



Kexi and Glom – lean database tools compared

Semi-Flyweights

Kexi and Glom aim to create databases with little overhead. We compare the power of these two tools. *By Frank Hofmann*

Users usually don't have direct access to the functionality of a mature relational database because managing the database typically requires root privileges. But, if you need something more than a simple table, Glom [1] and Kexi [2] are two programs that give you the ability to create a database, form author queries, and generate reports.

Both applications are very similar in terms of their field of use. In addition to handling the preparatory work, they create the database schemas and selection masks. These two programs allow users without any database skills to

AUTHOR

Frank Hofmann studied computer science at the Technical University of Chemnitz, Germany. He currently works at Büro 2.0, Berlin, an office network of open source experts (<http://www.buero20.org>), where he provides services in the print and pre-press field (<http://www.efho.de>). Frank is a member of the Potsdam Linux User Group (upLUG).

achieve initial results with very little overhead.

Kexi is an spinoff of Calligra, the office suite by the KDE project, whereas Glom comes from the Gnome camp. Neither of these programs is an unknown quantity, and both are regarded as pretty stable. Kexi, for example, is available directly as a package from the Debian, Ubuntu, Red Hat/Fedora, and Mandriva/Mageia repositories.

Thus far, Glom has been available from the project's website as a download for Ubuntu, Fedora, and Mandriva. Both programs can be set up easily with the respective package managers. If you are looking for a similar free project, the obvious candidates are Open/LibreOffice Base [3][4], H2 [5], and DBDesigner 4 [6]. On the commercial market, Glom and Kexi's competitors are Microsoft Access, FileMaker Pro for Mac OS X or the iPad [7], and Oracle Forms [8].

Connectors

Kexi offers connectors for MySQL, PostgreSQL, and SQLite databases. If you opt

for the default installation, it will store its data in an SQLite database and store the database as a file in the user's home directory. This means that only the user can access the database.

In contrast to this, Glom stores its data in a PostgreSQL database, which it either deposits in the user's home directory or provides centrally on the system. The program only uses a subset of the PostgreSQL functions and can therefore only modify tables that it has created itself. Therefore, if several users run Glom, there is no danger of any conflict

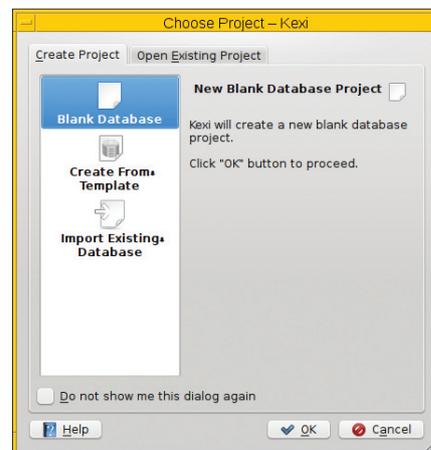


Figure 1: When you launch Kexi, a wizard pops up to help you create a new database or edit a previously saved project.

in terms of the PostgreSQL database used.

Kexi

Kexi adopts the look of other Calligra applications and initially launches a wizard (Figure 1). In the dialog, you can use the tabs to select whether you want to create a new project or continue working on an existing one.

When you start the program, you can choose an empty document, access data via a template, or import data from an existing database. To edit an existing project, you can take the project data either from a database server (MySQL, PostgreSQL, xBase) or from a local file. Figure 2 shows this selection with the database formats supported by Kexi in local files.

After this step, a window pops up with a three-panel view. The menu and navigation bar are located at the top; this is where you open and close projects, create database schemas in the form of tables and SQL queries, design dialog windows and reports, and import and export data and database content.

The project navigator on the left-hand side provides a tree view with an overview of all of the tables, SQL queries, dialogs, reports, and scripts for the current project. The central area in the window contains the details of the selected table, SQL query, or dialog. The appearance of this part of the user interface depends on the object on which you are currently working.

If you are defining the individual fields in a table (under the *Design* button), this part of the window will show you an editable row of input fields. The Property

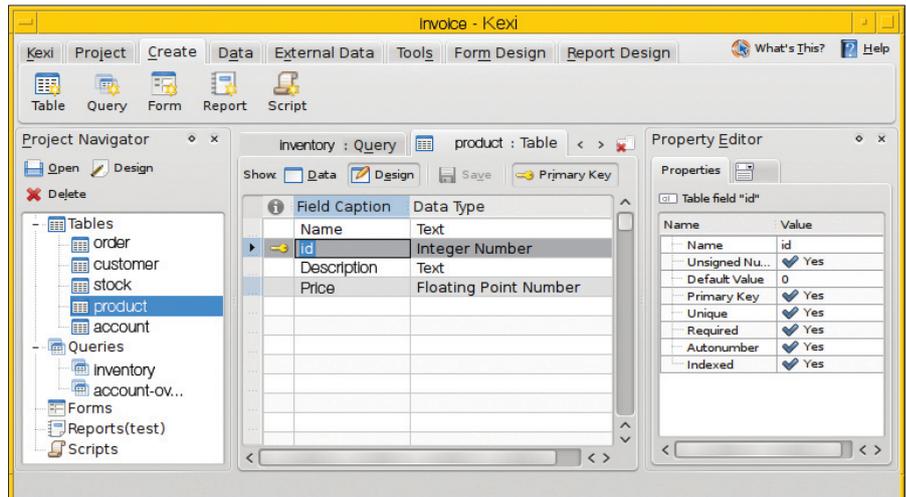


Figure 3: For each field in the table, Kexi offers a dialog with detailed properties.

Editor pane appears on the right-hand side, where you can use it to specify the individual fields (Figure 3).

The *Data* button takes you to view in which you can populate the table with data. Depending on the number of fields in the table, you will see a number of editable input fields organized into rows (Figure 4).

When you are editing an SQL query (*SQL* button), Kexi shows you an import field with multiple lines, where the key words are highlighted to help you keep track of complex queries. If needed, you can press the *Check Query* button to see whether the query is valid. If so, you will see a check mark in the input field and a note to the effect that *The query is correct*.

To check whether you have connected to the correct tables, you can go back to the *Design* button and toggle to visual mode (Figure 5). Kexi lets you launch

the SQL query in the *Data* tab and shows you the results, which are in the typical database style, with a slightly more attractive look.

Kexi's feature scope includes functions for creating tables (fields and data), creating SQL queries, and evaluating the results by, say, creating input fields or reports. Unfortunately, the developers seem reticent about documenting the latter two functions, which makes their use and integration less than intuitive. In daily use, I noticed a couple of things to be aware of when working with Kexi. For example, if you retrospectively modify the schema of an existing table, the application will rewrite part of the SQLite database. Unfortunately, the data in a modified table will be lost.

Kexi initially defines text fields with a length of 200 characters; however, you can modify this length in the Properties dialog. Another thing to watch out for is

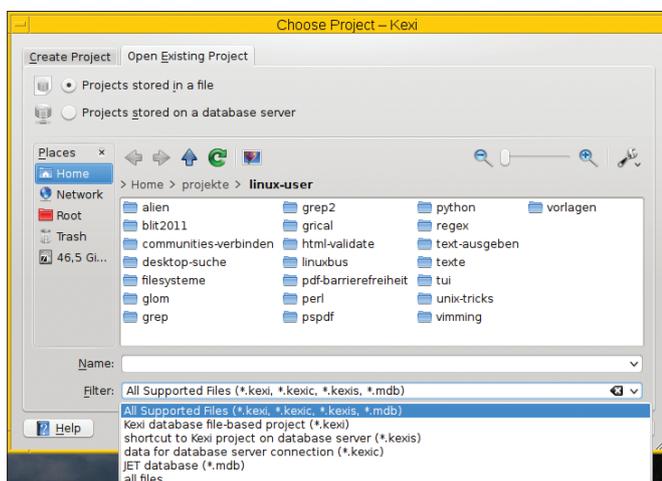


Figure 2: Kexi supports a number of database formats in local files.

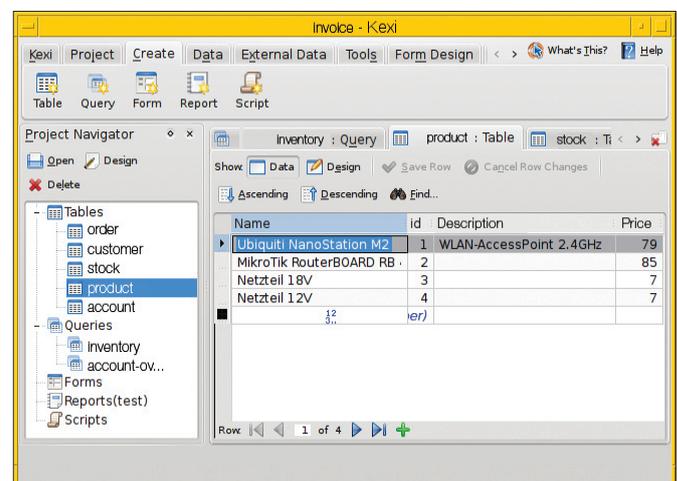


Figure 4: A form lets the user populate the selected table.

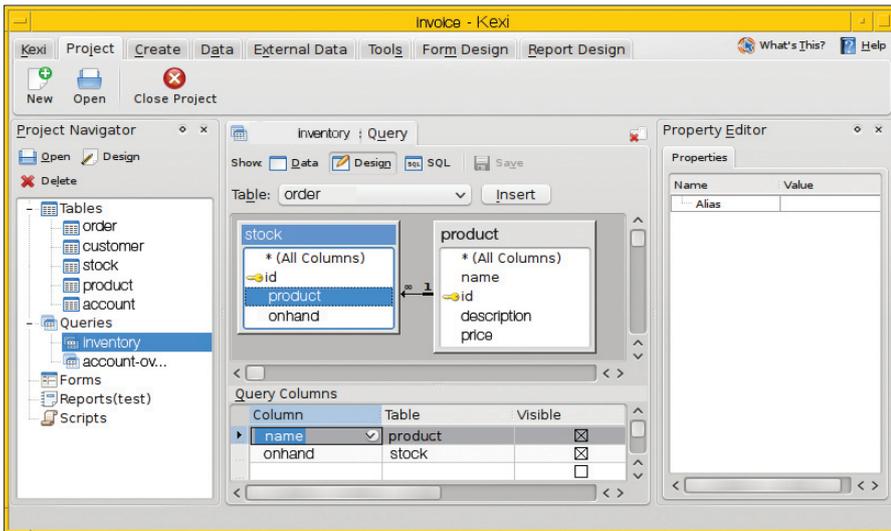


Figure 5: If needed, you can click to check the connections between the tables.

that the program does not save all the content (tables, data, SQL queries) until you close a project, and you don't have a button to save data on the fly.

Kexi stores local project data in the SQLite database format. This opens up an option for running the SQLite3 command-line tool to investigate the Kexi files and make any necessary corrections. This feature is useful for troubleshooting, and it allows you to peek at the inner workings of the database whenever you need to do so.

Overall, Kexi leaves you with a fairly middling impression. From a production point of view, it is suitable for small projects and small volumes of data. Some parts of Kexi seem to be highly unstable – this is particularly true of the series 1.0 versions. Many positive changes have been made as of version 2.0. In the lab, I used version 2.3.3 running on Ubuntu 11.10.

Glom

When you launch Glom, a wizard pops up to accompany you through the steps

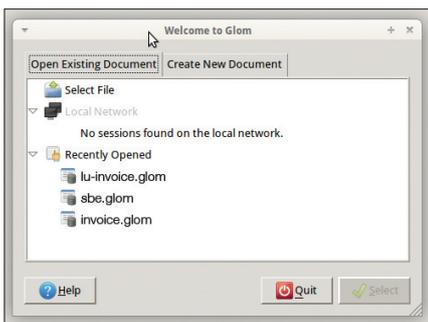


Figure 6: The Glom wizard helps you create a new project.

that follow (Figure 6). It offers you the choice of opening an existing document or creating a new project. If you choose the first option, you can choose to access the document on the network or load a local file.

To create a new project, you either create an empty document or use one of the templates that come with the program. The templates include some fairly complex examples with tables, connections, and reports that might help you learn what information to enter where in Glom. After this step, you define the name and the file name for the project. Additionally, you need to define the title and database type – either local or on an external database server (Figure 7).

The next step is to define the database schema. To do so, you first must define the individual tables. Pressing the *Add* button in the dialog takes you to a new

line with input fields and adds a new table (Figure 8). *Delete* removes the table that you selected in the list. The *Open* button takes you to the main Glom window with the selected table, where you can enter the details (fields and field formats).

The main Glom window offers a menu bar from which you can access most actions directly. The buttons that are available in the bar depend on the current application mode. In *Developer* mode, you can modify the database schema, whereas in *Operator* mode, you can only enter data – the menu items used to modify the schema are grayed out.

To switch the mode in which you are working, you can select the menu item *Developer | Operator mode* or *Developer | Developer mode*. Below the menu, the application shows you all the information for the currently selected table.

To define the fields and field formats for a table, first select the desired table in *Tables | Table name*. Then, go to the *Developer | Fields* menu to open a separate window where you can specify the field definitions (Figure 9). Pressing *Delete* removes the selected field from the list.

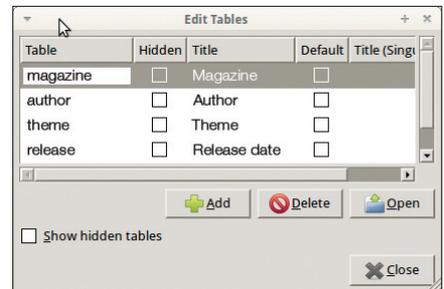


Figure 8: Glom provides an easy approach to creating tables.

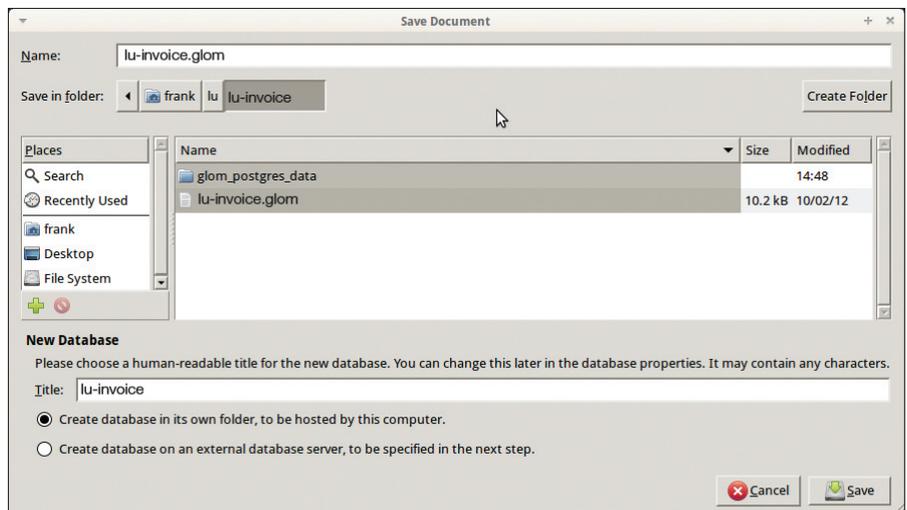


Figure 7: With just a few details, you can create a new project.

Too Swamped to Surf?

Our ADMIN Online website offers additional news and technical articles you won't see in our print magazines.

Subscribe today to our free ADMIN Update newsletter and receive:

- Helpful updates on our best online features
- Timely discounts and special bonuses available only to newsletter readers
- Deep knowledge of the new IT

ADMIN
Network & Security



www.admin-magazine.com/newsletter

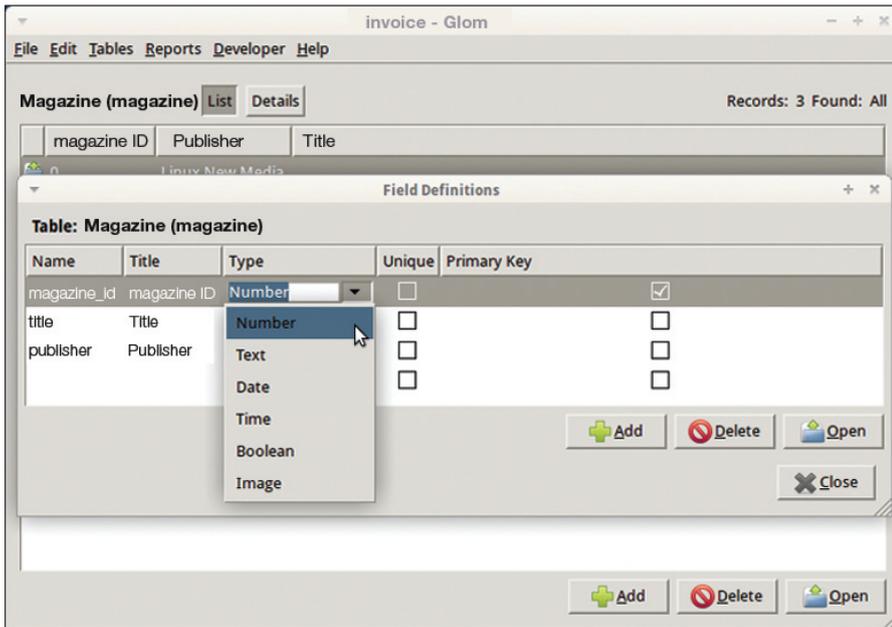


Figure 9: Defining field types in Glom involves just a few mouse clicks.

Pressing the *Add* button lets you supplement the details for a field. The *Name* column is for the field's object name in the database schema. *Title* gives you the field name as it appears as a column header in the table overview. The field *Type* defaults to *Number*.

Clicking on the *Type* field opens a drop-down menu where you can modify the value: You can choose between the *Text*, *Date*, *Time*, or *Image* field types. Pressing the *Open* button takes you to another dialog box that lets you specify a field in more detail – for example, by defining a default value or the numeric format.

After defining all of the fields, you can go to the *Developer | Layout* menu item to specify the order and view of the fields in the table. You can use the buttons to change the order or position in the view. Pressing the *Delete* button removes a field.

To change the way in which a field is displayed (e.g., text color, font), you need to press the *Formatting* button. The *Edit* button takes you to the window described earlier, where you can edit the details of the field.

The *Add Field* button opens a dia-

log with a selection list of the previously defined fields (Figure 10), and pressing *Select* adds the field that is selected in the list.

After defining all the tables and populating them with values, you can move on to more complex actions. Glom gives you an *Edit | Search* menu item which you can use to apply a search (you can also use the keyboard shortcut Ctrl + F). The search is launched against the entire table, but the results will only show you the lines that contain matches.

Pressing *Search all*, which is located next to the number of matches, lets you remove the filter. Clicking on a column header will sort the table on the selected column; clicking the header again toggles from ascending to descending sort order.

Glom does not give you an option for running SQL commands against tables

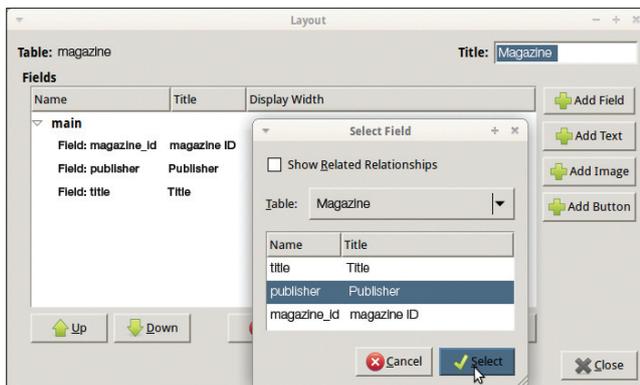


Figure 10: You can arrange the previously defined fields in the table to suit your needs.

(e.g., to output tables or connect them). The Python interface, which you can access via *Developer | Script library*, is very sparsely documented. The same thing applies to relations between tables (*Developer | Relations for this table*) and the *Developer | Reports* function. Just as in Kexi, the program lacks the buttons for saving data on the fly – the software doesn't save the content until you close the project.

In the lab, I used Glom version 1.20.7 on Xubuntu 11.10. In daily use, you will probably appreciate the mostly organized and understandable manual that the developers integrated into the Glom interface and that appears in a separate window.

Right now, Glom is an excellent choice for smaller projects – for example, an address book or a task manager. The ability to connect to tables would make Glom suitable for larger projects, such as a combined customer and invoice management system.

Conclusions

Kexi and Glom are both promising tools that facilitate the use of a database for newcomers to the topic. Both programs have some rough edges, but they are already usable for smaller projects. Although these tools are unable to compete with commercial applications such as FileMaker Pro or Oracle Forms at this time, there is no reason for that to remain true. ■■■

ACKNOWLEDGMENT

I would like to say thank you to Wolfram Eifler for his critical appraisal and comments on this article.

INFO

- [1] Kexi: <http://kexi-project.org>
- [2] Glom: <http://www.glom.org>
- [3] OpenOffice.org Base: <http://www.openoffice.org/product/base.html>
- [4] LibreOffice Base: <http://www.libreoffice.org/features/base/>
- [5] H2: <http://www.h2database.com>
- [6] DBDesigner 4: <http://www.fabforce.net/dbdesigner4/>
- [7] FileMaker: <http://www.filemaker.com/>
- [8] Oracle forms: <http://www.oracle.com/technetwork/developer-tools/forms/overview/>