Business intelligence with OpenOffice.org Base

BASE BUILDER

Learn how to analyze your business data with the help of OpenOffice.org Base and the Sun Report Builder extension. **BY DMITRI POPOV**

I f you own a business or work as a freelancer, analyzing your business data can be as useful as it is for corporations. Investing thousands of dollars in high-end business intelligence (BI) tools isn't required; in many cases, the OpenOffice.org Base application combined with the Sun Report Builder extension can do the job just fine.

Sun Report Builder [1] uses the same reporting engine as the Pentaho BI suite [2], and its pedigree makes it a perfect tool for analyzing your business data. OpenOffice.org Base, in turn, can connect to a variety of database systems, so you can use it to pull data from virtually any business application that uses a database and analyze it with the tools of-

THE AUTHOR

Dmitri Popov holds a degree in Russian language and computer linguistics. He has been writing exclusively about Linux and open source software for several years, and his articles have appeared in Danish, British, North American, German, and Russian magazines and websites.

fered by Sun Report Builder. Take, for example, BambooInvoice [3], an easy-to-use invoicing application based on MySQL/PHP [4]. Although it does a good job managing invoices, BambooInvoice is rather limited when it comes to analyzing the invoicing data. OpenOffice.org enters the picture here. Using the Base/Sun Report Builder combo, you can create a solution that allows you to view the invoicing data from different angles and turn numbers into easy-to-digest graphs.

Before you start, make sure that you have the Sun Java Runtime Environment and the latest version of the Sun Report Builder extension installed on your system. Connecting OpenOffice.org Base to BambooInvoices's database back end is the first order of business. To do this, you need a small piece of software, called a connector, that acts as a bridge between the MySQL-based back end of BambooInvoice and an OpenOffice.org Base database.

MySQL provides two connectors: Connector/ODBC and Connector/J [5]. In the example here, I'll use the latter because it's significantly easier to install and configure. After you download Con-

nector/J from MySQL's website, unpack the file and move the resulting *mysql-connector-java-x.x.x-bin.jar* file into the desired location (e.g., your home directory).

In OpenOffice.org, choose *Tools | Options | Java*, then make sure that the appropriate Java Runtime Environment is selected and click the *Class Path* button. Next, click the *Add Archive* button and select *mysql-connector-java-x.x.x-bin.jar*. To save the settings, click *OK*, then close the window and restart OpenOffice.org.

Now you are ready to connect OpenOffice.org to the MySQL database. In OpenOffice.org, choose *New | Database*. In the Database Wizard, select the *Connect to an existing database* option, select *MySQL* from the drop-down list, and click *Next*. Then select the *Connect using JDBC (Java Database Connectivity)* option and click *Next*. In the appropriate fields, enter the name of the database (by default, BambooInvoice uses a database called *bambooinvoice*) and the server address (Figure 1).

To make sure that the MySQL JDBC driver works properly, click the *Test class* button. When the driver has been loaded successfully, click *Next*, then enter the
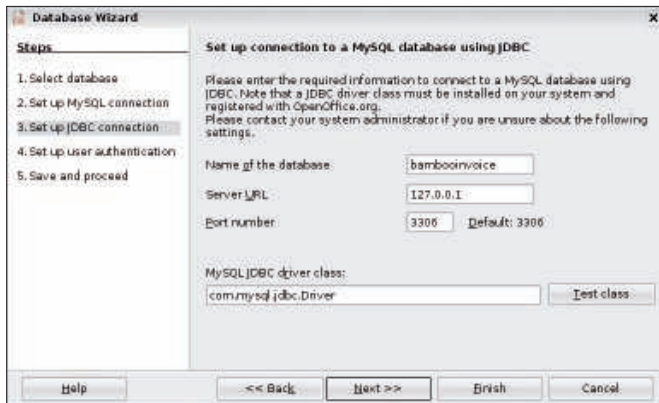
Figure 1: Configuring Connector/J.

Figure 2: Query for the first report.

database username in the *User name* field and tick the *Password required* checkbox. To see whether everything works as it's supposed to, click the *Test Connection* button, then click the *Next* button, select the *Yes, register the database for me* option, and click *Finish*. After you name the database (e.g., "BambooInvoice"), save it, open the database, and click on the *Tables* button. After entering the database's password, you should see a list of all tables in the *bambooinvoice* database.

Creating Queries and Reports

Now I'll explain how to create a simple report that prints a list of all invoices stored in BambooInvoice. Each item in the list will contain the invoice number, customer name, invoice creation date, and the total payment due. All the data is stored in the different BambooInvoice tables, so you must create a query that pulls the desired records from them. Switch to the *Queries* section and click on the *Create Query in Design View* link

to create a new query. In the Query window, add three database tables: *bamboo_invoices*, *bamboo_invoice_items*, and *bamboo_clients*. Next, you have to create relations between these three tables, which is done by dragging the mouse from a field in one table to the target field in another table. This adds a connector between the two fields.

In this sample case, you need to create two relations: *bamboo_invoices.id* to *bamboo_invoice_items.invoice_id* and *bamboo_clients.id* to *bamboo_invoices.client_id* (Figure 2). Then add the *invoice_number*, *dateIssued*, *name*, and *amount* fields to the query by dragging them from the tables onto the *Field* columns in the lower pane of the Query window.

After you've added the fields, you have to group the *invoice_number* field so that all records with the same invoice number appear as one item in the report. To do this, select *Group* in the *Function* row of the *invoice_number* column. Also, you have to set the *Function* option
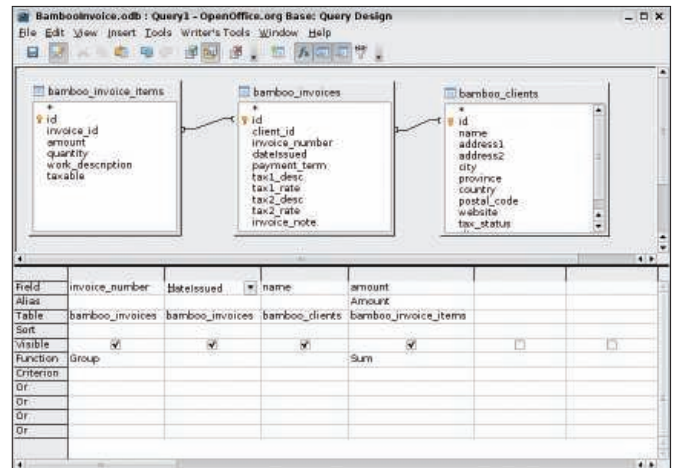
of the *amount* field to *Sum*. This way, the *amount* field will contain the total sum of all invoice payments with the same invoice number.

To better understand what exactly the *Group* and *Sum* functions do, try to run the query without them. Listing 1 shows results similar to those you should receive. As you can see, a single invoice has a separate record for each invoice item. Now apply the *Group* and *Sum* functions as described and run the query again. The result should be a single record, which is exactly what you need (Listing 2).

Next, save the query and give it a name – for example, "Query1." Then you can start working on a new report.

For a new report, switch to the *Reports* section and click on the *Create Report in Design View* link. This opens a blank report in Sun Report Builder. In the *Data* section of the Properties pane, specify a data source by selecting *Query* from the *Content Type* drop-down list and the cre-
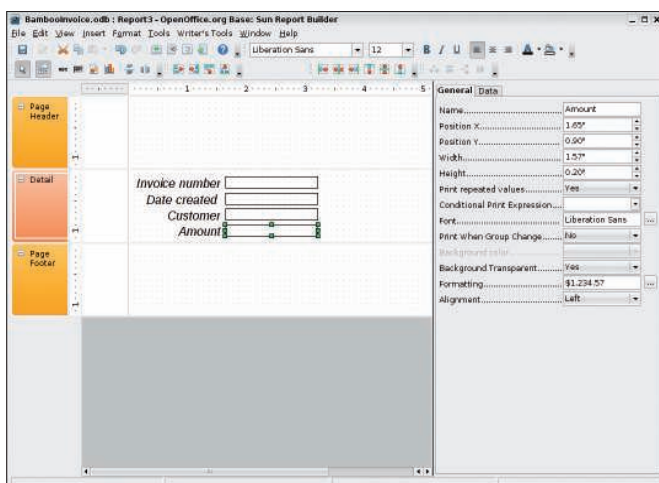

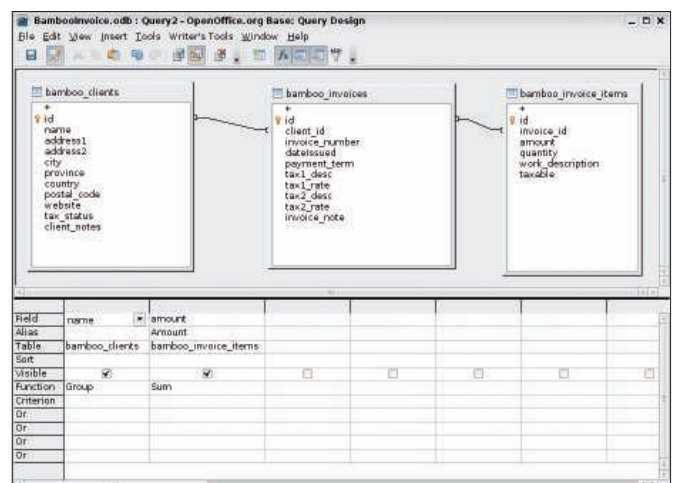**Figure 3: Creating a simple report.**

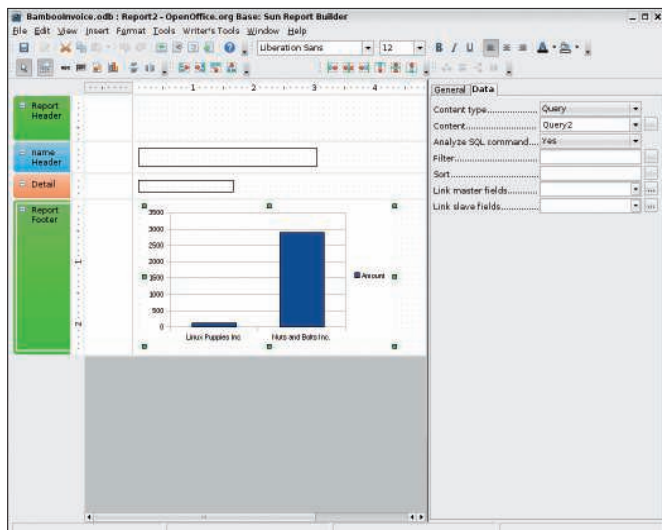**Figure 4: Query for the second report.**
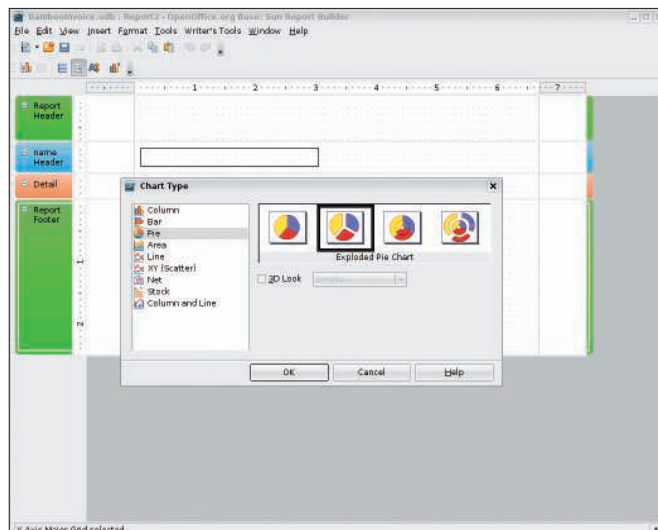
Figure 5: Adding a chart.

Figure 6: Easily replace the default bar chart.

ated *Query1* from the *Content* list. Then drag the fields from the *Add field* window onto the *Detail* section of the report and arrange and format them the way you want (Figure 3). That's it – your report is ready. To test it, press the *Execute Report* button on the Report Builder's main toolbar.

Adding Charts

Because Sun Report Builder supports charts, you can explore your data visually through graphs instead of as plain rows of text and numbers. For example, you can create a report with a chart representing the invoiced amount for each customer. Similar to a regular report, a report with a chart pulls data either directly from a table or from a query. Because you have to obtain records from different tables, you need the latter. In this case, you need to get data from the *name* field of the *bamboo_clients* table and the *amount* field of the *bamboo_invoice_items* table.

As in the previous query, you must set the *Function* option of the *name* and *amount* fields to *Group* and *Sum* (Figure

4). After you save the query, create a new report that uses it as a data source (Figure 5). Because you might want the chart to appear in the report only once, you must place it in either the *Report Header* or *Report Footer* section. To add these sections to the report, choose *Edit | Insert Report Header/Footer*, then press the *Chart* button on the Report Builder's main toolbar and draw a chart in the report section you want. After making sure that the chart is selected (i.e., it has green handles around it), choose the appropriate query as its data source.

Note that instead of creating the query in the *Queries* section, you can specify it as the chart's data source directly in the report. To do this, select *SQL command* from the *Content type* drop-down list, then click on the button next to the *Content* list to open the Query window. Then use it to construct the query. When you save the created query, the system converts it into a SQL statement and inserts it in the *Content* field.

By default, Sun Report Builder inserts a bar chart, but you can change it to another chart type. To do this, click twice

on the chart to put it into editing mode, right-click on the chart, and select the *Chart Type* item. This opens the Chart Type wizard, which helps you choose and configure the chart type you want (Figure 6).

Final Word

Using Sun Report Builder, you can turn OpenOffice.org Base into a powerful BI tool. If you don't feel like creating the described solution from scratch, you can download the sample *BambooInvoice. odb* database [6] containing the described queries and reports.

The database requires a working BambooInvoice installation, and you must replace the default connection settings. To do this, open the *BambooInvoice.odb* database, choose *Edit | Database | Properties*, and replace the default username and password. ■

INFO

[1] Sun Report Builder: *extensions. services.openoffice.org/project/ reportdesign*

[2] Pentaho: *www.pentaho.com*

[3] BambooInvoice: *www. bambooinvoice.org*

[4] "BambooInvoice Web-Based Invoicing Solution" by Dmitri Popov, *Linux Magazine*, December 2008, *http:// www.linux-magazine.com/issues/ 2008/97/smooth_solution*

[5] MySQL connectors: *www.mysql. com/products/connector/*

[6] Sample BambooInvoice database: *www.mediafire.com/?gi44bdx1z4m*

Listing 1: Query without Group and Sum

```
01 invoice_number  dateIssued      name      amount
02 3       11/27/08        Nuts and Bolts Inc    375
03 3       11/27/08        Nuts and Bolts Inc    2157
04 3       11/27/08        Nuts and Bolts Inc    211
```

Listing 2: Query with Group and Sum

```
01 invoice_number  dateIssued      name      amount
02 3       11/27/08        Nuts and Bolts Inc     2743
```