

KEXI – la prima guida italiana - credo (MS Access per GNU/Linux?)

Augusto Scatolini (webmaster@comunecampagnano.it)
Ver. 1.0 Marzo 2010



KEXI

Introduzione

Obiettivi

Database

Tabelle

Database Relazionale

Ora entra in gioco il DataBase Relazionale
C'è un problema

Maschere (Forms)

Pulsanti

Ricerca

Stampa

Menu

Introduzione

Kexi è un programma che permette di sviluppare rapidamente applicazioni per archiviare dati, i cosiddetti DataBase, in gergo si dice che è un RDAD (Rapid Database Application Development). Con le dovute differenze e specificità si può dire che fa parte del gruppo RDAD che comprende MS Access e OpenOffice.org Base. E' un progetto giovane, acerbo, ma molto interessante.

E' interessante, a mio avviso, proprio per la sua limitazione, infatti è meno raffinato di OpenOffice.org Base e molto meno di MS Access, ma è proprio la complessità di questi ultimi due prodotti che ne hanno determinato una scarsa o nulla usabilità e utilizzazione da parte degli "utenti normali".

Kexi potrebbe rappresentare una buona opportunità per avvicinarsi al fantastico mondo dei database, infatti io ritengo che il "database" sia la summa dell'informatica ma purtroppo per la sua astrusità (del database) è relegato nelle mani di pochi specialisti.

Kexi è un programma per GNU/Linux rilasciato con licenza GPL, ne esiste una versione (porting) per Windows ma è a pagamento, esiste una versione demo (gratuita) per Windows che può essere usata tranquillamente per piccole applicazioni perché ha pochissime limitazioni (le principali sono che non importa dati da archivi MS Access – come invece fa la versione originale – e non si possono creare più di 5 oggetti per ogni categoria che sono Tabelle, Query, Maschere e Script.

Fa parte del pacchetto Koffice, nel progetto KDE del quale esiste un porting per Windows.

Per ultimo, non esiste praticamente documentazione, tanto meno in Italiano e ancora non esiste una sua localizzazione in lingua italiana a meno che non si installi su Gnome anche KDE localizzato.

Home page: <http://www.kexi-project.org/>

Download versione per Windows: <http://www.kexi.pl/en/Download>

Gli utenti GNU/Linux possono scaricare i rispettivi pacchetti DEB o RPM o compilarne il sorgente

Obiettivi

Gli obiettivi di questa guida sono illustrare le principali funzionalità di Kexi costruendo una applicazione prototipo "la solita Biblioteca" e illustrare - strada facendo - i fondamenti del database.



Alla fine di questa guida avremo creato un database composto da un menu che ci permetterà di gestire le varie Tabelle, le varie Query e la Maschera principale dell'applicazione per la gestione dei libri. Per ogni libro sarà disponibile una scheda con un certo numero di informazioni inclusa la foto/immagine dell'autore e tutti i controlli classici di un database, cerca, cancella, modifica, salva, collegamento agli oggetti, chiudi, esci, ecc.

DataBase

Un DataBase, tipicamente, è composto da un certo numero di Tabelle, di Query, di Maschere e di Macro o Script (scritti in vari linguaggi) che vengono associati generalmente a Pulsanti che quando vengono premuti eseguono, appunto, le istruzioni contenute nelle Macro o Script.

Nei sistemi RDAD, tutti questi oggetti sono contenuti in un unico file, nel nostro caso il file si chiamerà Biblioteca.kexi

Tabelle

La Tabella di un DataBase è un oggetto che contiene tutte le informazioni archiviate, apparentemente è simile a un foglio elettronico dove le colonne rappresentano i CAMPI mentre le righe rappresentano i RECORD

Id_autore	autore	nato a	nato il	morto a	morto il	foto
1	Leopardi Giacomo	Lecce	19/5/1956	Palermo	20/6/1956	
2	Marx Calo	Bombay	29/6/1832	Zagarolo	12/12/2009	
3	Alighieri Dante	Napoli	2/2/1922	Brindisi	18/11/1998	
4	Attivissimo Paolo	Cairo	26/11/1950			

In questo esempio la Tabella contiene **4 Record**, ogni Record è composto da **7 Campi**.

I Campi possono essere di vario tipo, contatore, testo data e immagini.

Se il numero dei Campi è limitato si può lavorare direttamente sulla Tabella, aggiungere, modificare, cancellare, cercare e indicizzare. Quando il numero di Campi è difficilmente gestibile attraverso la Tabella è più utile e comodo lavorare tramite una Maschera.

Database Relazionale

Il DataBase Relazionale serve a risolvere il problema della Ridondanza inutile e pericolosa.

La Tabella dove verranno registrate le informazioni dei libri sarà qualcosa come questo:

ID libro	autore	genere	lingua	titolo
1	Leopardi Giacomo	storia	italiano	Pinocchio
2	Marx Calo	religione	arabo	Biancaneve
3	Alighieri Dante	religione	italiano	Il Capitale
4	Attivissimo Paolo	storia	italiano	Hackers

In questa Tabella oltre al campo immagini (che vanno inserite) c'è un campo di tipo Contatore (il primo) che è automatico e quindi non va digitato, c'è il campo titolo che è diverso per ogni libro e quindi va digitato poi ci sono informazioni (campi) come autore, genere e lingua che sono ripetitivi. Nessuno ci impedisce di scriverli ogni volta (per ogni libro) ma oltre alla perdita di tempo può accadere di commettere degli errori per cui una volta si scrive "italiano" e un'altra "italiana".

Questo comporterà errori nelle future analisi dell'archivio perché (in questo esempio) verrebbe conteggiata due volte.

E' più funzionale creare una tabella per i libri e delle Tabelle specifiche per gli autori, per il genere e per la lingua e metterle in relazione in modo tale il dato dell'autore (per esempio) non deve essere digitato ma selezionato dalla relativa Tabella, lo stesso per il genere e per la lingua.

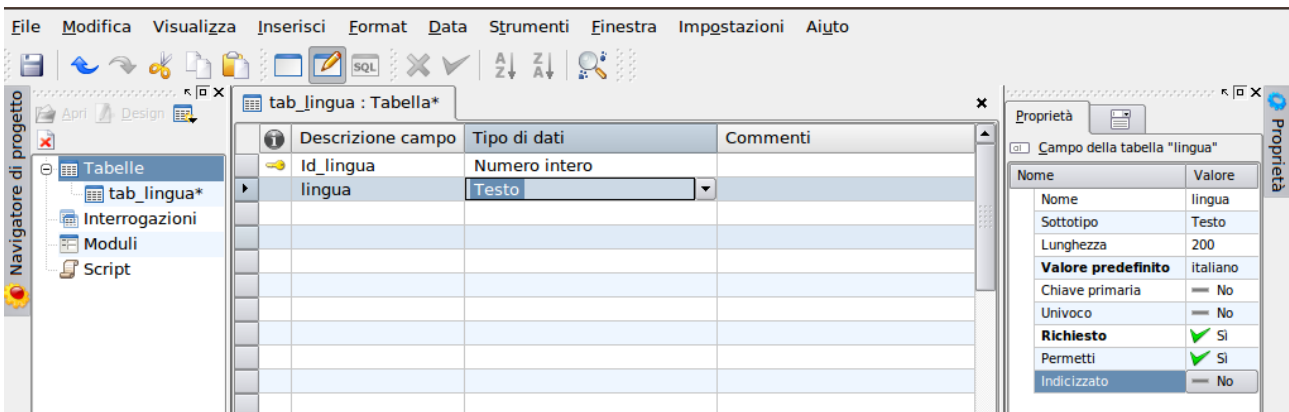
In questo modo si ha anche un altro enorme vantaggio: supponiamo di aver registrato nella Tabella lingua la lingua "INDI", poi supponiamo di registrare 3000 o 4000 libri in lingua "INDI" e poi supponiamo di scoprire che invece di "INDI" avremmo dovuto scrivere "HINDI"

E' sufficiente aprire la Tabella lingua e modificare la parola INDI in HINDI, automaticamente – per via della relazione – tutti i 3000 o 4000 libri avrebbero la parola INDI sostituita con HINDI.

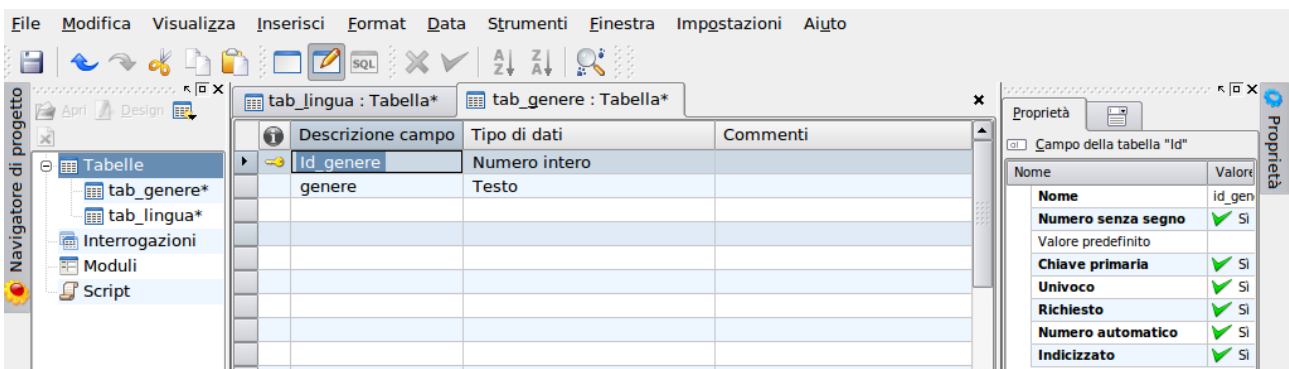
Quindi iniziamo a costruire la Tabella **tab_lingua**

Cliccando con il tasto destro su Tabelle e poi su Crea Oggetto Tabella si apre in modalità struttura una nuova Tabella. Inseriamo solo due campi, il contatore automatico **id_lingua** di tipo "numero intero" e la **lingua** di tipo Testo. Nel pannello a destra Proprietà si può stabilire se il campo è richiesto, se è **chiave primaria** ecc.

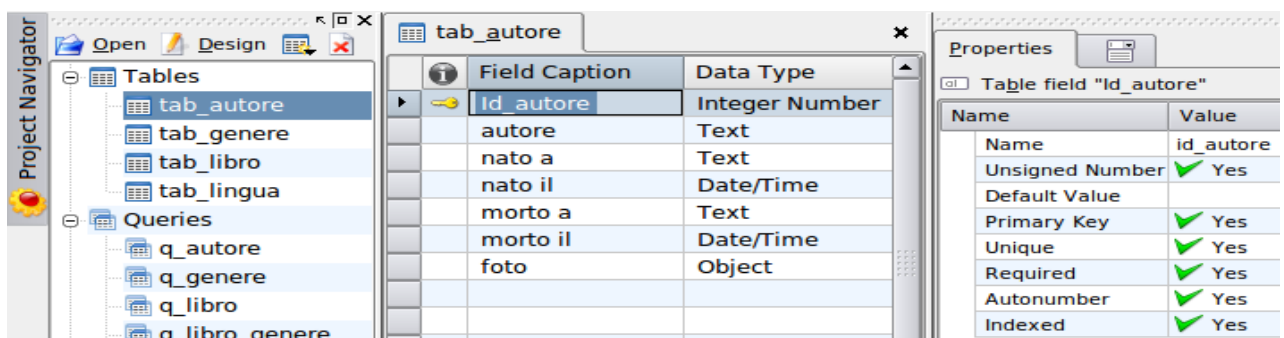
Se ci si dimentica di creare un campo chiave primaria, al momento del salvataggio della Tabella il programma ce lo ricorda e propone di crearne una in automatico.



Analogamente creeremo la Tabella **tab_genere**



e la Tabella **tab_autore** con il campo Object (OLE per Windows) per il campo foto



Ora entra in gioco il DataBase Relazionale

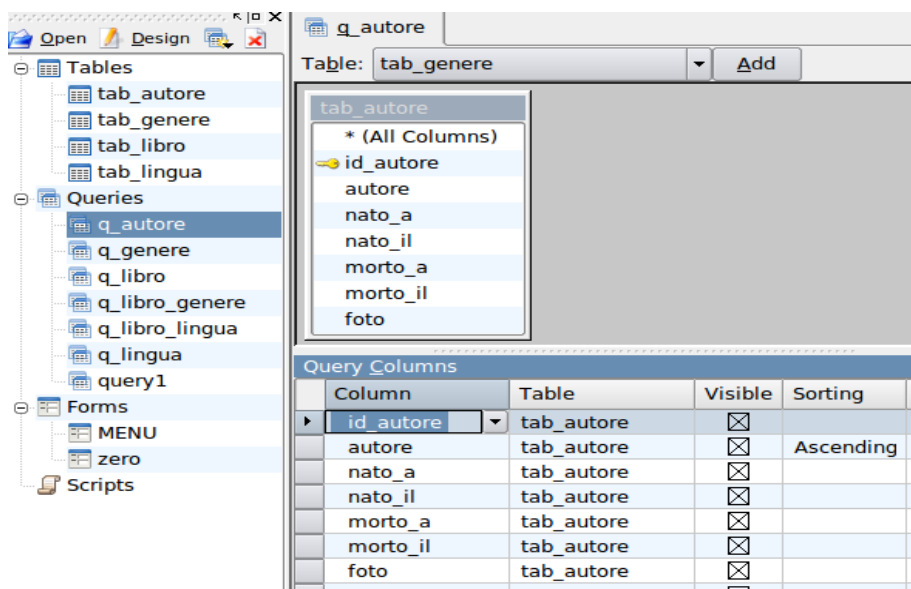
Per la costruire la struttura della Tabella **tab_libro** dobbiamo indicare al programma che le informazioni autore, genere e lingua sono già registrate nelle corrispondenti tabelle. In questo modo quando operativamente dovremo inserire in una scheda di registrazione un autore (per esempio) apriremo una **finestra a tendina** (combo box) dove appariranno tutti gli autori e sarà sufficiente selezionare quello desiderato.

C'è un **problema**, operando in questo modo quando apriremo il menu a tendina è vero che appariranno tutti gli autori ma l'elenco **non sarà in ordine alfabetico** e pertanto sarà molto difficile selezionare l'autore prescelto.

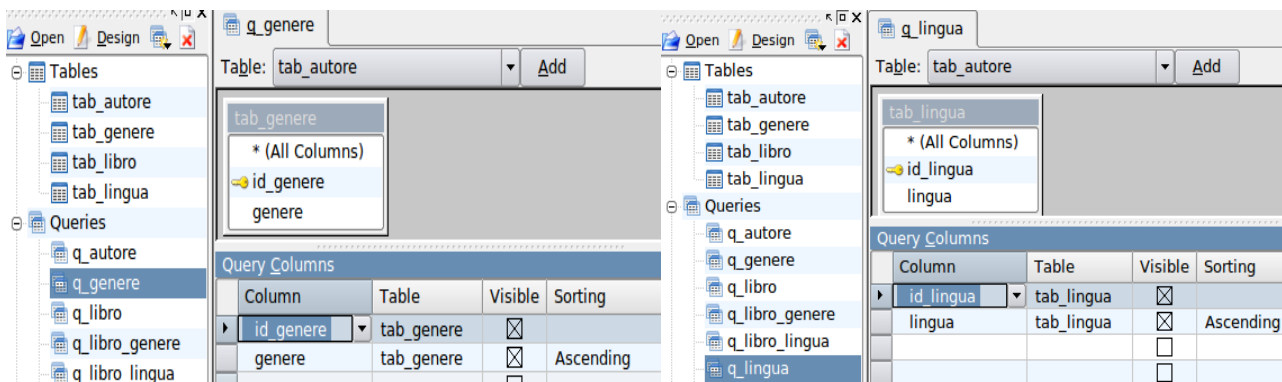
Questo problema si risolve utilizzando una apposita **Query al posto della Tabella**.

La query in oggetto altro non è che la Tabella ordinata con un certo criterio, in questo caso con i nomi degli autori in ordine alfabetico.

Quindi creiamo la Query (Interrogazione) **q_autore** cliccando con il tasto destro del mouse su Query, selezioniamo la Tabella di origine (tab_autore), selezioniamo tutti i campi applichiamo il criterio "Ascending" sul campo "autore"



Analogamente creiamo la Query **q_genere** e **q_lingua**



Vediamo in pratica in pratica la differenza di visualizzazione tra Tabelle e Query:

Questa è la Tabella q_autore (non ordinata)

Id_autore	autore	nato a	nato il	morto a	morto il
1	Leopardi Giacomo	Napoli	12/1/1950	Calcutta	22/3/1970
2	Alighieri Dante	Milano	20/11/1930	Napoli	31/12/1960
+ <i>hero automatico</i>					

Questa è la Query q_autore con i nomi ordinati alfabeticamente

autore	nato a	nato il	morto a	morto il
Alighieri Dante	Milano	20/11/1930	Napoli	31/12/1960
Leopardi Giacomo	Napoli	12/1/1950	Calcutta	22/3/1970

Tabella tab_genero e Query q_genero

Id_genero	genere
1	storia
+ <i>hero automatico</i>	
	religione

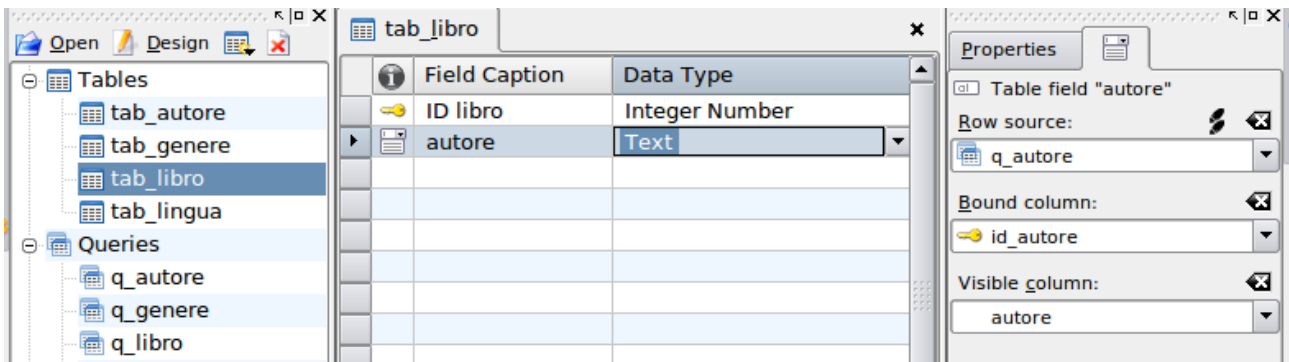
genere
religione
storia

Tabella tab_lingua e Query q_lingua

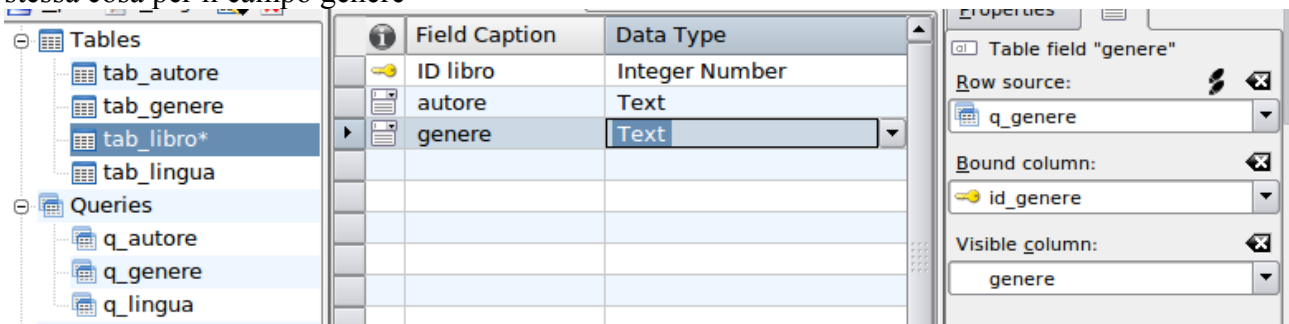
Id_lingua	lingua
1	italiano
2	arabo
+ <i>hero automatico</i>	
	italiano

lingua
arabo
italiano

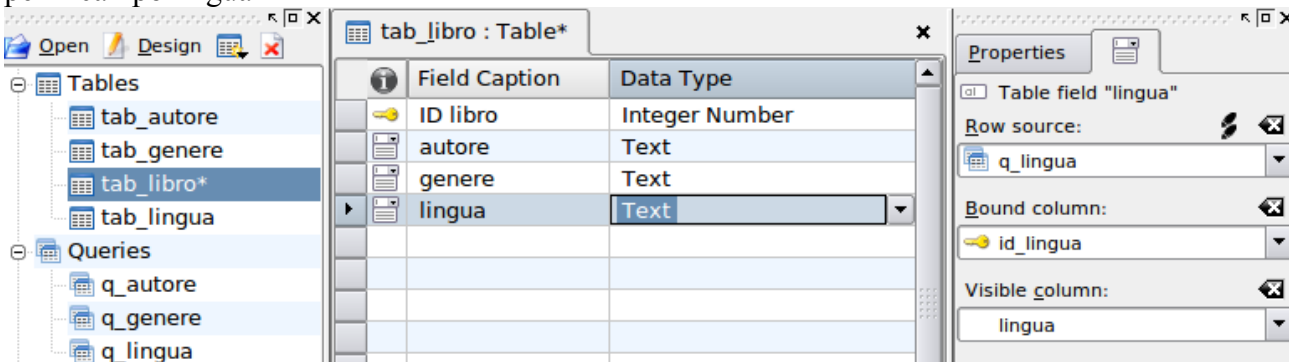
Ora si può creare la Tabella **tab_libro**. Dopo il solito campo contatore, creiamo il campo autore, si passa nella scheda a destra di proprietà (**lookup**) e si indica la **Query di provenienza (Row source)** “q_autore”, il campo collegato (**Bound column**) “id_autore” e il campo visibile (**Visible column**) “autore”



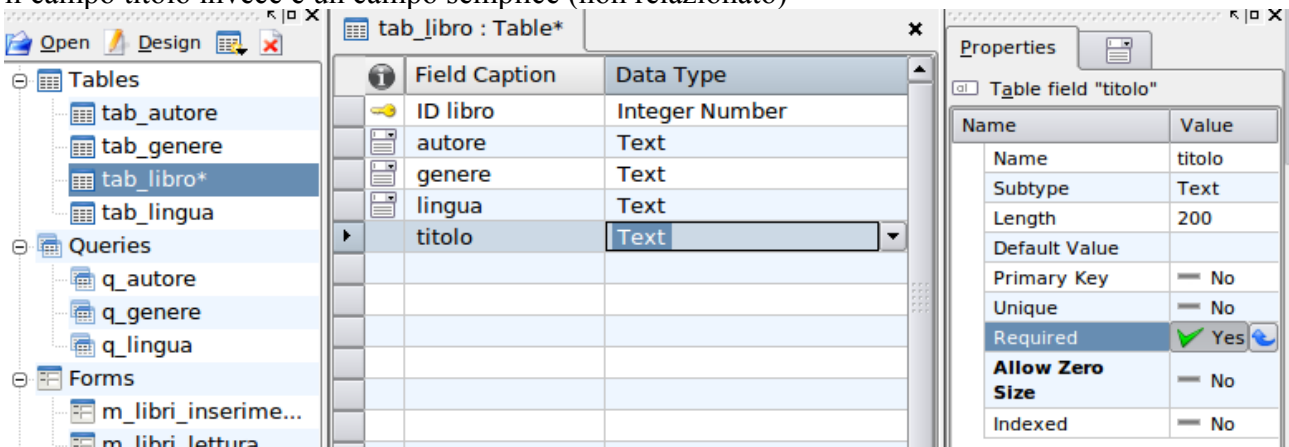
stessa cosa per il campo genere



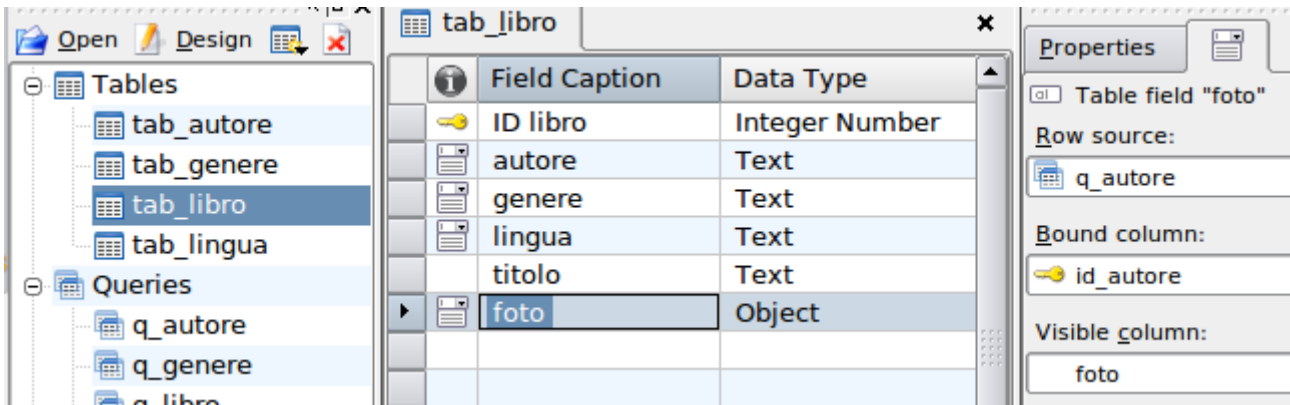
per il campo lingua



il campo titolo invece è un campo semplice (non relazionato)



anche il campo foto è un campo relazionato alla Query q_autore



Quando si lavorerà sulla Tabella tab_libro inserendo i record si dovrà scrivere solamente il titolo del libro mentre tutte le altre informazioni saranno relazionate (tramite la selezione nel menu a tendina) ai dati ordinati (Query) presenti sulle relative Tabelle.

Ovviamente prima di inserire un nuovo libro bisogna accertarsi che le informazioni “relazionate” siano presenti nelle Tabelle di provenienza.



Ovviamente, una volta creata e popolata la Tabella tab_libro, tramite Query ad hoc si possono ordinare i dati in tutti i modi possibili (per autore, per genere e per lingua)

autore	genere	lingua	titolo
Attivissimo Paolo	storia	italiano	Hackers
Alighieri Dante	religione	italiano	Il Capitale
Marx Calo	religione	arabo	Biancaneve
Leopardi Giacomo	storia	italiano	Pinocchio

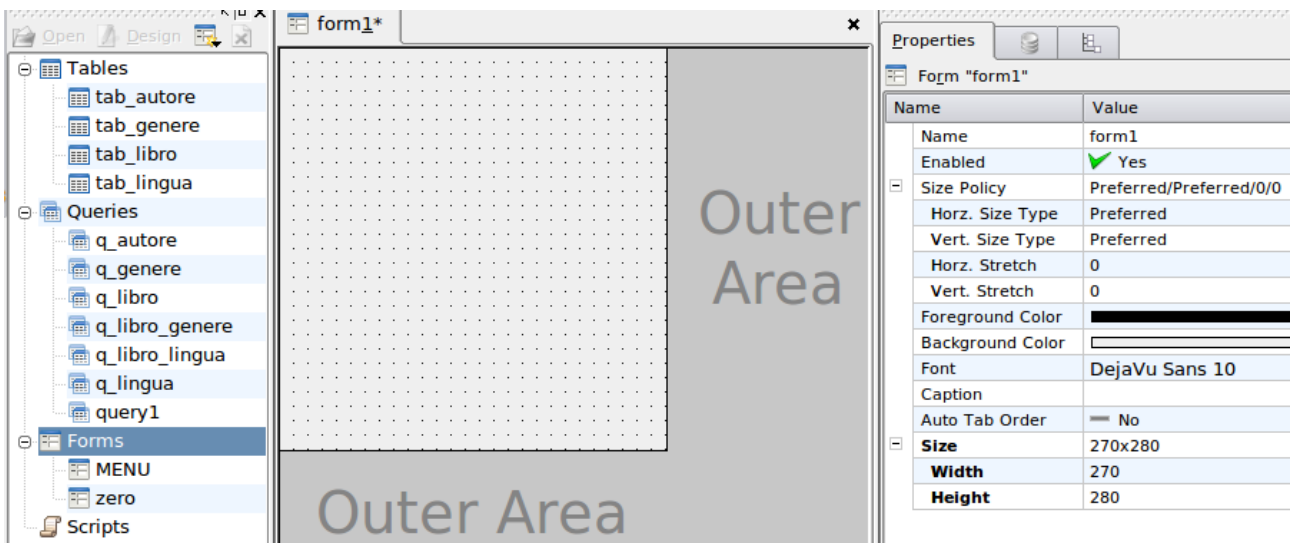
autore	genere	lingua	titolo
Marx Calo	religione	arabo	Biancaneve
Alighieri Dante	religione	italiano	Il Capitale
Leopardi Giacorr	storia	italiano	Pinocchio
Attivissimo Paol	storia	italiano	Hackers

autore	genere	lingua	titolo
Marx Calo	religione	arabo	Biancaneve
Leopardi Giacorr	storia	italiano	Pinocchio
Alighieri Dante	religione	italiano	Il Capitale
Attivissimo Paol	storia	italiano	Hackers

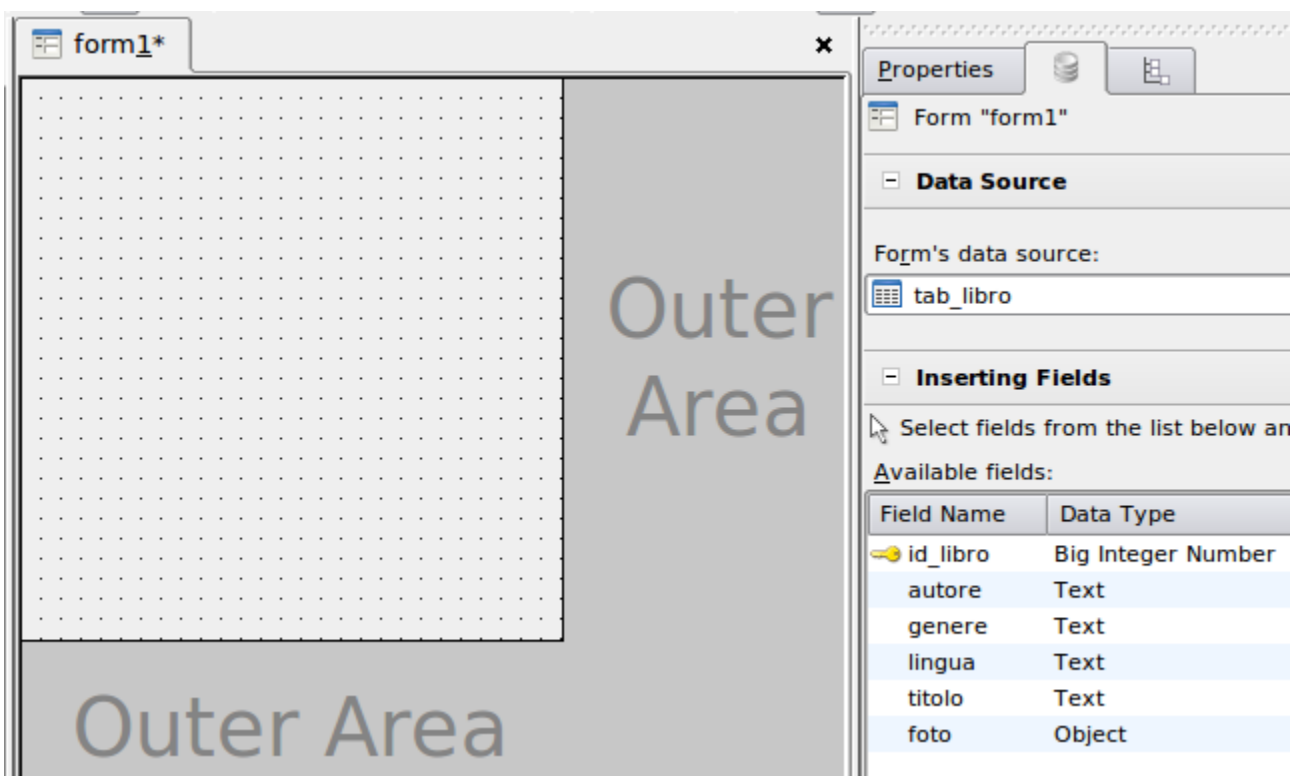
Maschere (Forms)

Ricapitolando, i dati sono registrati nelle Tabelle, relazionate o meno, queste possono essere manipolate (ordinate) tramite le Query. Le Maschere sono delle finestre attraverso le quali si possono gestire tutte le informazioni di un singolo record. L'uso delle Maschere non è affatto obbligatorio ma è indispensabile quando il numero di campi di un record è talmente alto da rendere difficoltosa la gestione di un record tramite la Tabella.

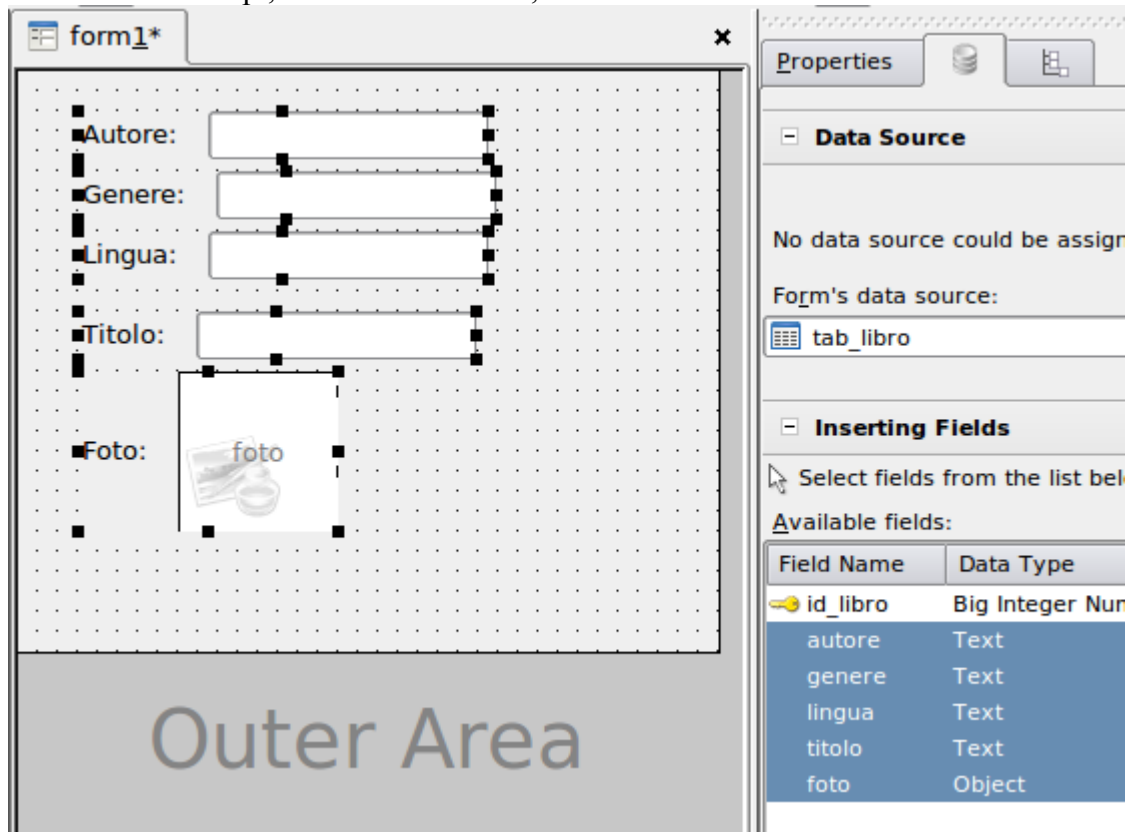
Cliccando con il tasto destro del mouse su Fors si può creare la struttura di una Maschera. Appare uno spazio grigliato dimensionabile, a destra appare la scheda delle proprietà dove si può operare per settare le proprietà come colore, dimensioni, ecc dell'oggetto selezionato nella Maschera



Nella scheda "Data Source" si seleziona la Tabella "sorgente" dalla quale prelevare i campi. Selezionando, per esempio, la Tabella tab_libro appariranno i campi disponibili "available fields".



Selezionando tutti i campi, escluso il contatore, e trascinandoli sulla maschera si otterrà:



A questo punto si possono spostare e sistemare i campi nella posizione più opportuna.

Operando opportunamente sui 14 controlli disponibili



1. Pointer (puntatore)
2. Auto Field
3. Label (etichetta)
4. Image Box
5. Text Box
6. Text Editor
7. Command Button (pulsante)
8. Combo Box (menu a tendina)
9. Check Box
10. Line
11. Frame
12. Group Box
13. Tab Widget
14. Spring (molla)

Si può ottenere qualcosa come mostrato nella figura seguente.

Come si può notare l'unico campo dove si deve scrivere è quello del titolo.

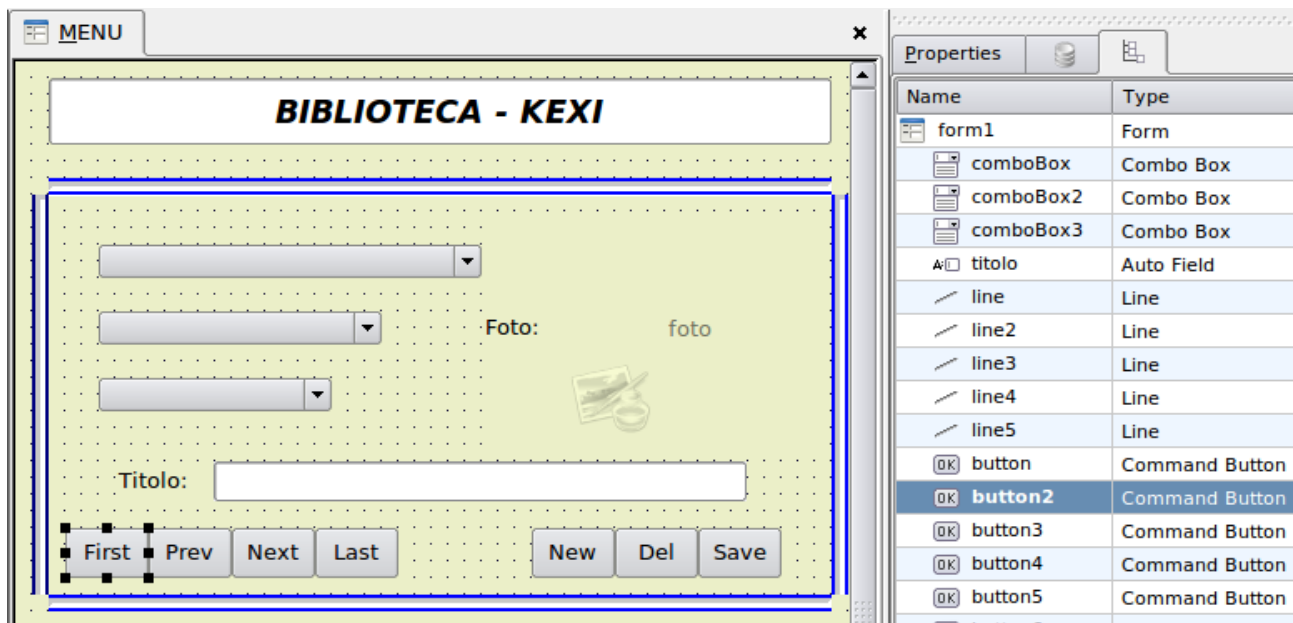
I campi autore, genere, lingua e foto sono dei combo box tramite i quali si selezionano gli opportuni valori dall'elenco ordinato delle Query alle quali i campi sono relazionati.

I pulsanti che aprono le Tabelle autore, genere e lingua servono per inserire i dati quando non appaiono nei menu a tendina.

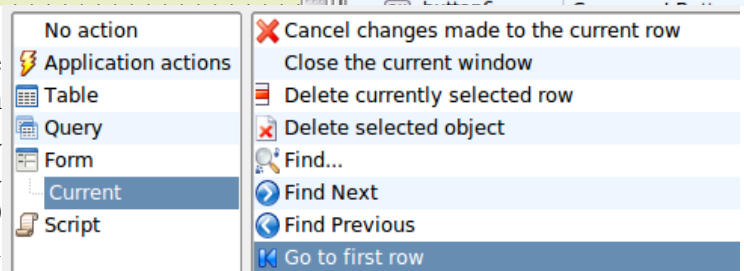


Pulsanti

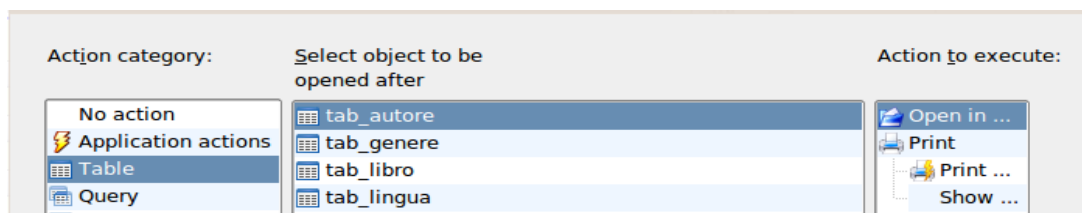
Selezionando un pulsante e aprendo la scheda “widget” (la terza) appare l'oggetto corrispondente.



Cliccando con il tasto destro del mouse sull'oggetto si può cliccare su “Assign action” - assegna azione - si aprirà una nuova finestra dove a sinistra si seleziona l'oggetto (in questo caso Form - Current) e a destra appariranno tutte le azioni possibili associabili al pulsante, in questo caso “Go to first row” - vai al primo record -.

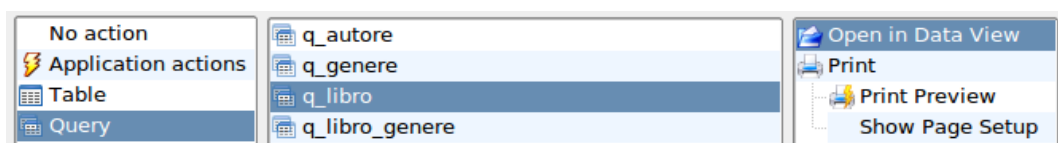


Molto utili sono i pulsanti Autori, Genere e Lingua quando nell'inserimento di un libro ci rendiamo conto che quell'autore non è presente nella tabella autori.
L'azione associata al pulsante Autori sarà Apri la Tabella tab_autore

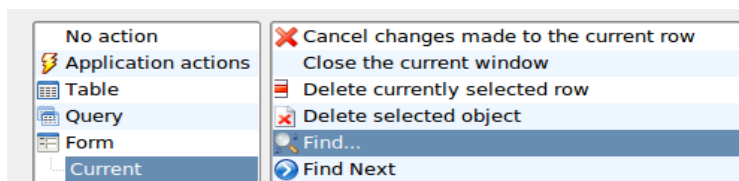


Ricerca

Per ricercare un determinato libro nel DataBase si possono usare due pulsanti il pulsante "View Query" che apre la Query q_libro (indicizzato per autore)

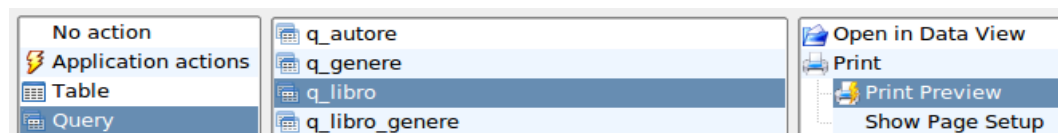


oppure il pulsante "Cerca" che apre appunto il modulo della ricerca

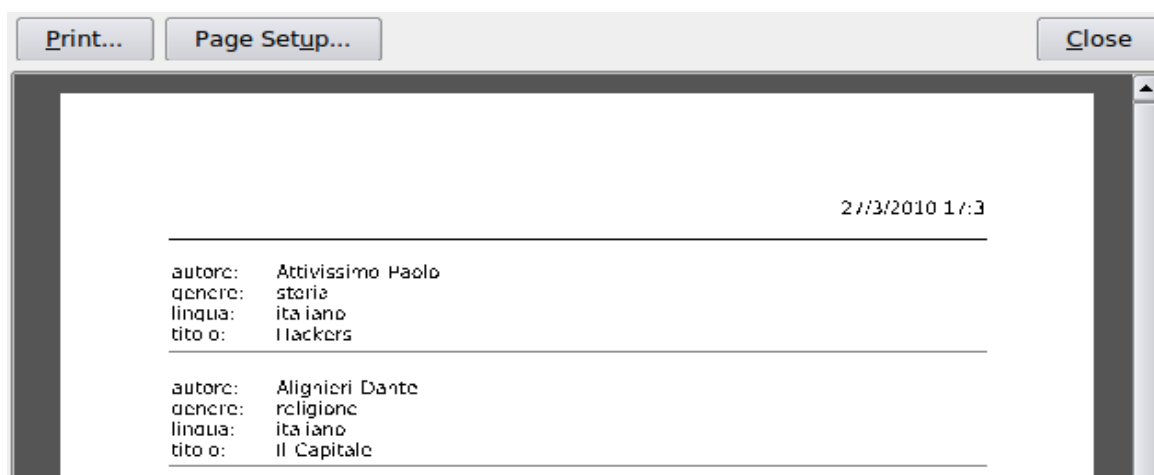


Stampa

Per la stampa del registro dei libri si può usare il pulsante "print Query" che è associato all'azione "print Preview"



che quando cliccato mostrerà qualcosa come questo



Menu

Il menu è una Maschera non associata ad alcuna Tabella o Query che contiene solamente un po' di grafica e dei pulsanti per aprire i vari oggetti: Tabelle, Query e Maschere



download http://www.comunecampagnano.it/gnu/mini-howto/kexi_la_prima_guida_italiana/biblioteca.zip

http://www.comunecampagnano.it/gnu/mini-howto/KEXI_la_prima_guida_italiana/biblioteca.kexi

FINE

Questo documento è rilasciato con licenza Copyleft
(tutti i rovesci sono riservati)
altre miniguide

<http://www.comunecampagnano.it/gnu/miniguide.htm>