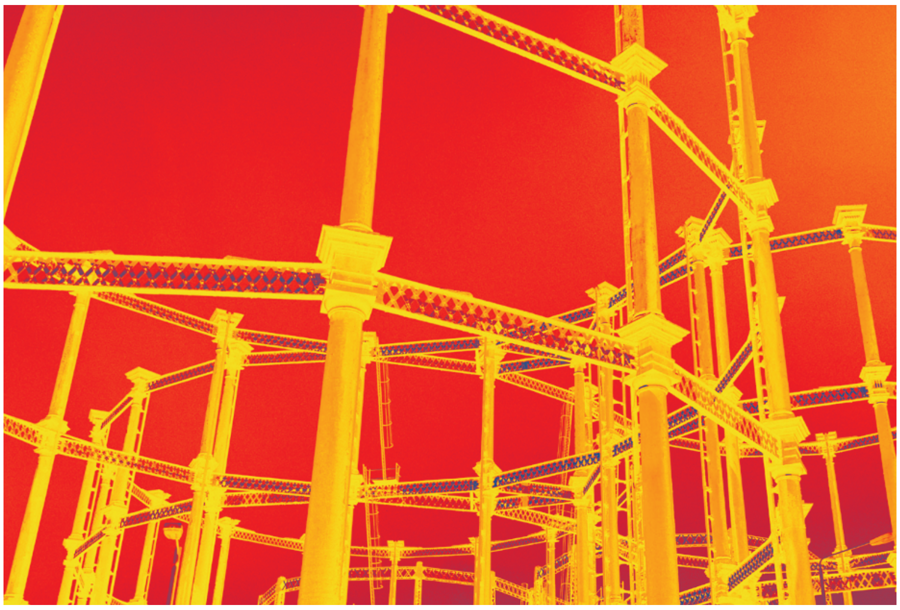
Data Management in KOffice

Modular Architecture

The Kexi project with its KDE database program not only provides useful basic functions for end users, but also interesting technologies for database developers. The complete rewrite of Kexi has allowed the developers to add new ideas to the existing architecture.

BY LUCIJAN BUSCH



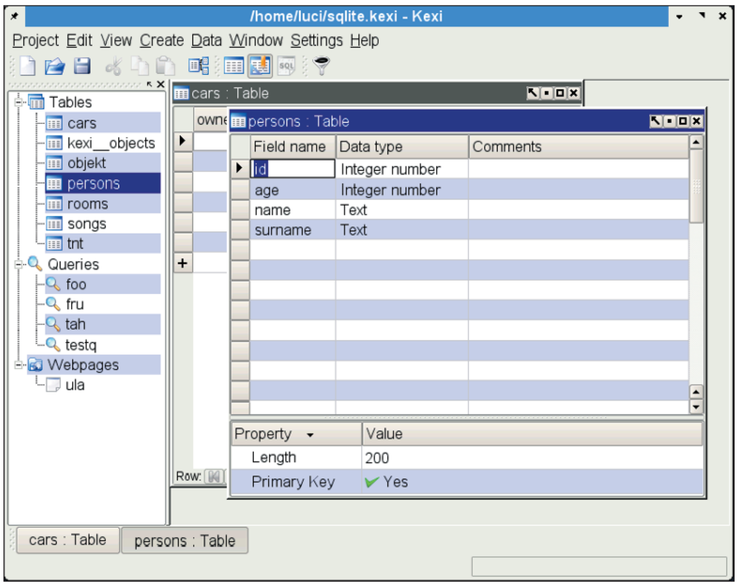
No office suite is complete without a database front-end. This is Kexi, in KOffice's case. In contrast to comparable programs, Kexi can be used to create small databases without needing to first set up a server. Kexi stores the data in a single file in this case. Of course, if you need to handle larger amounts of data, or provide access to multiple users, it does make sense to use a server.

Starting Over is Hard to Do

Although the beta1 version of Kexi had a good range of features, the developers decided to go in for a complete rework of the architecture. This explains why the front-end provides only basic database management facilities at time of writing. Users can create tables and edit the data in them, but Kexi does not currently have a front-end that would allow users without SQL skills to define database queries. Users that speak SQL can use drag and drop to create a query, as Figure 1 shows.

To perform a query across multiple tables, users first need to define the relationships between the tables in the database schema editor. Again, they can use drag and drop to do so. More experienced users who prefer direct SQL entries can use an editor with syntax highlighting and a history function. Self-programmed SQL queries can be reviewed and modified in draft mode. Both in draft and in SQL direct entry mode, Kexi will store only the SQL input. The program will store incorrect statements allowing you to correct them later.

Kexi includes an integrated database engine called KexiSQL that can be used to create simple projects. Kexi will store your project data in a file, allowing users to exchange data by email for example, or publish the data on a website. KexiSQL is derived from the popular SQLite engine. Although Kexi has a native database format, it can still access other DBM systems such as MySQL and PostgreSQL. Kexi is independent of the underlying DBM system both from the user's and the developer's perspective. An ODBC driver is in the pipeline.



Field name	Data type	Comments
id	Integer number	
age	Integer number	
name	Text	
surname	Text	

Property	Value
Length	200
Primary Key	✓ Yes

Figure 1: Convenient database manipulation with Kexi. Although the program is at an early beta stage, it can handle basic operations such as creating tables.

Back to the Future

The beta1 version of Kexi, which the developers released in April 2003, had a quite a few additional modules, such as forms, reports, and macros. However, to achieve a more flexible program architecture, the development team decided first to rework the underlying libraries, before going on to add the components that build up on these libraries: the advantage of this approach to the user being a mature and stable front-end. At the moment, the developers are focusing on stabilizing the API.

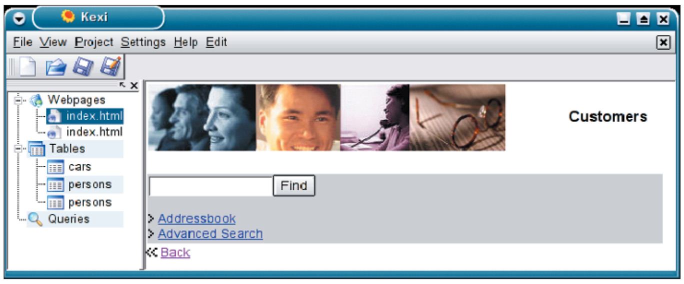


Figure 2: In future, Kexi will allow users to create HTML forms. Enterprises will be able to model database forms on their websites.

The idea is to provide Qt Designer compatible forms, and use the KParts technology to embed BLOB (Binary Large Object) field content. This allows users to create forms in which Kexi can display PDF files or play MP3s. The developers plan to use the Kugar [3] generator for reports. Kugar is being reworked for KOffice 2.0 and should have an improved user interface and provide zoom support. Instead of the QSA (Qt Script for Applications [4]) environment, in future Kexi will probably use KJSEmbed, an ECMAScript-based macro environment.

Also, Kexi looks set to have a facility unknown in comparable environments: the ability to embed HTML pages. This module would allow organizations to model the user interface on their homepage design. Kexi would use Javascript and special URLs to add data and functionality to the template (see Figure 2).

Kexi should be back in synch with the KOffice release series in time for KOffice 2.0. Currently, the developers are looking to allow KWord to access Kexi resources for mail merge operations, and to allow the KSpread spreadsheet directly to import Kexi data. It will also be possible to create KSpread diagrams that directly reference database information.

Despite the impressive list of features, the developers have even grander plans for Kexi. They envisage a Kexi version with complete RAD (Rapid Application Development) facilities some time in the future. This version will allow developers to generate shared libraries, and add ready-made Kexi projects – such as tables, forms, or reports – to their own applications. Additionally, Kexi will be able to generate executables that run on systems without a Kexi environment. The Kexi team also plans to write a Quanta plug-in. This combination of a database program with a Web development tool would allow programmers to create dynamically generated Web pages, simply by allow the plug-in to generate PHP code.

Open Office Polska is actively involved in Kexi development, the aim being to provide Open Office integration and offer the results to customers on today's major platforms. Kexi currently runs on a variety of Unix architectures and Microsoft Windows.

Core and Split-Offs

For the most part, Kexi is programmed in C++, based on KDE and Qt, and highly modular. The executable itself has a tiny 13 KByte footprint, but it does need a number of libraries and plug-ins. There are three main libraries: *KexiCore*, *KexiWidgets*, and *KexiDB*. KexiCore provides a plug-in framework and manages the main window with the module view and plug-in windows. The main window uses the new KMDI (Multi Document Interface) provided by KDE 3.2. A compatibility library is available for older KDE versions. The KDE developers created KMDI with integrated development environments in mind: KDevelop also uses this interface.

Thanks to KMDI, a number of views are available for database projects: a tabbed view (tab-page and ideal modes), with sub-windows in a main window (child-frame mode), or with independent single windows (top level mode). KexiCore is also an interface that allows users to embed Kexi in other applications, such as KWord or KSpread.

KexiWidgets handles shared widgets such as tables and SQL query editors. Each module is a plug-in. For example, there is a table plug-in, another for queries, and yet another for relations.

As Kexi loads plug-ins dynamically, the program launches extremely quickly. And thanks to the plug-in facilities, it is quite easy to integrate Kexi with your own applications. Developers simply need to understand the interface, but do not need to dig through the whole codebase.

As Kexi was released under the LGPL, enterprises can develop and sell proprietary plug-ins. This opens up opportunities for tying Kexi in to commercial packages.

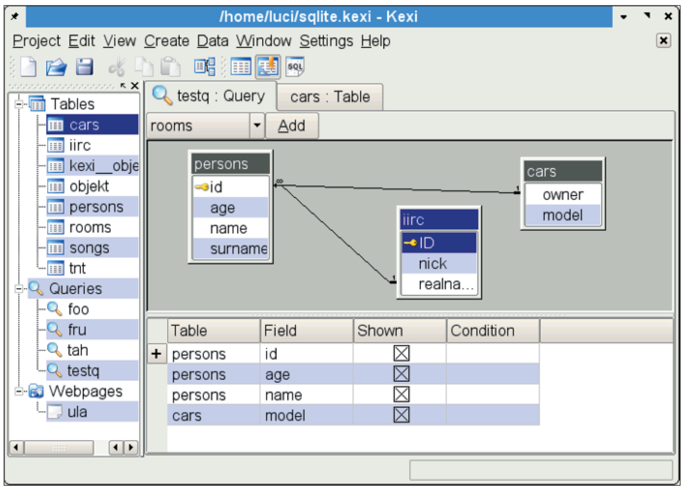


Table	Field	Shown	Condition
persons	id	☒	
persons	age	☒	
persons	name	☒	
cars	model	☒	

Figure 3: Users can point and click to create queries with Kexi. Kexi also displays the relations between individual tables.

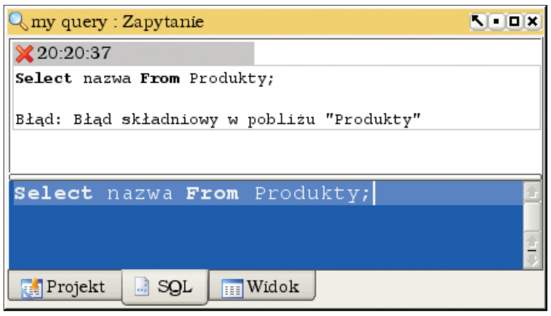
Flexible and Polyglot

KexiDB is a database abstraction layer that seems to work just like QtSQL and ODBC at first glance. However, the developers were not happy with either, and decided to create their own library instead. KexiDB has drivers for various database systems. In contrast to many other libraries of this kind, KexiDB has a very flexible approach to variations on database design. This allows developers to ignore the design of the database and simply build on the KexiDB abstractions. Despite the high degree of abstraction, the operations are fast and do not need a complex API.

KexiDB provides a useful set of meta-information for queries or tables. The querying application can access this information by means of schema classes. Schemas are editable, allowing users to add new fields to tables or define queries. After completing the extended schema definition, drivers translate the meta-information into SQL statements. These will be select statements for queries, and create or alter statements in the case of tables and indices.

The KexiDB library ensures that programmers can do without hard coded queries to a greater extent, at the same time avoiding issues with SQL dialects or misinterpretations. KexiDB also has a parser that converts stored SQL queries back into metadata. Kexi then uses these data to generate the query view as shown in Figure 3.

One of the Kexi parser's special features its ability to display error messages, for example in case of invalid input, in the local language. Figure 4 shows an example in Polish. Additionally, the parser marks the point where the error occurred in the user interface. This could be the position of an invalid entry in the input window for example.



```
20:20:37
Select nazwa From Produkty;

Błąd: Błąd składniowy w pobliżu "Produkty"

Select nazwa From Produkty;
```

Figure 4: Thanks to the flexible implementation of the KexiDB SQL parser, it is quite easy to output multilingual messages. This shows an example in Polish.

The parser is quite error-tolerant and speaks a subset of SQL, allowing it to understand the SQL dialects used by the most popular engines. KexiDB also supports an extended transaction system, allowing for named and nested transactions, assuming the database supports them. The developer plans to support asynchronous database operations in a future release.

Programmers can use KexiDB independently of Kexi itself. Some time in the future, there may even be a special developer version that will run without KOffice, and only require Qt.

Future

The basic Kexi API still needs to be stabilized. As soon as this has happened, developers can look forward to engine-independent database access, a tolerant parser, and a transaction system that is easy to handle. This said, Kexi will probably need some time before it can provide a powerful and versatile GUI to users. The development team plans to release a test version that should be of interest to non-developers in the third quarter of 2004. The next stable version will accompany KOffice 2.0. ■

INFO

[1] Kexi project: *http://www.kexi-project.org/*

[2] Quanta: *http://quanta.sourceforge.net/*

[3] Kugar report generator: *http://www.koffice.org/kugar/*

[4] Qt Script for Applications: *http://www.trolltech.com/products/qsa/*

[5] Open Office Polska: *http://www.openoffice.com.pl/*

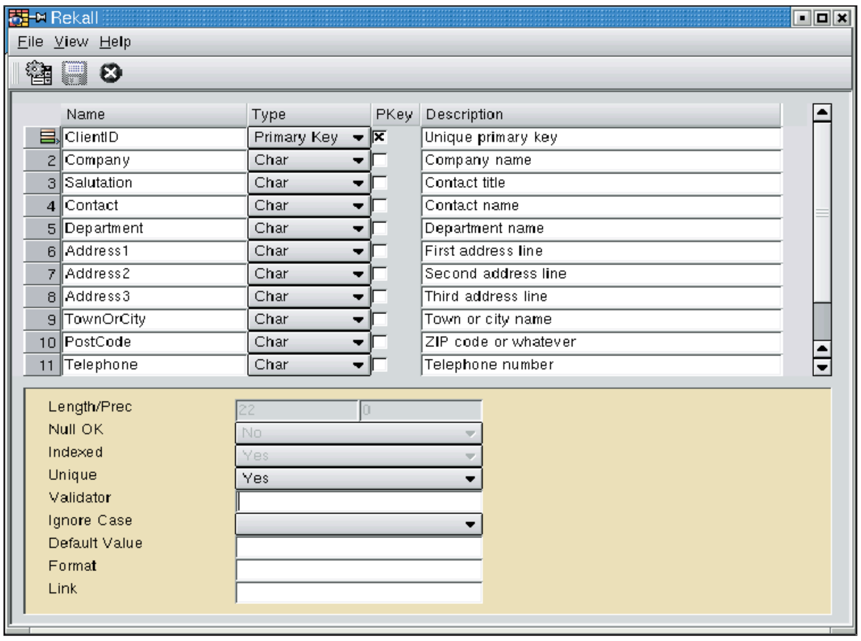
[6] The Kompany: *http://www.thekompany.com/*

Rekall

Of course, KDE has a few other database front-ends besides Kexi. Rekall, which was originally developed as a commercial product by TheKompany.com [6], is one of them. Rekall has been available under the GPL since late last year. The program uses the Qt libraries, and thus runs on Linux, MacOS X, and Windows.

Rekall has facilities for creating tables, data entry, creating forms, and even scripting database applications which can be run as executables. This range of features qualifies Rekall as a RAD application, and allows developers to create applications that run independently of Rekall.

One of the major differences between Kexi and Rekall is the fact that Kexi provides direct KDE integration, whereas Rekall runs independently of KDE. Also, Kexi uses ECMAScript for scripting, whereas Rekall uses Python. The user interfaces are also quite different. There is a version of Rekall for the Zaurus PDA, which was developed by TheKompany.com.



	Name	Type	PKey	Description
	ClientID	Primary Key	✓	Unique primary key
2	Company	Char		Company name
3	Salutation	Char		Contact title
4	Contact	Char		Contact name
5	Department	Char		Department name
6	Address1	Char		First address line
7	Address2	Char		Second address line
8	Address3	Char		Third address line
9	TownOrCity	Char		Town or city name
10	PostCode	Char		ZIP code or whatever
11	Telephone	Char		Telephone number

Length/Prec	22	0
Null OK	No	
Indexed	Yes	
Unique	Yes	
Validator		
Ignore Case		
Default Value		
Format		
Link		

Figure 5: The Rekall database front-end is an alternative to Kexi. It became available under the GPL late last year. Rekall has a wide range of functions. Programmers can use it as a Rapid Application Development environment.