

Теперь размещаем объекты. На бэнд "Page Header" помещаем объект "Текст" и в его редакторе набираем  $\$S\{now()\}$ . На бэнд "Report Header" помещаем объект "Текст", который будет содержать текст "Hello!". А на бэнде "Page Footer", мы разместим объект текст в котом будет использована системная переменная содержащая номер страницы  $\$V\{\#PAGE\}$



Запустим отчет на выполнение и увидим, что объекты в готовом отчете разместились именно так, как нам нужно.



Итак, за размещение объектов в нужном месте отчета отвечают бэнды. В зависимости от типа бэнда мы можем расположить объект сверху или внизу страницы, на первой странице, на последней странице. Основные бэнды, которые могут нам понадобиться в большинстве

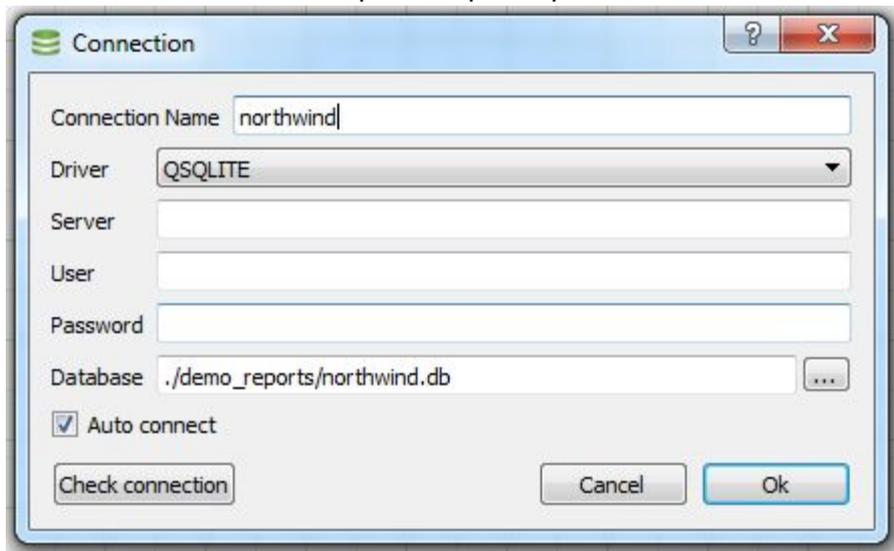
отчетов, работают следующим образом: - бэнд "Page Header" выводится в самом верху на каждой странице; - бэнд "Page Footer" выводится в самом низу на каждой странице; - бэнд "Report Header" выводится на первой странице отчета вверху, но после бэнда "Page Header" "Report Footer" выводится в самом конце отчета, на свободном месте

## 2.4 Бэнды-данные

Далее мы рассмотрим возможности выводить на печать данные из наборов данных. Что такое набор данных в данном случае? Это заранее неизвестное количество строк (записей), каждая из которых содержит определенное количество колонок (полей). Для печати такого рода информации LimeReport использует особый тип бэндов - бэнды-данные, или дата-бэнды. Это бэнды с названиями "Data" и "SubDetail". Чтобы напечатать всю таблицу или некоторые ее поля, необходимо: - добавить дата-бэнд в отчет; - подключить его к таблице; - разместить на нем объекты "Текст" с полями, которые мы хотим распечатать. При построении отчета LimeReport повторит печать бэнда столько раз, сколько записей в нашей таблице. При этом, если закончилось свободное место на странице, будут сформированы новые страницы отчета

## 2.5 Отчет "Список клиентов"

Данный отчет будет содержать данные из таблицы клиентов из демонстрационной БД "northwind". Для начала нажмем кнопку новый отчет . Далее мы создадим подключение к демонстрационной базе данных. Для этого мы воспользуемся кнопкой  (Add database connection), расположенной на панели инструментального окна "Data Browser". В открывшемся диалоге "Connection" настроим параметры подключения.



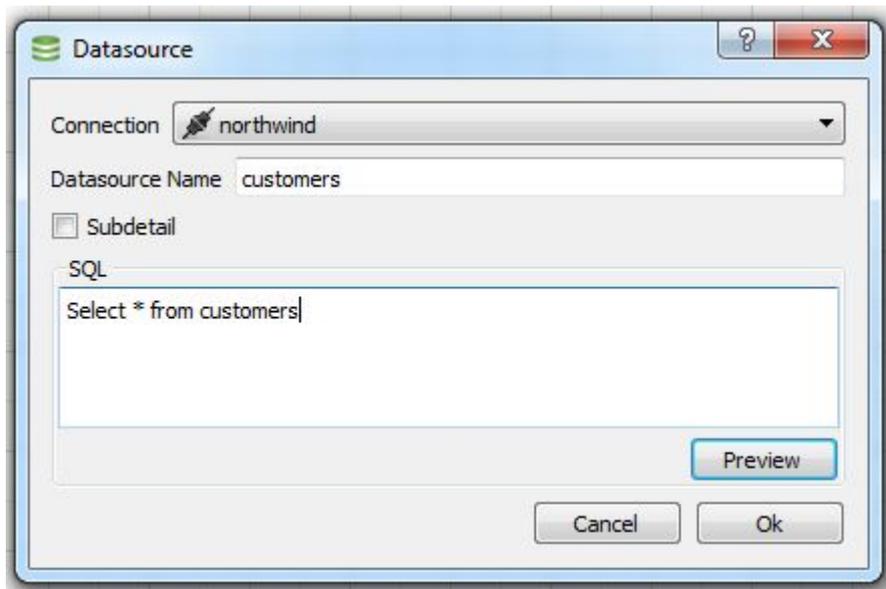
Connection Name - наименование подключения

Driver - драйвер для подключения к БД (SQLite)

DataBase - путь к базе данных

AutoConnect - автоматически подключаться после создания описателя подключения и загрузки отчета.

Далее создадим набор данных “Customers” основанный на SQL запросе. Для этого воспользуемся кнопкой  расположенной на панели инструментального окна “Data Browser”. Используя диалог “Datasource” настроим параметры источника данных



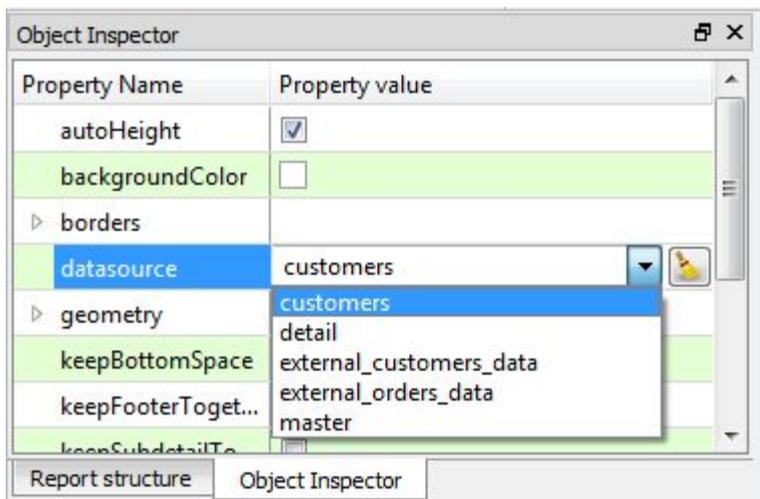
Connection - наименование подключения (Выбираем ранее созданное подключение “northwind”)

Datasource Name - наименование набора данных (Устанавливаем в “customers”)

SQL - текст SQL запроса (Набираем Select \* from customers)

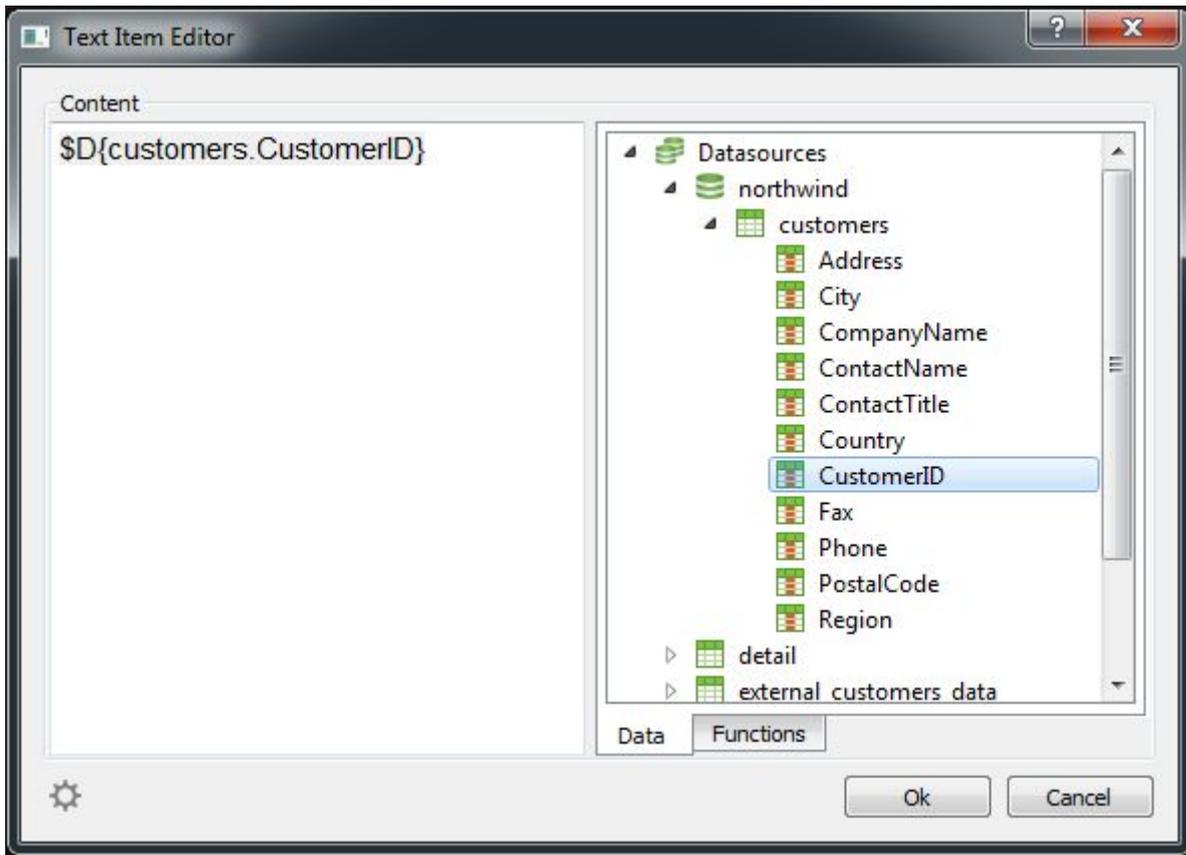
Теперь можно приступить с созданию формы отчета. Для этого мы с помощью кнопки  (Вставить бэнд) добавляем к отчету бэнд “Report Header” и размещаем на нем объект “Текст”

 содержащий текст “Список клиентов”. Далее добавляем “Data” бэнд. Устанавливаем с помощью “Обозревателя объектов” в свойство “datasource” значение “customers”



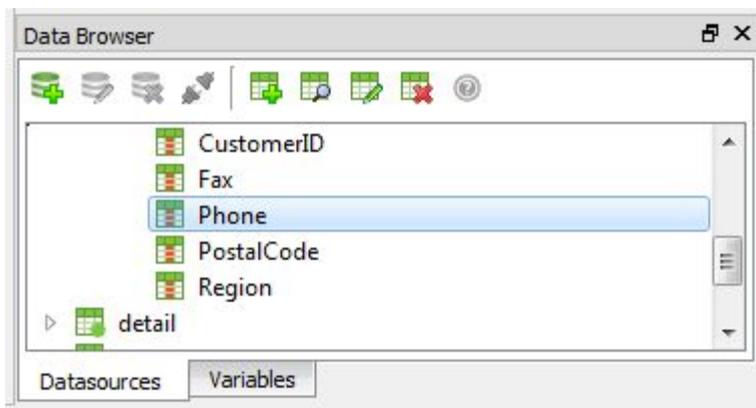
Теперь разместим на бэнде четыре объекта, которые будут отображать номер клиента, его наименование, телефон и факс. Сделаем это разными способами, чтобы продемонстрировать широкие возможности дизайнера LimeReport. Первый объект “Текст”

положим на бэнд и наберем в нем текст “\$D{customers.CustomerID}”. Это самый неудобный способ, т.к. приходится ссылку на поле писать вручную, и мы можем легко ошибиться. Чтобы облегчить вставку таких ссылок в текст, можно воспользоваться деревом источников данных расположенным в правой части редактора текстовых объектов.



Раскрыв дерево и найдя нужное нам поле осуществим двойной клик на нем и в редактор слева в текущую позицию курсора вставится нужное нам выражение \$D{customers.CustomerID}

Второй способ - drag&drop нужного поля из служебного окна "Data Browser" в отчет. Это самый простой и наглядный способ. Схватите мышкой поле "Phone" и перетащите его на бэнд.



Итак наш отчет готов

Список клиентов			
$\$D\{custo$	$\$D\{customers.CompanyName\}$	$\$D\{customers.$	$\$D\{customers.$

PageFooter1

$\$V\{\#PAGE\}$

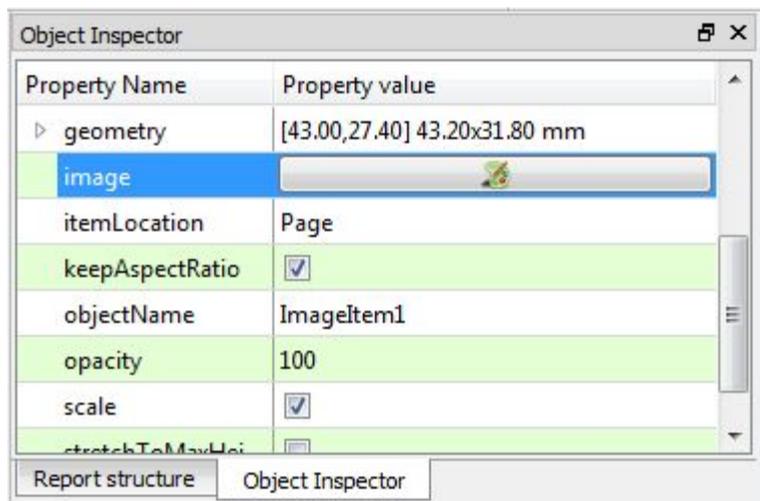
Нажимаем на предварительный просмотр и видим что получилось.

Список клиентов			
ALFKI	Alfreds Futterkiste	030-0074321	030-0076545
ANATR	Ana Trujillo Emparedados y helados	(5) 555-4729	(5) 555-3745
ANTON	Antonio Moreno Taqueraa	(5) 555-3932	
AROUT	Around the Horn	(71) 555-7788	(71) 555-6750
BERGS	Berglunds snabbkap	0921-12 34 65	0921-12 34 67
BLAUS	Blauer See Delikatessen	0621-08460	0621-08924
BLONP	Blondel pare et fils	88.60.15.31	88.60.15.32
BOLID	Balido Comidas preparadas	(91) 555 22 82	(91) 555 91 99

## 2.6 Объект "Рисунок"

Следующий объект, который мы рассмотрим - это объект "Рисунок". Он также довольно часто используется в отчетах. С помощью объекта вы можете вставить в отчет логотип вашей фирмы, фотографию сотрудника или любую другую графическую информацию.

Давайте рассмотрим возможности объекта. Создайте пустой отчет и поместите на лист отчета объект "Рисунок". В обозревателе объектов выберите свойство image и нажмите кнопку выбора и загрузки изображения.



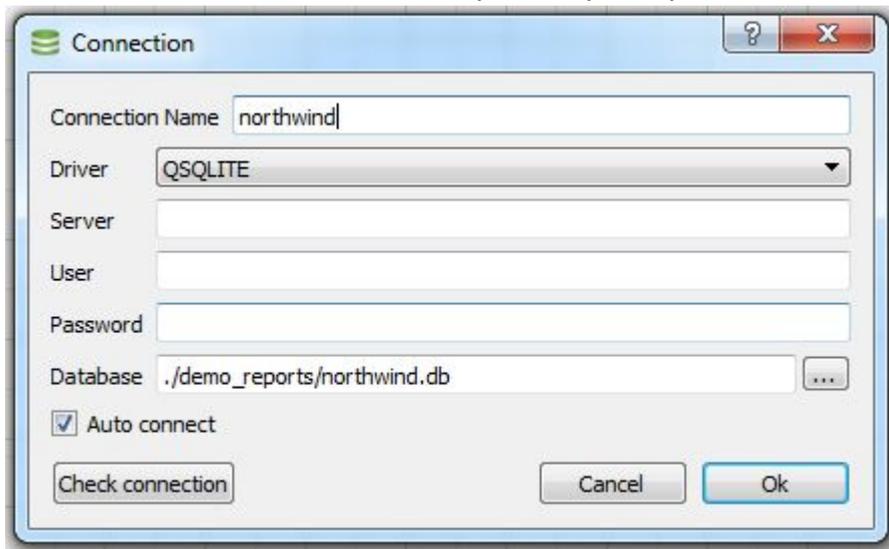
В обозревателе объекта мы также можем наблюдать следующие опции: `autoSize`, `scale` (включено по умолчанию), `center` (включено по умолчанию), `keepAspectRatio` (включено по умолчанию). Включив опцию "Авторазмер" мы увидим, что объект принял размеры, соответствующие находящемуся в нем рисунку. Иногда такая возможность бывает полезна, если надо отображать рисунки разных размеров. По умолчанию эта опция выключена, что подходит для большинства случаев. Опция "scale" включена по умолчанию, что заставляет рисунок растягиваться внутри объекта. Изменяйте размеры объекта мышкой, и вы увидите, что размер картинки все время соответствует размеру объекта. Если опцию отключить, то рисунок будет отображаться в исходных размерах. Это поведение отличается от опции "autoSize" тем, что размеры объекта не подгоняются под размер рисунка, т.е. объект можно сделать больше рисунка или меньше. Опция "center" позволяет отцентрировать рисунок внутри объекта. Опция "keepAspectRatio" включена по умолчанию и выполняет очень полезную задачу: не позволяет пропорциям рисунка искажаться при изменении размеров объекта. Эта опция работает только в паре с опцией "Растягивание". При любом изменении размеров объекта нарисованный круг останется кругом, а не превратится в овал. При этом растянутый рисунок занимает не весь внутренний объем объекта, а только часть, необходимую для отображения картинки в правильных пропорциях. Если опцию отключить, то картинка растянется на весь объем объекта, и, если размеры объекта не соответствуют исходным пропорциям картинки, картинка исказится.

## 2.7 Отчет с картинками

Объект "Рисунок", как и другие объекты в LimeReport, умеет отображать данные из БД. Подключение объекта к нужному полю БД осуществляется с помощью свойств `datasource`, `field` в обозревателе объектов. В отличие от объекта "Текст", это единственный способ подключить объект к данным. Продемонстрируем все вышесказанное примером отчета, который будет содержать изображения категорий товаров вместе с их названиями. Для этого нам опять потребуется демонстрационная база данных `nortwind`, идущая в комплекте с LimeReport.

Зайдем в дизайнер и нажмем кнопку "Новый отчет", чтобы LimeReport автоматически создал пустой шаблон.

Подключим таблицу к отчету в окне. Для этого мы воспользуемся кнопкой  (Add database connection), расположенной на панели инструментального окна "Data Browser". В открывшемся диалоге "Connection" настроим параметры подключения.



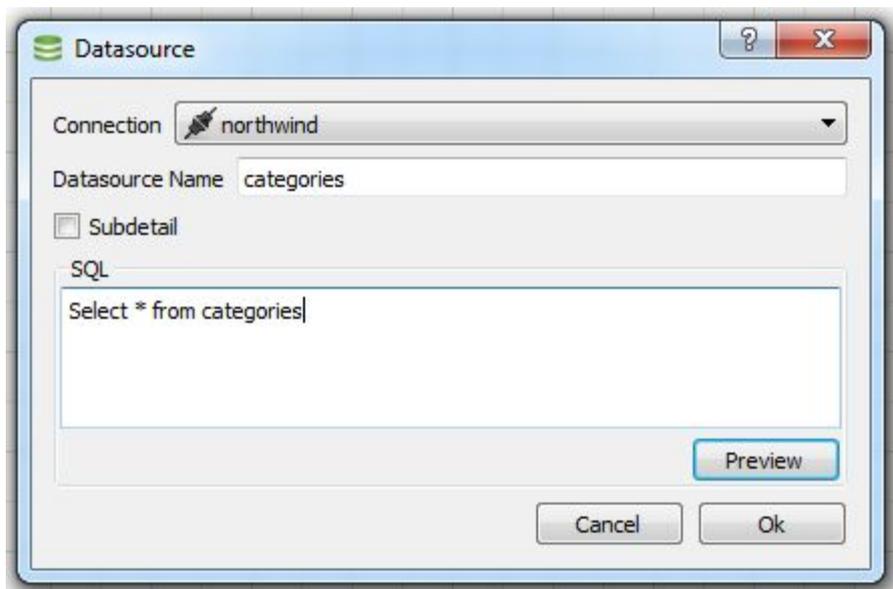
Connection Name - наименование подключения

Driver - драйвер для подключения к БД (SQLite)

DataBase - путь к базе данных

AutoConnect - автоматически подключаться после создания описателя подключения и загрузки отчета.

Далее создадим набор данных "Categories" основанный на SQL запросе. Для этого воспользуемся кнопкой  расположенной на панели инструментального окна "Data Browser". Используя диалог "Datasource" настроим параметры источника данных



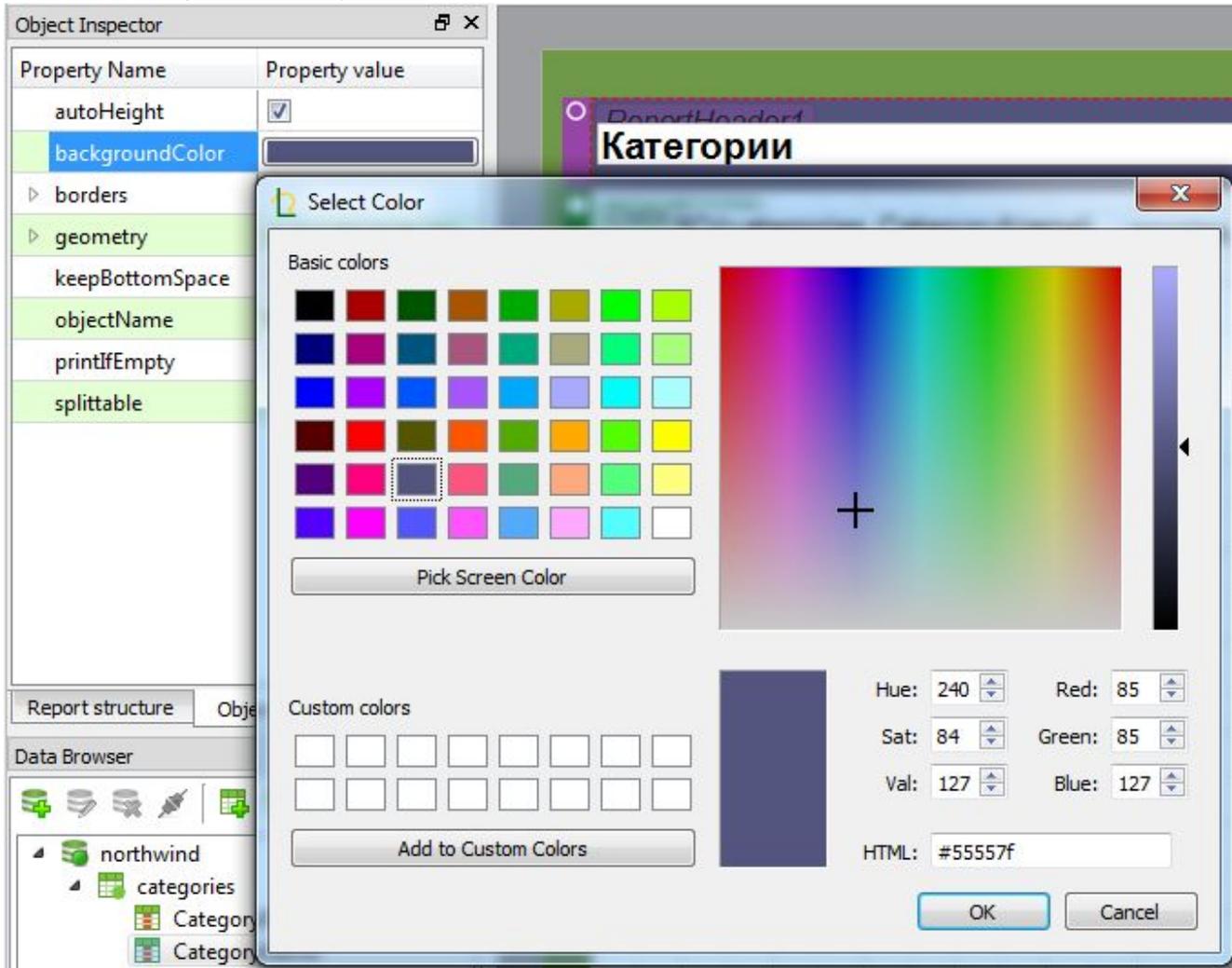
Connection - наименование подключения (Выбираем ранее созданное подключение "northwind")

Datasource Name - наименование набора данных (Устанавливаем в "categories")

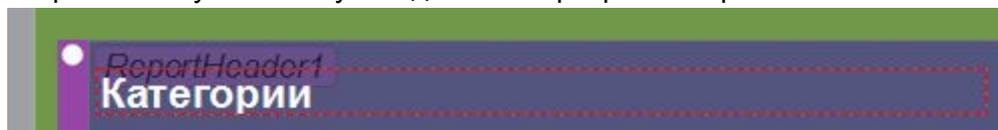
SQL - текст SQL запроса (Набираем Select \* from categories)

Теперь можно приступить с созданию формы отчета. Для этого мы с помощью кнопки  (Вставить бэнд) добавляем к отчету бэнд "Report Header" и размещаем на нем объект "Текст"

 содержащий текст "Категории". Далее выберем цвет заливки бэнда воспользовавшись свойство backgroundColor у объекта бэнд.



Как мы можем заметить объект "Текст" сильно выделяется на фоне заливки бэнда. Для того чтобы это исправить мы поменяем свойства fontColor и backgroundMode у этого объекта. Устанавливаем "backgroundMode" в "TransparentMode" а fontColor устанавливаем в белый. Таким образом получаем белую надпись на прозрачном фоне.

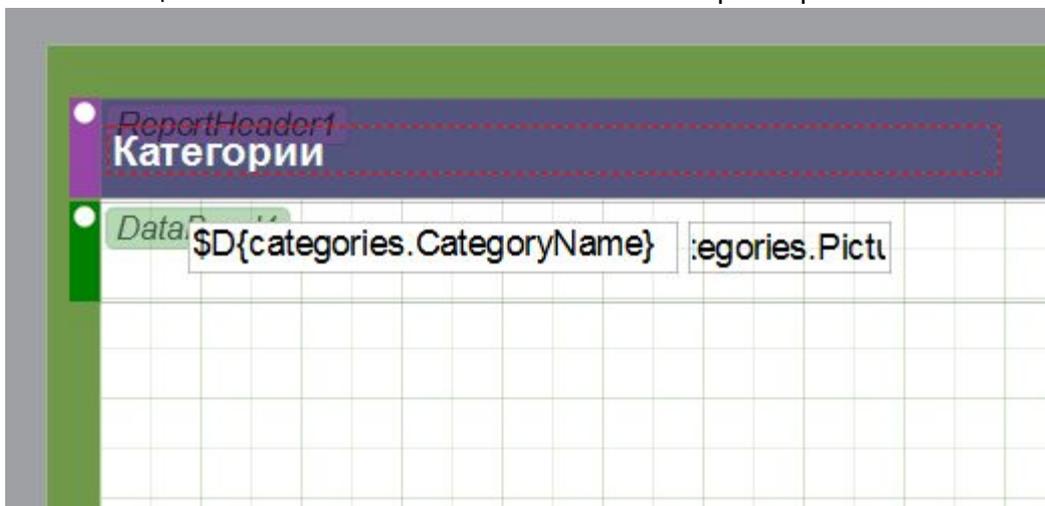


Далее добавляем  "Data" бэнд. Устанавливаем с помощью "Обозревателя объектов" в свойство "datasource" значение "categories". На бэнд положим объект "Текст"  и подключим его к полю `SD{categories.CategoryName}` одним из способов, описанных выше.

Рядом разместим объект "Рисунок"  и подключим его к полю "Picture".

datasource	categories
field	Picture
geometry	CategoryID CategoryName Description
image	Picture
itemLocation	Picture
keepAspectRatio	<input checked="" type="checkbox"/>
objectName	ImageItem1

Для этого в обозревателе объектов настроим свойства: `datasource = "categories"`, `field = "Picture"` напомним, что оба этих свойства - типа "список", поэтому нужные значения можно выбрать с помощью мыши. А так же отметим галочкой параметр `autoSize`.



Все отчет готов



## 2.8 Отображение многострочного текста

Вернемся к предыдущему примеру с категориями. В таблице "categories" есть поле "Description", которое содержит описание каждой категории. Давайте модернизируем наш отчет, добавив в него это поле. На первый взгляд все просто: добавляем на бэнд с данными объект "Текст", подключаем его к полю "Description" и устанавливаем размеры объекта. Запускаем отчет на выполнение и видим, что получилось не совсем то, чего мы ожидали:

Категории		
Beverages	Drinks, or beverages, are liquids intended for human consumption. In addition to basic	
Condiments	A condiment is a spice, sauce or other food preparation that is added to food to impart a	
Confections	Sugar confections include sweet, sugar-based foods, which are usually eaten as snack food.	

Однако, LimeReport всего лишь сделал то, что его просили сделать. Поле "Description" содержит многострочный текст, размер которого может варьироваться. А наш объект "Текст", отображающий информацию из этого поля, имеет фиксированный размер. Вот некоторые строки и не влезли в объект и были обрезаны. Как поступить в данной ситуации? Можно, конечно, подобрать размеры объекта с запасом или уменьшить размер шрифта. Однако, это приведет к неэкономному использованию места на листе: одни категории имеют длинное описание, другие - короткое. В LimeReport есть средства, позволяющие решить эту проблему. Речь идет о возможности бэнда подбирать свою высоту таким образом, чтобы вместить все имеющиеся в нем объекты. Для этого надо всего лишь включить свойство "autoHeight". Однако это не все - объект с длинным текстом и сам должен уметь растягиваться. Объект "Текст" умеет это делать. Объект может автоматически подбирать свою высоту или ширину, чтобы полностью вместить имеющийся в нем текст. Для этого служат свойства "autoWidth" и "autoHeight". Свойство "autoWidth" подбирает ширину объекта таким образом, чтобы уместились все строки, без переносов слов. Этот режим удобен, когда объект содержит единственную строку текста. Свойство "autoHeight" позволяет подобрать высоту объекта так, чтобы поместился весь текст. Ширина объекта при этом не меняется. Включите свойство "autoHeight" у объекта содержащего поле "Description". Также включите свойство "autoHeight" у бэнда.

Запустим отчет и убедимся, что теперь все работает как надо.

## Категории

### Beverages

Drinks, or beverages, are liquids intended for human consumption. In addition to basic needs, beverages form part of the culture of human society. Although all beverages, including juice, soft drinks, and carbonated drinks, have some form of water in them, water itself is often not classified as a beverage, and the word beverage has been recurrently defined as not referring to water .



An alcoholic beverage is a drink containing ethanol, commonly known as alcohol, although in chemistry the definition of an alcohol includes many other compounds. Alcoholic beverages, such as wine, beer, and liquor, have been part of human culture and development for 8,000 years.

Non-alcoholic beverages often signify drinks that would normally contain alcohol, such as beer and wine but are made with less than .5 percent alcohol by volume. The category includes drinks that have undergone an alcohol removal process such as non-alcoholic beers and de-alcoholized wines.

### Condiments

A condiment is a spice, sauce or other food preparation that is added to food to impart a particular flavor, to enhance its flavor,[1] or in some cultures, to complement the dish. The term originally described pickled or preserved foods, but has shifted meaning over time.[2]

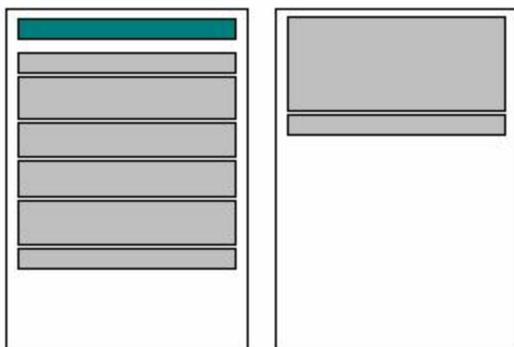


Как видим, при построении отчета LimeReport заполняет объекты данными, растягивает объекты со включенной опцией "autoHeigh" и потом подбирает высоту бэнда таким образом, чтобы уместить все объекты. Если опция "autoHeigh" у бэнда отключена, то подбор высоты бэнда не производится, и бэнд выводится с той высотой, что была установлена в дизайнера. Если мы попробуем отключить эту опцию, мы увидим, что объекты с длинным текстом по-прежнему растягиваются, а бэнды - нет, что приводит также к обрезке текста.

## 2.9 Разрыв данных

Обратим внимание на одну особенность отчета с категориями: на некоторых страницах внизу остается много пустого места. Почему это происходит? Когда отчет строится, ядро LimeReport заполняет свободное место листа бэндами. После вывода каждого бэнда текущая позиция смещается все ниже и ниже. Когда LimeReport обнаруживает, что места для вывода очередного бэнда не хватает (его высота больше, чем высота оставшегося места на листе), то формируется новая страница и вывод бэндов продолжается на ней. И так до тех пор, пока есть записи в наборе данных

Наш отчет как раз содержит объект с большим количеством текста, поэтому высота бэндов получается довольно большая. И если большой бэнд не помещается на страницу, он переносится на следующую, а внизу страницы остается много неиспользованного места. Это видно на следующем рисунке:



Чтобы рациональнее использовать бумагу, воспользуемся возможностью LimeReport разбивать содержимое бэндов на части. Все, что нужно - это включить опцию "splittable" у бэнда "Data". Мы видим, что пустого места внизу страниц отчета значительно поубавилось:



Следует отметить, что алгоритм разрыва не обеспечивает 100% качества получаемого отчета. Поэтому используйте эту опцию аккуратно.

## 2.18 Печать данных в виде таблицы

Часто бывает необходимо отобразить отчет в виде таблицы с обрамлением. Один из примеров такого отчета – это прайс-лист. Чтобы построить такой отчет в LimeReport, надо всего лишь включить обрамление у объектов, лежащих на бэнде "Данные". Рассмотрим несколько вариантов обрамления на примере тестового отчета. Создадим отчет следующего вида:

	<code>SD{categories.</code>	<code>SD{categories.CategoryName}</code>	<code>ategories.Pictur</code>
---	-----------------------------	--	-------------------------------

Разместим объекты на бэнде встык, а также уменьшим высоту бэнда до минимального размера. Первый и самый простой тип таблицы – с полным обрамлением. Для этого надо у каждого объекта включить все линии рамки:

1	Beverages	
2	Condiments	
3	Confections	
4	Dairy Products	
5	Grains/Cereals	
6	Meat/Poultry	
7	Produce	
8	Seafood	

Следующий тип обрамления – только горизонтальные или только вертикальные линии – делается аналогично, у объектов включается горизонтальное или вертикальное обрамление.

1	Beverages	
2	Condiments	
3	Confections	
4	Dairy Products	
5	Grains/Cereals	
6	Meat/Poultry	
7	Produce	
8	Seafood	

Все вышеприведенные примеры содержали бэнды, которые имели фиксированный размер. Но как вывести таблицу, если бэнд растягиваемый? Покажем это на примере. Включим свойство autoSize у объекта "Изображение". В этом случае высота бэнда будет подбираться в зависимости от размера изображения. Мы получим отчет следующего вида:

1	Beverages	
2	Condiments	
3	Confections	
4	Dairy Products	

Немного не то, что нам нужно – хотелось бы, чтобы рамки соседних объектов тоже растягивались. LimeReport позволяет легко решить эту проблему. Для построения подобных отчетов достаточно включить у всех объектов, которые должны быть растянуты, свойство "stretchToMaxHeight". При этом ядро LimeReport сначала считает максимальную высоту бэнда, затем "дотягивает" объекты с включенной опцией до нижнего края бэнда. Т.к. вместе с объектом растягивается и его рамка, в результате вид отчета меняется:

1	Beverages	
2	Condiments	
3	Confections	
4	Dairy Products	

## 2.22 Отчет с двумя уровнями данных (master-detail)

До сих пор мы рассматривали отчеты, в которых присутствовал только один дата-бэнд – "Data". Это давало возможность печатать данные из одной таблицы БД. LimeReport позволяет печатать отчеты, содержащие неограниченное количество уровней данных. Рассмотрим создание двухуровневого отчета. Он будет содержать данные из таблиц Customer и Orders. Первая таблица – это список клиентов, вторая – список заказов, сделанных клиентами. Вторая таблица содержит список всех заказов, сделанных всеми компаниями. Чтобы получить список заказов, сделанных конкретной компанией, из таблицы следует отобрать записи, у которых поле CustomerID = идентификатору выбранной компании. Отчет, построенный на таких данных, будет выглядеть следующим образом:

<b>ALFKI</b>		<b>Alfreds Futterkiste</b>
--------------	--	----------------------------

ALFKI	10062	2005-04-23
-------	-------	------------

ALFKI	10643	2005-04-16
-------	-------	------------

ALFKI	10692	2005-03-31
-------	-------	------------

ALFKI	10702	2005-02-19
-------	-------	------------

ALFKI	10835	2005-03-03
-------	-------	------------

ALFKI	10952	2005-02-16
-------	-------	------------

ALFKI	11011	2005-04-20
-------	-------	------------

<b>ANATR</b>		<b>Ana Trujillo Emparedados y helados</b>
--------------	--	---

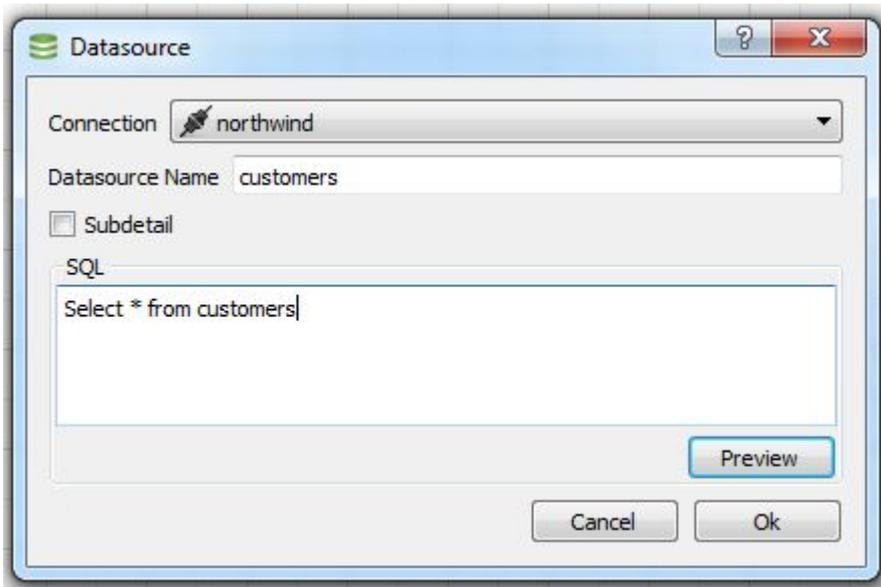
ANATR	10308	2005-04-04
-------	-------	------------

ANATR	10625	2005-04-03
-------	-------	------------

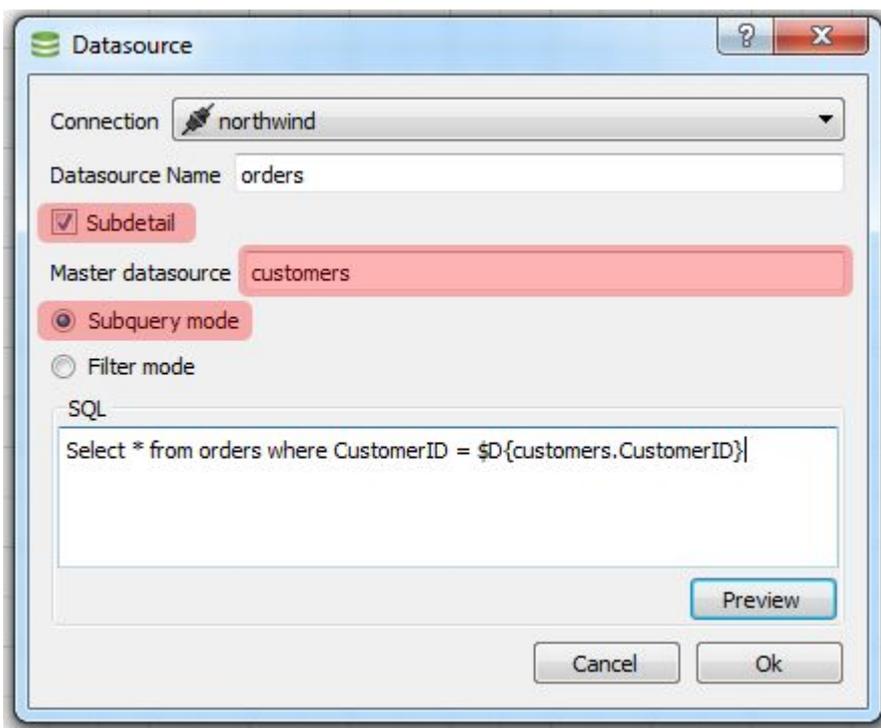
ANATR	10759	2005-02-04
-------	-------	------------

ANATR	10926	2005-05-01
-------	-------	------------

Приступим к созданию отчета. Первым делом создадим соединение с базой данных "northwind" (Аналогично тому как было показано ранее). Далее создадим набор данных "customers".



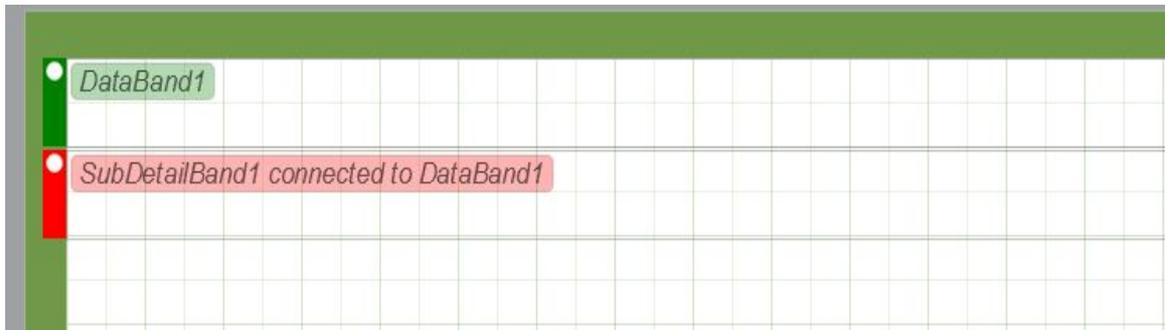
Далее создадим связанный с "customers" набор данных "orders". Обратите ваше внимание на что параметр "Subdetail" отмечен галочкой, в качестве главного набора данных, указан набор данных "customers", а также выбран режим "Subquery mode"



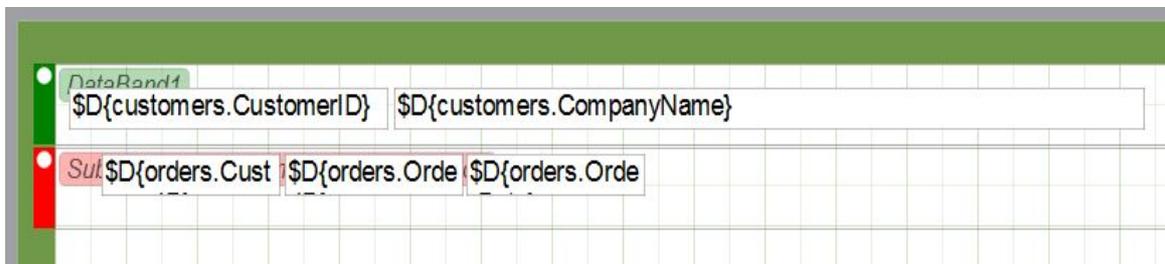
После того, как все наборы данных подключены к отчету, добавляем бэнд "Data" и подключаем её к набору данных "customers" ( параметр "datasource" устанавливается с помощью обозревателя объектов). Затем нам необходимо добавить "SubDetail" бэнд.

Нужно отметить что этот бэнд возможно вставить только в том случае если бэнд к которому "SubDetail" будет привязан активен (Левый клик мыши по "Data" бэнду). Нажимаем

вставить бэнд  и выбираем "SubDetail". Таким образом у нас должен получиться следующий отчет.



Размещаем на "DataBand1" поля "CustomerID", "CompanyName" из набора данных "customer". Далее размещаем поля "CustomerID", "OrderID", "OrderDate" из набора данных "orders". Отчет готов.



Запускаем предварительный просмотр и получаем:

ALFKI		Alfreds Futterkiste
ALFKI	10062	2005-04-23
ALFKI	10643	2005-04-16
ALFKI	10692	2005-03-31
ALFKI	10702	2005-02-19
ALFKI	10835	2005-03-03
ALFKI	10952	2005-02-16
ALFKI	11011	2005-04-20
ANATR		Ana Trujillo Emparedados y helados
ANATR	10308	2005-04-04
ANATR	10625	2005-04-03
ANATR	10759	2005-02-04
ANATR	10926	2005-05-01

Продолжение следует ....