

Modulare Architektur

Kexi ist die Datenbankanwendung für KOffice. Anwender erstellen mit ihr Abfragen, Formulare und Berichte. In seiner jungen Geschichte hat es bereits einen kompletten Rewrite erfahren, von dem vor allem Entwickler profitieren. Das Kexi-Team hat eine neue Architektur erarbeitet und sie sauber implementiert. Lucijan Busch



www.photocase.de

Zu einer kompletten Office-Suite gehört ein Datenbank-Frontend. Bei KOffice ist das Kexi. Es lässt sich mit ausgewachsenen Datenbankservern betreiben, arbeitet aber auch in Standalone-Umgebungen mit einem eigenen Dateiformat. Vor allem kleine Heimanwendungen profitieren von diesem Format. Bei großen Datenmengen oder vielen Benutzern, die gleichzeitig auf die Bestände zugreifen, empfiehlt es sich jedoch, einen Server zu verwenden.

Aller Anfang ist schwer

Obwohl die Beta-1-Version vom April 2003 bereits viele Features enthielt, haben sich die Entwickler dazu entschlossen, das Programm von Grund auf zu überarbeiten. Im Frontend sind daher zum jetzigen Zeitpunkt erst Grundfunktionen eines Datenmanagements implementiert. Anwender können zwar Tabellen erstellen und die enthaltenen Daten bearbeiten. Die Entwickler arbeiten jedoch noch an weiteren Funktionen für

das Frontend. Per Drag & Drop lassen sich Abfragen auf einfache Weise auch von SQL-Einsteigern zusammenstellen, wie **Abbildung 1** zeigt.

Sind von einer Abfrage mehrere Tabellen betroffen, muss der Anwender die Relationen zwischen diesen zuvor im Datenbankschema-Editor festlegen. Auch das geschieht per Drag & Drop. Für erfahrenere Benutzer, die eine direkte SQL-Eingabe bevorzugen, gibt es einen Editor mit Syntax-Highlighting und History. Diese selbst geschriebenen SQL-Abfragen lassen sich später im Entwurfsmodus ansehen und weiterverarbeiten. Sowohl im Entwurfsmodus als auch bei der direkten SQL-Eingabe speichert Kexi lediglich die pure SQL-Anweisung. Auch fehlerhafte Statements speichert das Programm ab. Sie lassen sich dann nachträglich korrigieren.

Die Programmierer haben eine integrierte Datenbankengine namens KexiSQL entwickelt. Mit ihr erstellen Anwender in kurzer Zeit und unkompliziert kleine Projekte. Alle relevanten Daten

landen in einer Datei, die Benutzer zum Beispiel per E-Mail untereinander austauschen oder auf einer Webseite veröffentlichen. KexiSQL ist von der populären Engine SQLite abgeleitet.

Obwohl Kexi ein natives Datenbankformat hat, ist es möglich, auf verschiedene andere DBM-Systeme wie MySQL und PostgreSQL zuzugreifen. Sowohl für den Benutzer als auch für den Programmierer lässt sich Kexi unabhängig vom darunter liegenden DBM-System bedienen. Ein ODBC-Treiber ist in Arbeit.

Zurück in die Zukunft

In der Beta-1-Version waren zahlreiche weitere Module wie Formulare, Berichte und Makros enthalten. Um aber eine größere Flexibilität in der Architektur des Programms zu erreichen, hat sich das Entwicklerteam entschlossen, die zugrunde liegenden Bibliotheken neu zu konzipieren und erst danach schrittweise die darauf aufsetzenden Komponenten zu realisieren. Der Benutzer profitiert dann von einem ausgereiften und stabilen Frontend. Momentan arbeiten die Entwickler noch an der Stabilisierung des API.

Es ist geplant, QT-Designer-kompatible Formulare zur Verfügung zu stellen, die mit Hilfe der KParts-Technologie Inhalte von Blob-Feldern (Binary Large Object) einbetten. Mit diesen Feldern erstellt ein Anwender dann Formulare, in denen Kexi direkt PDF-Dateien und Bilder anzeigt oder MP3-Dateien abspielt. Für Berichte planen die Programmierer den Berichtsgenerator Kugar **[3]** zu verwenden. Dieser wird für KOffice 2.0 größtenteils neu geschrieben und soll neben einer besseren Benutzerschnittstelle auch

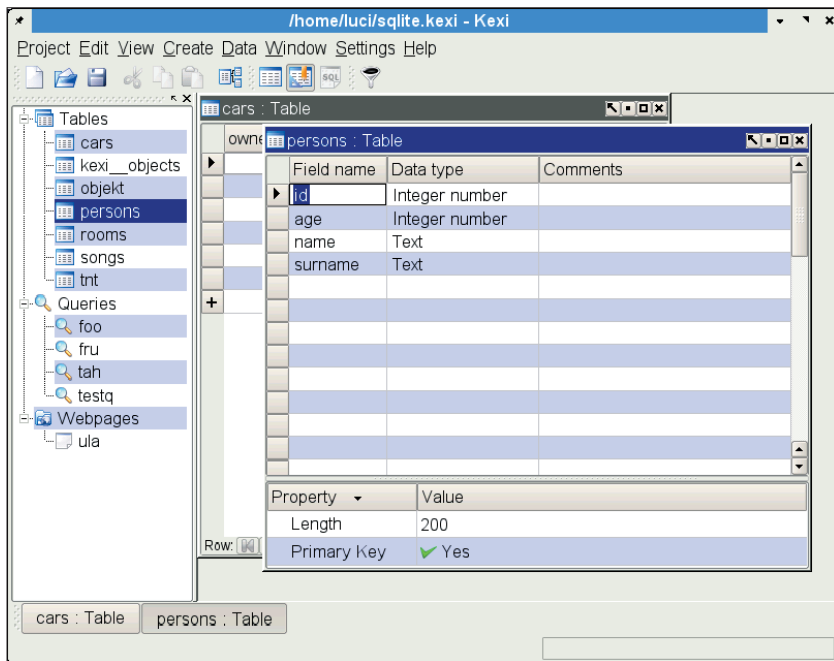


Abbildung 1: Mit Kexi verwalten Anwender komfortabel ihre Datenbanken. Obwohl das Programm noch in einer frühen Betaphase steckt, beherrscht es bereits die Grundfunktionen, wie hier das Anlegen von Tabellen.

das Zoomen unterstützen. Statt der Skriptumgebung QSA (QT Script for Applications [4]) wird Kexi künftig KJS-Embed verwenden, eine ECMA-Skriptbasierte Umgebung für Makros.

Außerdem soll Kexi bald ein in vergleichbaren Umgebungen unbekanntes Modul enthalten: die Einbindung von HTML-Seiten. Mit diesem Modul gestalten zum Beispiel Firmen die User-Schnittstelle nach dem Design ihrer Homepage. Die Daten und Funktionen fügt Kexi durch die Verwendung von Javascript und speziellen URLs in der Vorlage ein (siehe [Abbildung 2](#)).

Ab der Release-Serie von KOffice 2.0 soll Kexi wieder in den Release-Zyklus von KOffice integriert sein. In diesem Stadium planen die Entwickler, dass KWord

für Serienbriefe auf die Ressourcen von Kexi zurückgreift und die Tabellenkalkulation KSpread Daten direkt in seine Tabellen importiert. Außerdem wird es möglich sein, Diagramme in KSpread zu erstellen, die direkt aus einer Datenbank gespeist sind.

Hoch hinaus

Doch mit all diesen Funktionen geben sich die Entwickler noch nicht zufrieden. Sie wollen in späteren Versionen eine komplette RAD-Umgebung (Rapid Application Development) bereitstellen, die es auch erlaubt, Shared Libraries zu generieren. Programmierer bauen dann fertige Projekte wie Tabellen, Formulare oder Berichte direkt in ihre eigene Soft-

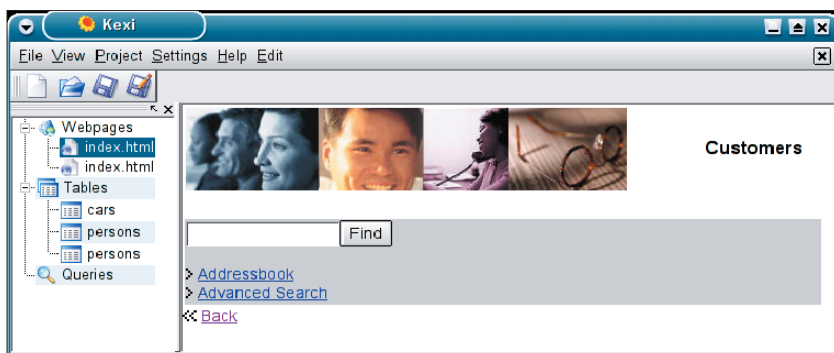


Abbildung 2: Mit Kexi sollen Anwender in Zukunft HTML-Formulare erstellen können. So kreieren Firmen beispielsweise Datenbankformulare, die sich der Website angleichen.

ware ein. Außerdem wird Kexi ausführbare Dateien erstellen, die auf Systemen ohne installiertes Kexi laufen. Das Entwicklerteam plant auch ein Quantal-Plugin zu schreiben. Durch die Verbindung des Datenbankprogramms mit der Web-Entwicklungssoftware würde es Programmierern vereinfacht, dynamisch generierte Webseiten zu erstellen, indem das Plugin zum Beispiel automatisch PHP-Code erzeugt.

Die Firma Open Office Polska unterstützt die Entwicklung von Kexi aktiv. Sie ist daran interessiert, das Programm in Open Office zu integrieren, um es ihren Kunden auf deren Plattformen anbieten zu können. Derzeit läuft Kexi auf diversen Unix-Architekturen und Microsoft Windows.

Kern und Spaltung

Kexi ist hauptsächlich in C++ geschrieben, basiert auf KDE und QT und ist stark modularisiert. Die eigentliche ausführbare Datei ist gerade mal 13 KByte groß. Hinzu kommt noch eine Menge Bibliotheken und Plugins. Es gibt drei Hauptbibliotheken: »KexiCore«, »Kexi-Widgets« und »KexiDB«. KexiCore dient als Basis für die Plugins und verwaltet das Hauptfenster mit der Modulansicht und den Plugin-Fenstern. Das Haupt-

fenster nutzt das in KDE 3.2 neu hinzugekommene KMDI (Multi Document Interface); für ältere KDE-Versionen gibt es eine Kompatibilitätsbibliothek. Die KDE-Entwickler haben dieses Interface speziell für integrierte Entwicklungsumgebungen geschaffen, auch KDevelop nutzt dieses Interface.

Mit Hilfe von KMDI zeigt KexiCore ein Datenbankprojekt in verschiedenen Modi an: als Karteireiter-Ansicht (TabPage-Modus und Ideal-Modus), mit allen Unterfenstern in einem großen Hauptfenster (Child-Frame-Modus) und mit allen Fenstern einzeln (Toplevel-Modus). KexiCore ist außerdem die Schnittstelle, die Kexi in Anwendungen wie KWord oder KSpread einbettet.

Kexiwidgets kümmert sich um gemeinsam genutzte Widgets wie Tabellen und SQL-Query-Editoren. Jedes Widget-Modul ist ein Plugin, es gibt ein Tabellen-, ein Abfragen- und ein Relationen-Plugin. Da sie nur bei Bedarf nachgeladen werden, startet das Programm sehr schnell. Eigene Software in Kexi zu integrieren ist dank der Plugin-Funktionalität sehr komfortabel, die Entwickler müssen lediglich die Schnittstelle kennen. Da Kexi unter der LGPL steht, können auch Firmen Plugins entwickeln und verkaufen. Hier wäre zum Beispiel eine Anbindung an kommerzielle Software denkbar.

KexiDB ist ein Datenbank-Abstraktions-Layer, der ähnlich wie QSQL und ODBC funktioniert. Die Entwickler waren mit beiden Bibliotheken aber nicht wirklich zufrieden und beschlossen daher eine eigene Bibliothek zu erstellen. KexiDB nutzt für jedes Datenbanksystem einen eigenen Treiber.

Schemaklassen

Im Gegensatz zu vielen anderen Bibliotheken dieser Art geht KexiDB sehr flexibel mit unterschiedlichen Datenbankdesigns um. Der Programmierer benutzt einfach die Abstraktionsfunktionen aus KexiDB für jede Datenbank. Trotz des hohen Abstraktionsgrades arbeiten die Operationen ohne nennenswerten Geschwindigkeitsverlust und bedürfen keines komplizierten API.

KexiDB liefert umfangreiche Metainformationen, die die anfragende Applikation in Form von Schemaklassen erhält. Die Schemata sind editierbar, um zum Beispiel neue Felder in Tabellen oder Abfragen einzufügen. Steht das ergänzte Schema fest, übersetzt der entsprechende Treiber die Metainformationen in SQL-Anweisungen.

Die KexiDB-Bibliothek stellt somit sicher, dass der Entwickler so wenig fest programmierte Abfragen wie möglich

Eine Alternative: Rekal

Neben Kexi gibt es für KDE noch weitere Datenbank-Frontends. Eines davon ist Rekal, das ursprünglich die Firma The Kompany.com [6] als kommerzielles Produkt entwickelte. Seit Ende letzten Jahres steht Rekal unter der GPL. Das Programm benutzt die QT-Bibliotheken und läuft daher auf Linux, Mac OS X und Windows. Mit Rekal lassen sich Tabellen anlegen, Daten eingeben, Formulare erstellen und sogar eine Datenbankapplikation per Skript erstellen und als eigene Anwendung starten. Diese Funktion macht Rekal zu einer RAD-Applikation, um von Rekal unabhängige Programme zu erstellen.

Einer der Hauptunterschiede zwischen Kexi und Rekal ist, dass Ersteres direkt in die KDE-Umgebung integriert ist, während Rekal unabhängig arbeitet. Außerdem benutzt Kexi ECMA-Script für die Skriptprogrammierung, Rekal arbeitet mit Python. Von Rekal gibt es auch eine Version für den Zaurus-PDA, die von The Kompany.com entwickelt wurde. Im Gegensatz zu Kexi ist Rekal bereits produktiv einsetzbar.

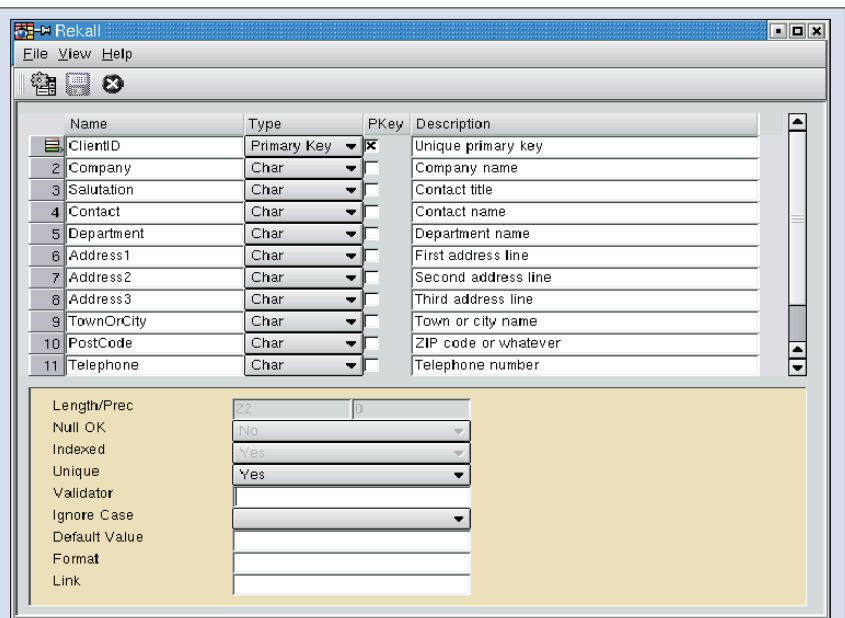


Abbildung 3: Eine Alternative zu Kexi ist das Datenbank-Frontend Rekal. Seit Ende letzten Jahres steht es unter der GPL. Rekal beherrscht viele Funktionen und dient als RAD-Umgebung.

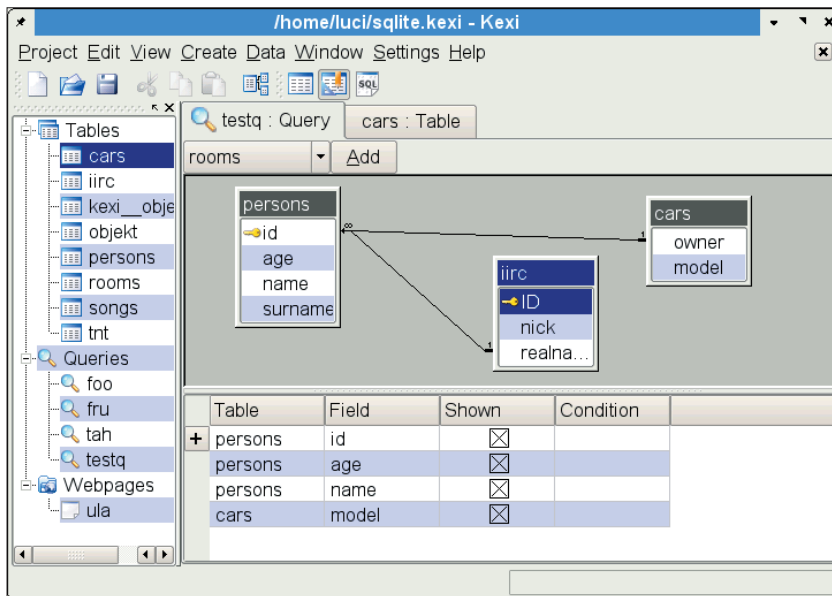


Abbildung 4: Abfragen lassen sich mit Kexi per Point & Click erstellen. Auch die Relationen zwischen den einzelnen Tabellen zeigt Kexi an.

benötigt und Problemen mit SQL-Dialekten oder Missinterpretationen aus dem Weg geht. Außerdem bietet KexiDB einen Parser, der gespeicherte SQL-Abfragen wieder in das Metaformat umwandelt. Daraus erstellt Kexi dann die Abfrage-Oberfläche wie in [Abbildung 4](#).

Internationalisierung

Eine Besonderheit des Kexi-Parsers besteht darin, Fehlermeldungen, zum Beispiel bei falschen Abfragen, in der jeweils eingestellten Sprache auszugeben. Zudem ermöglicht es der Parser, im User Interface jene Stelle zu markieren, an der der Fehler aufgetreten ist. So springt der Cursor zum fehlerhaften Code im Eingabefenster.

Der Parser ist fehlertolerant und so konzipiert, dass er ein eigenes SQL-Subset spricht, mit dem er die SQL-Dialekte der populärsten Engines versteht. KexiDB unterstützt noch ein erweitertes Transaktionssystem. So sind – sofern von der Datenbank unterstützt – Named und Nested Transactions möglich. Für eine spätere Release plant das Entwicklerteam asynchrone Datenbankoperationen zu unterstützen.

Programmierer können KexiDB auch komplett unabhängig von Kexi nutzen, es wird möglicherweise später eine eigene Version geben, die ohne KOffice funktioniert und nur QT benötigt. So-

bald das zugrunde liegende API von Kexi für eine Vielzahl möglicher Anwendungen ausreichend stabil ist, profitieren Entwickler von verschiedenen Technologien wie Engine-unabhängigen Datenbankzugriffen, einem toleranten Parser und einem einfach zu handhabenden Transaktionssystem.

Allerdings braucht Kexi wohl noch etwas Zeit, bevor es dem Endanwender auch ein leistungsfähiges und vielseitig einsetzbares grafisches Interface bietet. Die Entwickler planen im dritten Quartal 2004 eine auch für User interessante Testversion zu veröffentlichen. Die nächste stabile Version kommt mit KOffice 2.0. (mwe) ■

Infos

- [1] Kexi: [<http://www.kexi-project.org/>]
- [2] Quanta: [<http://quanta.sourceforge.net/>]
- [3] Report-Generator Kugar: [<http://www.koffice.org/kugar/>]
- [4] Qt Script for Applications: [<http://www.trolltech.com/products/qsas/>]
- [5] Open Office Polska: [<http://www.openoffice.com.pl/>]
- [6] The Kompany.com: [<http://www.thekompany.com/>]

Der Autor

Lucijan Busch besucht die 4. Klasse der Handelsakademie I in Klagenfurt. Er ist Initiator und Maintainer des Kexi-Projekts.