



Consejos y trucos para OpenOffice.org Base

INFORMES

Aprenda a usar herramientas avanzadas de OpenOffice.org Base mediante la creación de una solución de base de datos. Nuestra aplicación de facturación de muestra también ayuda a demostrar las capacidades para la creación de informes de la extensión Sun Report Builder. **POR DMITRI POPOV**

Crear una aplicación de base de datos simple con OpenOffice.org Base no es complicado: Añadir una o dos tablas, diseñar un formulario, integrar consultas, y nuestra solución de base de datos estará lista. Pero con OpenOffice.org Base podemos hacer mucho más que crear bases de datos de tablas para guardar nuestras recetas y libros favoritos.

Para ilustrar las herramientas más avanzadas de Base y demostrar un par de técnicas útiles, mostraré cómo crear una solución de base de datos, para ser concretos, una aplicación de facturación. Aparte de ser una herramienta útil en sí misma, una aplicación de este tipo demuestra las capacidades de creación de informes de Sun Report Builder [1]. Esta extensión supera drásticamente el motor integrado de informes de Base, que es más bien limitado.

Antes de comenzar a trabajar en una aplicación de facturación, debemos descargar e instalar la última versión de la extensión. Recuérdese, sin embargo, que Sun Report Builder sólo funciona con OpenOffice.org 2.3 o superior, y si usamos una versión más antigua de la suite, deberemos actualizarla primero para poder obtener todas las ventajas de la nueva herramienta.

Sun Report Office se distribuye como paquete *.oxt*, así que se instala como una extensión OpenOffice.org convencional. Sencillamente, descargamos el último paquete *sun-report-builder_x.x.x.oxt*, arrancamos OpenOffice.org y elegimos *Herramientas | Administrador de Extensiones*. Luego seleccionamos la sección *Mis Extensiones*, pulsamos el botón *Agregar*, seleccionamos la extensión descargada, pulsamos *Cerrar* y reiniciamos OpenOffice.org.

Sun Report Builder no añade ningún elemento de menú, sino que se integra directamente en OpenOffice.org Base. Para llamar al constructor de informes debemos cambiar a la sección *Informes* en la ventana principal de Base y pulsar el botón *Crear Informe en la vista Diseño*, o abrir un informe ya existente para su edición.

Si deseamos crear una base de datos de facturación necesitamos comenzar desde cero. Afortunadamente, ya he creado una base de datos llamada *minilInvoices* [2], a partir de la cual podremos aprender unas cuantas técnicas útiles.

Dentro de minilInvoices

Como *minilInvoices* está diseñada para escritores independientes, nos proporciona una manera de mantener al día artículos y publicaciones y crear facturas (Figura 1). La

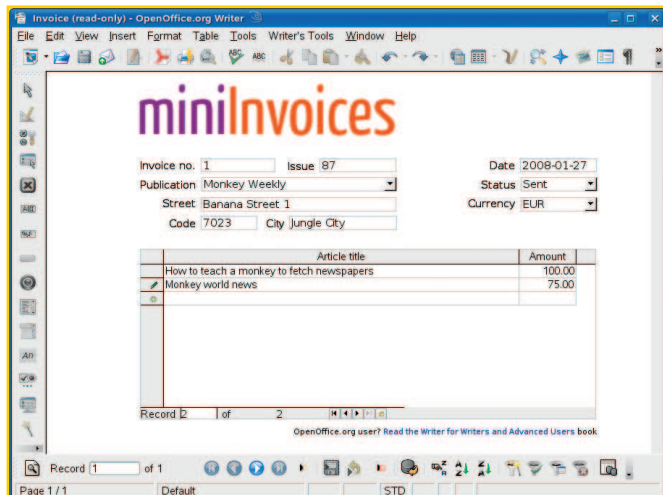


Figura 1: Factura de minilInvoices.

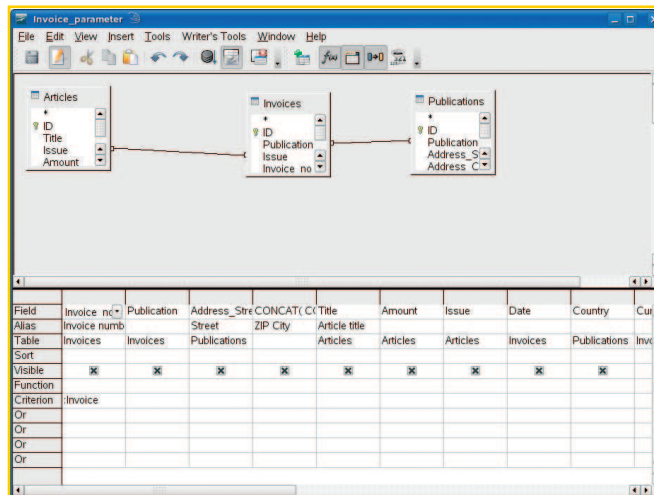


Figura 2: Consulta Invoice_parameter.

base de datos consta de tres tablas: Artículos, Publicaciones y Facturas. Podemos añadir otros campos y tablas para adaptarla a nuestras necesidades particulares fácilmente.

Para recuperar los datos de las tablas, la base de datos usa dos consultas: *Invoice_query* y *Invoice_parameter*. Pulsando sobre el elemento *Invoice_query* en la sección *Consultas* y seleccionando *Edit* se abre la consulta en modo edición. Esta consulta contiene las tres tablas de la base de datos y las conecta a través de las relaciones *Article.Issue-Invoices.Issue* y *Publications.Publication-Invoices.Publication*, de manera que la consulta puede recuperar datos de los campos relacionados en las tablas.

La consulta *Invoice_parameter* (Figura 2) es muy similar a *Invoice_query*, excepto por dos diferencias. En primer lugar, el campo *Invoice_no* está definido como criterio *:Invoice*. Este criterio nos permite limitar la consulta a una factura específica. Cuando la ejecutamos, el criterio nos presenta el cuadro de diálogo *Entrada de Parámetros*. Si deseamos ver los datos correspondientes a la factura 21, por ejemplo, introducimos 21 en el campo *Valor* y la consulta nos da el resultado para el número de factura especificada. ¿Por qué deseáramos ejecutar una consulta para una factura? Porque necesitamos esta consulta para crear un informe que imprima una única factura.

CONCAT

La consulta *Invoice_parameter* contiene también otro truco. Si

echamos una rápida ojeada a los campos en la consulta, veremos uno un poco extraño que comienza con el comando *CONCAT*. Este campo calculado denominado con este mismo nombre usa una consulta SQL en lugar de un campo actual de la base de datos. En este caso, la consulta SQL está basada en el comando *CONCAT* que concatena dos o más cadenas (es decir, las “pega” juntas). Por ejemplo, si deseamos concatenar el código postal del campo *Address_Code* y la ciudad del campo *Address_City* en la tabla *Publications*, podemos utilizar el siguiente comando:

```
CONCAT("Publications".
"Address_Code",
"Publications"."Address_City")
```

Sin embargo, descubriremos un ligero problema con este enunciado SQL: no inserta un espacio en blanco entre los dos campos, por lo que el resultado será “81292-Munich”.

Para solucionarlo deberemos crear un enunciado *CONCAT* anidado como

```
CONCAT( CONCAT(
"Publications".
"Address_Code", ' ' ),
"Publications"."Address_City" )
```

La parte *CONCAT("Publications"."Address_Code", ' ')* crea una cadena que acaba con un espacio en blanco, y el segundo comando *CONCAT* pega las dos cadenas juntas. Cuando necesitamos visualmente un formato de informe consistente, este truco resulta muy útil. Por ejemplo, el código postal y la ciudad parecen habitualmente en la misma línea y creando dos cuadros de texto separados para cada campo llevaría al diseño siguiente:

```
81292 Munich
DK-8000 Arhus
```

¿Se ve el problema?

Usando concatenación nos aseguramos de que siempre haya un solo espacio en blanco entre los dos campos, sin importar la longitud de la cadena del código postal.

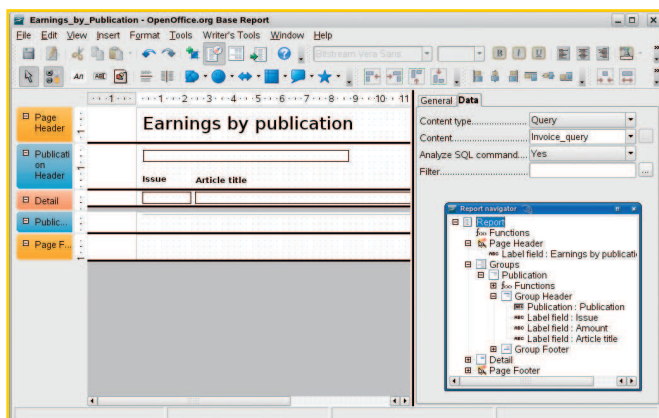


Figura 3: Diseñando un informe en Base.

Formularios e Informes

La próxima parada es la sección *Forms*. El formulario *Invoice* nos permite crear facturas y añadirles artículos. Su diseño no es particularmente complicado: consta de un formulario principal (*Invoices*) y de dos subformularios (*Articles* y *Publications*).

Explorando la estructura del formulario usando *Form Navigator*, podemos entender el resto fácilmente (Figura 3).

Cuando añadimos subformularios, no debemos olvidar especificar su relación con el formulario principal especificando los campos *Enlazar maestro* y *Enlazar esclavo* en la ventana de diálogo *Propiedades del Formulario* bajo la solapa *Datos*. Por ejemplo, el subformulario *Article* está unido al formulario *Invoices* mediante los campos *Issue*.

Cualquier base de datos de facturas es bastante inútil si no podemos imprimirlas. Gracias a Sun Report Builder, podemos diseñar informes para imprimir facturas y también para ver sus datos de diferentes maneras. *miniInvoices* presenta dos informes: el primero, *Earnings_by_Publication*, genera un informe de ganancias por publicación, y el segundo, *Single_Invoice*, nos permite imprimir una factura.

Informes, Grupos y Detalles

Los informes de Base constan de tres componentes principales: Informe, Grupos y Detalle. De manera similar a los formularios, los informes extraen los datos de una fuente de datos especificada. Por ejemplo, el informe *Earnings_by_Publication* se los trae de la consulta *Invoice_query*. El componente Grupos nos permite agrupar los datos por un campo específico. En este caso, el informe se agrupa por el campo *Publication*.

Finalmente, Detalle se usa para generar una perspectiva por elementos de los datos. Por ejemplo, el Detalle del informe genera

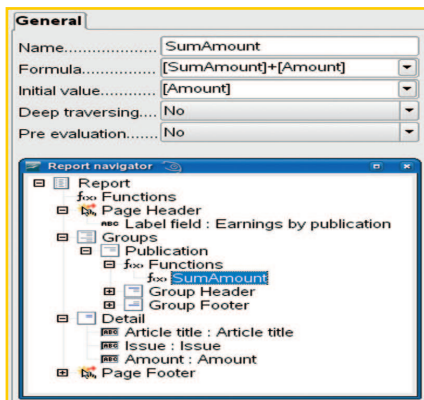


Figura 4: Definiendo una función con Sun Report Builder.

una lista de artículos escrita por una publicación específica simplemente colocando el título *Article* y otros campos relacionados de *Invoice_query* en la parte Detalle.

Además, podemos usar Sun Report Builder para añadir funciones a nuestro informe. El informe *Earnings_by_Publication*, por ejemplo, contiene una función simple que calcula las ganancias totales para cada publicación (Figura 4). Más información acerca de las funciones se encuentra disponible en la wiki de Sun Report Builder [3].

A diferencia de *Earnings_by_Publication*, el informe *Single_Invoice* está basado en la consulta *Invoices_parameter*. Esta consulta sólo devuelve datos de un número de factura específica, que es lo que necesitamos cuando imprimimos una única factura.

El informe *Single_Invoice* usa otra práctica técnica. Por ejemplo, si vivimos en España y enviamos una factura a una publicación española, la dirección de la factura no precisa incluir el país. Para evitar que la *City* aparezca en la factura, podemos especificar una condición para el cuadro de texto de la ciudad en el campo *Impresión de Expresiones Condicionada* en la ventana *Propiedades*.

En este caso, la condición debería ser la siguiente: `[Country] < > "Spain"`. Como habrá podido adivinarse, esta expresión imprime la ciudad solamente si su contenido no es "Spain".

Integración de miniInvoices

Echaremos una ojeada rápida para ver cómo podemos hacer que *miniInvoices* funcione con *Writer* de forma que podamos añadir rápidamente el documento editado actualmente a la tabla *Articles*. Simplemente registramos *miniInvoices* con *OpenOffice.org* como fuente de datos y añadimos la macro que se muestra en el Listado 1.

El elemento clave en la macro es la consulta SQL que escribe los datos en la tabla *Article*. Obsérvese que la macro acerca la información sobre el título y el tema del artículo de las propiedades del documento especificadas bajo *Archivo | Propiedades*. El título del artículo se ha recuperado del campo *Title* en la sección *Description*, y los valores del número de la revista y el importe se han tomado de los campos *Info 1* y *Info 2* de la solapa *Definida por el usuario*.

Una Última Cosa

Aunque la aplicación *miniInvoices* sólo usa unos cuantos trucos y técnicas, deberíamos encontrarlos útiles cuando compilamos nuestras propias soluciones *OpenOffice.org Base*.

Aparte de ver cómo funciona *miniInvoice*, tómate la libertad de hacerle mejoras o simplemente de usarlo tal y como está. ■

Listado 1: Añadir Article a la Macro Invoices

```
01 Sub AddArticleToInvoices()
02 DBContext=createUnoService("com.sun.star.sdb.DatabaseContext")
03 If not DBContext.hasByName("miniInvoices") then
04     MsgBox ("La conexión a la base de datos falló.", , "¡Conexión
fallida!") : End
05 End If
06 DataSource=DBContext.getByname("miniInvoices")
07 ConnectToDB=DataSource.GetConnection (" ", " ")
08 SQL=Query="INSERT INTO " & "Articles" & " " + (" " & "Title" & " " & "Issue"
& " " & "Amount" & " ") VALUES " & " (" & " " + ThisComponent.DocumentInfo.Title
& " " & " " + ThisComponent.DocumentInfo.getUserFieldValue(0) & " " & " "
& " " + ThisComponent.
09 DocumentInfo.getUserFieldValue(1) & " " & " " & " "
10 SQLStatement=ConnectToDB.createStatement
11 Result=SQLStatement.executeQuery (SQLQuery)
12 ConnectToDB.close
13 ConnectToDB.dispose()
14 MsgBox ("Los datos han sido guardados.", , "¡Terminado!")
15 End Sub
```

RECURSOS

- [1] Sun Report Builder: <http://extensions.services.openoffice.org/project/reportdesign>
- [2] *miniInvoices*: <http://code.google.com/p/writertools/downloads/list>
- [3] *miniInvoices*: http://wiki.services.openoffice.org/wiki/SUN_Report_Builder