

# SQLite (autore: Vittorio Albertoni)

## Premessa

La prima cosa che viene chiarita nella presentazione di SQLite che troviamo sul sito <https://sqlite.org/> è che SQLite non è paragonabile ai motori di database client/server come MySQL, Oracle, PostgreSQL, SQL Server e che non vuole competere con essi. SQLite, si dice, nasce per altri fini: primo tra tutti quello di essere un software molto leggero adatto per essere «embedded», cioè inserito in altri software all'interno dei quali può assolvere egregiamente la funzione di motore di gestione di database.

Tutto ciò significa che se vogliamo costruire un database destinato ad essere utilizzato contemporaneamente su più postazioni di lavoro, come può avvenire per il database dei correntisti di una banca o dei clienti di una multinazionale non possiamo certo ricorrere a SQLite. Ma se vogliamo costruire un database sul nostro personal computer, per i nostri usi personali o per la nostra piccola azienda, che possa essere utilizzato anche sul nostro tablet o sul nostro telefonino, SQLite è quello che fa per noi.

In questo manualetto mi propongo di presentare alcuni modi di utilizzare SQLite, dando per scontato che il lettore sappia in che cosa consista un database, in particolare un database relazionale.

Per un primo approccio alla materia, che può essere approfondita utilizzando uno dei tanti testi in circolazione o con una navigazione intelligente su Internet, rammento che un database, in italiano base di dati, è un archivio di dati strutturato per la gestione di informazioni e propongo un esempio.

Se abbiamo una biblioteca, o una cineteca, o una raccolta di dischi, al raggiungimento di un certo numero di elementi finisce, come si suol dire, che perdiamo il conto. Finisce che non ricordiamo più quali brani abbiamo di Brahms piuttosto che di Verdi, quali brani abbiamo diretti da Riccardo Muti, quali brani abbiamo della musica francese del '700, ecc.

Per disporre di queste informazioni, nel momento in cui non possiamo più fare affidamento sulla nostra memoria, dobbiamo creare un archivio che contenga tutti i dati che concorrono a formarle. Per sapere quali brani abbiamo di Brahms, e di quanti altri autori, dobbiamo creare un archivio che contenga autore e titolo del brano. Per sapere quali brani e di quali autori abbiamo diretti da Muti, dobbiamo creare un archivio che contenga autore, titolo del brano e interprete. E così via.

Questo archivio si organizza in una tabella nella quale ciascuna riga è dedicata a un brano e ciascuna colonna è dedicata ai dati che sono interessanti per le informazioni che vogliamo avere. Nel caso ci interessi solo sapere i brani per autore ci basteranno due colonne, una dedicata agli autori, una dedicata ai titoli dei brani. Nel caso ci interessino anche gli interpreti occorreranno tre colonne, una dedicata agli autori, una dedicata ai titoli e una dedicata agli interpreti. E così via.

Nel linguaggio dei database le righe della tabella costruita a questi fini si chiamano **record** e le zone del record corrispondenti alle colonne si chiamano **campi (fields)**.

Visto che si tratta di una tabella, in mancanza d'altro, potremmo costruirla utilizzando un foglio di calcolo. Anzi, i fogli di calcolo che ci sono in circolazione hanno funzioni per l'ordinamento e la ricerca di dati organizzati in tabella. Tuttavia, anche per le applicazioni più elementari, il foglio di calcolo si rivela ben presto inadatto per database di una certa dimensione e con pretese classificatorie di un certo tipo.

Partiamo dalla più banale esigenza: quella di gestire un archivio contenente semplicemente autore e titolo di un brano musicale. Se utilizziamo un foglio di calcolo in-

testiamo due colonne affiancate, una «autore» e una «titolo» e, riga per riga, inseriamo autore e titolo dei nostri brani musicali. Se abbiamo 18 opere di Ciaikovski dobbiamo ovviamente scrivere 18 titoli, molto probabilmente diversi tra loro, nel campo dedicato al titolo e dobbiamo anche scrivere per 18 volte la stessa cosa nel campo dedicato all'autore, augurandoci di scriverla sempre nello stesso modo: se una volta scriviamo Ciaikovski, un'altra Tchaikovsky, un'altra ancora Chaikovskij vai poi a tirare fuori i titoli delle opere di Ciaikovski che abbiamo nella nostra raccolta di dischi!

Se utilizziamo un altro modello di database in cui facciamo due tabelle, una per gli autori, l'altra per i titoli, entrambe con due colonne, scriveremo una sola volta, badando bene di scriverlo giusto, il nome dell'autore nella tabella degli autori, nella quale gli verrà a corrispondere, nella colonna a fianco, un numero identificativo assegnato solo a lui e scriveremo i nostri 18 titoli nella tabella dei titoli, nella quale, nella colonna a fianco indicheremo il numero dell'autore: il numero che compare in entrambe le tabelle costituisce la relazione tra le due tabelle e il database che funziona in questo modo si chiama database relazionale.

La banalità del contenuto dell'informazione da elaborare e la scarsità dei dati necessari per farlo, in questo esempio, non rende giustizia della enorme efficienza del database relazionale rispetto al database monotabellare gestibile con un foglio di calcolo.

La finalità dell'esempio era tuttavia quella di far capire la differenza che c'è tra le due impostazioni. Per la comprensione dell'efficienza rimando a quanto si scoprirà leggendo i prossimi paragrafi e seguendo le esercitazioni proposte.

SQLite serve per creare e gestire database relazionali.

## Indice

<b>1</b>	<b>Caratteristiche</b>	<b>2</b>
<b>2</b>	<b>Installazione</b>	<b>3</b>
<b>3</b>	<b>Uso di SQLite a riga di comando</b>	<b>3</b>
<b>4</b>	<b>Interfacce grafiche per SQLite</b>	<b>11</b>
<b>5</b>	<b>Uso di SQLite con LibreOffice Base</b>	<b>13</b>
<b>6</b>	<b>Collegamenti attraverso i principali linguaggi di programmazione.</b>	<b>27</b>

## 1 Caratteristiche

Il source code di SQLite è di pubblico dominio e può essere usato senza bisogno di licenza alcuna.

Esiste comunque un'organizzazione che se ne occupa, che rilascia licenze e assistenza a pagamento a chi utilizzi il software in costosi progetti complessi e che non voglia fondare questi progetti su un pubblico dominio che oggi c'è e domani non si sa. L'organizzazione si chiama Hwaci (pronuncia uàci) e, nonostante il nome, non è giapponese: sta per Hipp, Wyrick & Co., Inc. ed ha sede a Charlotte nel North Carolina. Hipp altro non è che Richard Hipp, l'inventore del software, la cui prima stesura risale al 2000, e Wyrick altro non è che la di lui moglie Ginger Wyrick.

Tutto in famiglia per una minuscola libreria software il cui kernel, scritto in linguaggio C, occupa poche centinaia di kB e che, con tutti gli accessori del caso, non occupa più di 5 mega sul nostro computer.

E questa piccola libreria si è rivelata il più efficiente motore di database che esista ed è diventato il database più diffuso al mondo.

Il limite principale lo abbiamo visto in premessa: non è un database client/server. Questo, tuttavia, può risultare un vantaggio, più che un limite. L'altro limite sta nel non riuscire a gestire database superiori a 2 TB (cioè 2 mila miliardi di byte), che penso, tuttavia, siano più che sufficienti per le esigenze di moltissima gente e di moltissime imprese.

I vantaggi vanno dal non essere un database client/server quando questa impostazione non serve e sarebbe ridondante per le nostre esigenze, al riunire in un unico file tutto il database, qualunque sia il numero di tabelle di cui è composto<sup>1</sup>.

SQLite è dotato di un suo linguaggio specifico per l'esecuzione di determinate operazioni e, fatta eccezione per alcuni comandi di raro utilizzo, riconosce i comandi del linguaggio SQL, che è lo standard per operare con i database (Structured Query Language).

## 2 Installazione

La versione corrente di SQLite si chiama **sqlite3**.

SQLite praticamente funziona su tutti i sistemi operativi; limitandomi a quelli più vicini a noi ricordo Linux, Windows, Mac OSX, Android, iOS e VSIX.

Per quanto riguarda apparecchiature equipaggiate Android non abbiamo nemmeno il disturbo di installarlo in quanto è già presente e così dovrebbe essere per tablet e smartphone equipaggiati con altri sistemi (iOS per quanto riguarda i prodotti Apple e VSIX per quanto riguarda Windows Phone).

Anche sui computer dovrebbe essere già installato insieme al sistema operativo. Per averne conferma basta lanciare il comando `sqlite3` e vedere cosa succede.

In ogni modo, dal sito <https://sqlite.org/> possiamo scaricare la versione adatta al nostro sistema operativo: l'ultima nata è la 3.15.1, rilasciata il 4 novembre appena scorso. Se usiamo Linux troviamo certamente la versione più adatta alla distro che abbiamo installato nel relativo repository.

Sullo stesso sito troviamo tutta la documentazione su SQLite, fatta molto bene ma solo in lingua inglese.

## 3 Uso di SQLite a riga di comando

Una volta installato, **sqlite3**, da solo, ci consente di creare e gestire un database senza bisogno di ricorrere ad altri programmi.

Essendo, come ho detto nel precedente paragrafo 1, un piccolissimo programma senza fronzoli e ridotto all'essenziale, se vogliamo usarlo da solo dobbiamo fare il sacrificio di scendere al suo basso livello e richiederne le prestazioni usando il linguaggio suo proprio attraverso il primordiale sistema dell'interfaccia a riga di comando.

---

<sup>1</sup>Occorre riconoscere che questi vantaggi li ritroviamo anche nel programma di gestione di database Access della Microsoft. Ma il paragone con Access è perdente, per Access, su tutti gli altri fronti: dal costo alla flessibilità di utilizzo.

L'interfaccia a riga di comando si basa su una interazione di tipo testuale tra utente ed elaboratore, che avviene attraverso l'uso della sola tastiera del computer per impartire istruzioni a questo, come si faceva un tempo, quando si dialogava con grossi mainframe, magari collocati in un luogo diverso da quello in cui lavorava l'utente, con apparecchiature prive di capacità elaborativa, chiamate terminali. Non a caso le interfacce a riga di comando che sono rimaste nei moderni computer si chiamano, pure loro, almeno per chi proviene dalla cultura informatica di quei tempi, terminali. E i programmi che ci propongono questo tipo di interfaccia sui moderni personal computer sono degli emulatori di terminale: in Linux e in Mac OS X, data la provenienza di entrambi i sistemi operativi dal sistema Unix, si chiamano **terminale** e ci vengono proposti nei menu delle applicazioni e in Windows si chiama **prompt dei comandi** e, ove non proposto nel menu, si può attivare con il comando cmd.

La seguente figura 1 mostra il terminale del computer su cui sto lavorando, equipaggiato Linux

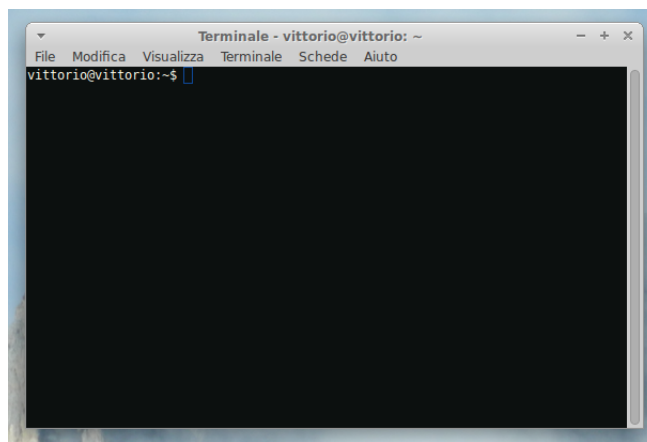


Figura 1: Terminale di un sistema Linux Ubuntu

Per avviare il programma SQLite dobbiamo scrivere nella prima riga del terminale, appena dopo il simbolo \$ che vediamo nella figura, il comando

sqlite3

e nella seguente figura 2 vediamo il risultato.

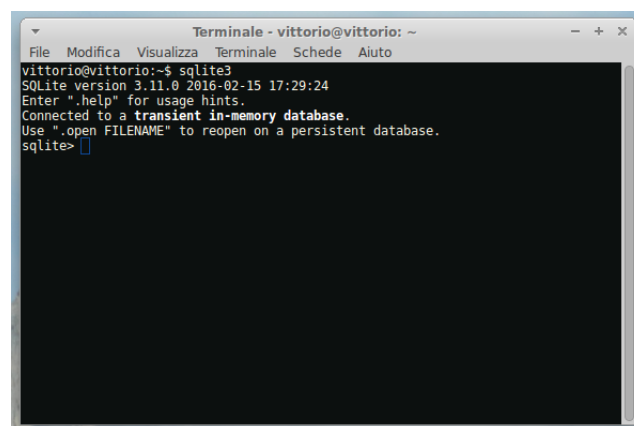


Figura 2: Prompt di sqlite3 in un terminale Linux Ubuntu

Nel terminale è comparso il prompt per i comandi SQLite, con cui veniamo avvertiti che SQLite è pronto (prompt) per ricevere comandi e scrivendo sulla riga che comincia con

sqlite>

possiamo chiedere a SQLite di fare ciò che ci serve.

Per chiederlo dobbiamo utilizzare un linguaggio che SQLite comprenda: il linguaggio SQL (Structured Query Language), quello che tutti i programmi di gestione di database comprendono, integrato con alcuni comandi speciali tipici del programma SQLite, detti comandi puntati, in quanto vengono impartiti preceduti da un punto.

L'inserimento dei comandi può indifferentemente avvenire con lettere maiuscole o minuscole. I comandi si possono scrivere su più righe, andando a capo con INVIO, e terminano con il punto e virgola ;.

I comandi SQL riconosciuti da SQLite non sono proprio tutti ma i non riconosciuti sono quelli che praticamente non si usano mai<sup>2</sup>. L'elenco che segue riguarda i comandi di uso più frequente e ne indico la sintassi per le più comuni applicazioni. Per ulteriori approfondimenti su questi e altri comandi rimando a uno dei numerosi manuali sul linguaggio SQL che si trovano in libreria o a ricerca su Internet.

CREATE TABLE, per creare una tabella

```
CREATE TABLE <nome_tabella> (<nome_colonna><tipo_dato>[attributi],  
                               <nome_colonna><tipo_dato>[attributi], ...);
```

gli attributi sono di inserimento facoltativo e i più frequentemente usati sono:

PRIMARY KEY, che rende il campo la chiave primaria della tabella (quella che serve per instaurare relazioni con altre tabelle);

NOT NULL, che rende il campo obbligatoriamente destinato a contenere un dato, contrariamente a quella che sarebbe l'impostazione di default;

AUTOINCREMENT, destinato solo a campi contenenti un numero per fare in modo che il contenuto del campo si incrementi automaticamente di una unità a partire dal più grande valore già presente<sup>3</sup>. Attributo molto utile per la primary key.

DROP TABLE, per cancellare una tabella

```
DROP TABLE <nome_tabella>;
```

INSERT, per inserire dati in una tabella

```
INSERT INTO <nome_tabella> (<colonna_interessata>, <colonna_interessata>, ...) VALUES  
    (<valore>, <valore>, ...);
```

Ovviamente i valori vanno elencati nello stesso ordine delle colonne. I valori numerici si inseriscono tali e quali, mentre i valori espressi come testo vanno racchiusi tra semplici apici '.

SELECT, per leggere i dati dalle tabelle del database

```
SELECT <nome_colonna>, <nome_colonna>, ... FROM <tabella>, <tabella>, ... WHERE  
    <condizioni>;
```

---

<sup>2</sup>Le funzionalità SQL non implementate in SQLite riguardano i comandi RIGHT e FULL OUTER JOIN (LEFT OUTER JOIN è implementato), GRANT e REVOKE. Non è implementata la funzione di scrittura nelle VIEWS (le VIEWS di SQLite sono in sola lettura). Per il comando ALTER TABLE sono implementate solo le funzioni RENAME TABLE e ADD COLUMN.

<sup>3</sup>Per chi conosce il linguaggio SQL chiarisco che, in luogo della scrittura AUTO\_INCREMENT tipica di quel linguaggio, nell'uso con SQLite si deve usare la scrittura AUTOINCREMENT.

Questo è il comando più importante del linguaggio SQL in quanto è l'unico che viene o può convenientemente essere utilizzato anche quando lavoriamo con SQLite non a riga di comando. Per meglio fissare i concetti sulla sua sintassi sottolineo che

- dopo SELECT va inserito, attraverso il nome di colonna, l'elenco dei campi che si vogliono evidenziare nel risultato della ricerca (in gergo la query), separati da virgola e nell'ordine in cui si vogliono evidenziare; non appena vi è il minimo rischio di ambiguità e, comunque, nel caso i campi da evidenziare non corrispondano, nell'ordine o nel numero, a quelli che sono coinvolti nella ricerca, il campo va indicato non semplicemente come nome\_colonna ma nella forma più completa tabella.nome\_colonna;
- dopo FROM va inserito l'elenco delle tabelle coinvolte nella ricerca, coinvolte o per l'estrazione di campi da evidenziare o semplicemente perché contengono almeno un elemento utile alla ricerca;
- dopo WHERE, separate eventualmente da AND, si indicano le relazioni tra le tabelle coinvolte nella ricerca nella forma tabella.nome\_colonna = tabella.nome\_colonna e poi, sempre con AND, si aggiungono le condizioni di ricerca
  - nella forma tabella.nome\_colonna like '.....' se abbiamo a che fare con dati espressi in forma di testo. Per esprimere la condizione like si possono usare i caratteri jolly che, nel linguaggio SQL sono:
    - \* \_ per un solo carattere,
    - \* % per un numero imprecisato di caratteri;
  - nella forma tabella.nome\_colonna = oppure > oppure < a un numero se abbiamo a che fare con dati numerici;
  - nella forma tabella.nome\_colonna is <valore> se vogliamo esprimere un criterio di perfetta identità; <valore> può essere una stringa di testo (espressa tra apici), un numero o il valore null, scritto tale e quale senza apici oppure, sempre scritto senza apici, uno dei valori booleani true o false.

Sembra tutto molto complicato ma, dopo un po' di pratica, ci accorgiamo che non è così: alla fine del paragrafo proporrò un piccolo esempio che spero incoraggi in tal senso.

Nel comando per creare una tabella abbiamo visto che occorre indicare, per ciascuna colonna destinata a scandire i campi dei record del nostro database, il tipo del dato che vi può essere inserito. In SQLite i tipi di dato gestiti sono solo cinque:

**integer**, cioè numero intero

**real**, cioè numero reale

**text**, cioè testo

**blob**, cioè testo in forma binaria

**null**, cioè sconosciuto.

Dal momento che in altri contesti, per esempio nel motore di database MySQL, i tipi sono molti di più, per rendere possibile l'importazione di dati da un database diverso da SQLite, sqlite3 riconosce delle affinità per incanalare sui propri tipi quelli di altri. Così verranno ascritti al tipo integer tutti i tipi altrove chiamati con un qualsiasi nome contenente la stringa int; verranno ascritti al tipo real tutti i tipi altrove chiamati con nomi contenenti la stringa floa o doub; verranno ascritti al tipo text tutti i tipi altrove chiamati con nomi contenenti la stringa char o clob.

Ci rimane da vedere quali sono i comandi tipici di `sqlite3`, i così detti comandi puntati.

Innanzitutto un chiarimento sul comando `sqlite3` che utilizziamo per entrare nel prompt di `sqlite3`. Se lo scriviamo da solo apriamo la finestra terminale che ho riprodotto nella precedente figura 2, con il prompt. Se lo scriviamo seguito dal nome di un database, se il database esiste verrà aperto insieme al prompt, se il database non esiste verrà creato non appena avremo creato almeno una tabella. Tutto ciò sapendo che `sqlite3` trova e memorizza i database nella directory dalla quale diamo il comando `sqlite3`. Pertanto, se vogliamo aprire un database esistente dobbiamo preventivamente recarci nella directory dove esso è memorizzato.

Una volta entrati nel prompt di `sqlite3`, con il comando puntato `.help` possiamo vedere l'elenco di tutti i comandi puntati disponibili con la descrizione di ciò che fanno.

Qui riporto quelli di uso più comune.

`.exit` oppure `.quit` per uscire dal prompt di `sqlite3`

`.schema` [`<tabella>`] mostra lo statement SQL di creazione di tutto il database o della tabella, se indicata

`.import` `<file>` `<tabella>` importa i dati del file indicato nella tabella indicata. Occorre preventivamente indicare a `sqlite3` il simbolo separatore che si trova nel file di dati con il comando `.separator` '`<simbolo>`': il simbolo potrà essere la virgola `,`, il punto e virgola `;`, il carattere `|`, ecc.

`.mode` `<modo>` imposta il formato di output.

<code>csv</code>	valori separati da virgola
<code>column</code>	valori in colonna allineati a sinistra
<code>line</code>	un valore per riga
<code>tabs</code>	valori separati da uno spazio di tabulazione
<code>tcl</code>	valori come lista di elementi TCL
<code>list</code>	valori separati dal simbolo scelto con <code>.separator</code>

`<modo>` sta per

`.output` `<file>` reindirige l'output sul file indicato, creandolo se non c'è

`.output stdout` reindirige l'output allo schermo (impostazione di default)

`.dump` salva il database in formato SQL. Per salvare in un file occorre preventivamente reindirizzare l'output con il comando `.output <file>`. Se si fa seguire al comando `.dump` il nome di una tabella, verrà salvata solo la tabella.

`.read` `<file>` ripristina il database leggendo il file in formato SQL creato con il comando `.dump`.

## Un esercizio istruttivo

Supponiamo di avere una biblioteca di quattro libri e, anche se totalmente inutile per padroneggiarla ma molto utile come esercizio in questa sede, vogliamo costruire un database per gestirla.

I libri sono

I promessi sposi di Alessandro Manzoni

Un luogo chiamato libertà di Ken Follett

I pilastri della terra di Ken Follett

Furore di John Steinbeck.

Il primo passo, sicuramente il più importante, è la progettazione del database e, per affrontarlo, dobbiamo innanzi tutto chiederci a cosa vogliamo che serva questo database. Nel nostro caso supponiamo che le esigenze siano: sapere le opere che abbiamo dei vari autori, sapere le opere che abbiamo scritte in certi periodi, sapere gli autori che abbiamo rappresentati nella nostra biblioteca per Paesi di appartenenza, sapere quali opere abbiamo di autori di un certo Paese.

Da qui deriviamo che i dati che dovremo inserire nel database saranno: i nomi degli autori, i titoli delle opere, le date di nascita e di morte degli autori, i Paesi di appartenenza degli autori.

Dal momento che la regola d'oro del buon database è «scrivere le cose una ed una sola volta», sia pure per colpa del solo Ken Follett, di cui abbiamo due libri, non possiamo ricorrere ad una tabella gestita in un foglio di calcolo in quanto dovremmo scrivere due volte il nome di Ken Follett, due volte la sua data di nascita, due volte il nome del Paese di appartenenza. Dobbiamo allora ricorrere ad un database relazionale e SQLite è qui per questo.

Una organizzazione ottimale del database che assolva alle finalità che ci siamo posti potrebbe essere quella di impostare il database su tre tabelle: una per gli autori, contenente le date di nascita e di morte, una per i Paesi di appartenenza e una per i titoli delle opere. La tabella degli autori si dovrà relazionare con quella dei Paesi e con quella dei titoli delle opere e quest'ultima basterà che si relazioni con quella degli autori, attraverso la quale finirà per essere relazionata anche con la tabella dei Paesi.

La tabella autori dovrà pertanto avere cinque colonne: idAutore, che assolverà la funzione di chiave primaria e conterrà il numero di riferimento corrispondente all'autore, nome, che conterrà il nome dell'autore, nascita, che conterrà la data di nascita dell'autore, morte, che conterrà la data di morte dell'autore e idPaese, che conterrà il numero di riferimento del Paese derivante dalla tabella dei Paesi.

La tabella paesi avrà due colonne: idPaese, che assolverà la funzione di chiave primaria e conterrà il numero di riferimento corrispondente al Paese e paese, contenente il nome del Paese.

La tabella titoli avrà tre colonne: idTitolo, che assolverà la funzione di chiave primaria e conterrà il numero di riferimento corrispondente al titolo di ciascuna opera, titolo, che conterrà il titolo dell'opera e idAutore, che conterrà il numero di riferimento corrispondente all'autore derivante dalla tabella autori.

La progettazione è finita ed ora dobbiamo passare alla realizzazione.

Apriamo un terminale (se non troviamo altrimenti il modo, in Windows, lanciando il comando cmd), se vogliamo ci portiamo in una directory comoda per l'archiviazione del nostro database (in tutti i sistemi operativi, per fare questo, si scrive il comando cd seguito dall'indirizzo della directory in cui vogliamo spostarci) e, volendo dare al nostro database il nome «biblioteca», scriviamo il comando

```
sqlite3 biblioteca
```

aprendo così il prompt di sqlite3, che apparirà come in figura 2, già pronto anche per memorizzare il database che costruiremo in un file chiamato «biblioteca». Da qui in poi scriveremo tutti i nostri comandi nella prima riga libera aperta che inizia con il prompt sqlite>.

Ora creiamo la prima delle tre tabelle che ci servono, che chiamiamo «autori», con il comando

```
create table autori (idAutore integer primary key autoincrement, nome text, nascita integer, morte integer, idPaese integer);
```

Poi creiamo la tabella dei Paesi, che chiamiamo «paesi», con il comando:

```
create table paesi (idPaese integer primary key autoincrement, paese text);
```



Infine creiamo la tabella delle opere, che chiamiamo «titoli», con il comando `create table titoli (idTitolo integer primary key autoincrement, titolo text, idAutore integer)`; sempre non dimenticando di chiudere i comandi con il punto e virgola.

Se facciamo qualche errore o scriviamo parole che `sqlite3` non comprende sarà sua cura avvisarci; se vogliamo comunque verificare che tutto sia andato bene possiamo scrivere il comando puntato `.schema` e nel terminale vedremo elencati i nostri tre comandi, a conferma che `sqlite3` li ha capiti ed acquisiti. Da questo momento, nella directory in cui stiamo lavorando, compare il file «biblioteca».

Non ci resta che inserire i dati. Questa è la cosa più brutta da fare lavorando da terminale e nei paragrafi successivi vedremo sistemi più comodi per farlo. Ma questo esercizio serve per capire come lavora `sqlite3` ed ha la sua utilità.

Cominciamo ad inserire i Paesi, visto che la chiave del Paese (`idPaese`) ci servirà per alimentare la tabella degli autori, e lo facciamo con i comandi

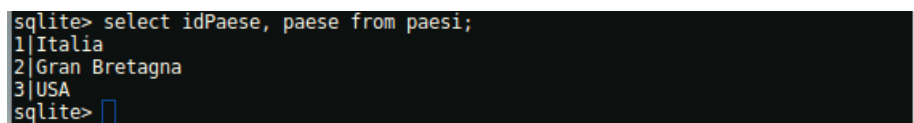
```
insert into paesi (paese) values ('Italia');  
insert into paesi (paese) values ('Gran Bretagna');  
insert into paesi (paese) values ('USA');
```

Abbiamo inserito i paesi, rispettivamente, di Alessandro Manzoni, di Ken Follett e di John Steinbeck, senza inserire il dato della prima colonna, `idPaese`, che avevamo definito autoincrementale.

Se scriviamo il comando

```
select idPaese, paese from paesi;
```

chiedendo a `sqlite3` di selezionare i dati contenuti nelle colonne `idPaese` e `paese` della tabella `paesi`, otteniamo la risposta



```
sqlite> select idPaese, paese from paesi;  
1|Italia  
2|Gran Bretagna  
3|USA  
sqlite> █
```

a dimostrazione del fatto che `sqlite3` ha attribuito automaticamente i valori 1, 2 e 3, nell'ordine, a Italia, Gran Bretagna e USA.

Inseriamo ora gli autori, con i comandi

```
insert into autori (nome, nascita, morte, idPaese) values ('Manzoni, Alessandro', 1785, 1873, 1);
```

```
insert into autori (nome, nascita, morte, idPaese) values ('Follett, Ken', 1949, null, 2);
```

```
insert into autori (nome, nascita, morte, idPaese) values ('Steinbeck, John', 1902, 1968, 3);
```

sempre trascurando l'inserimento del dato nella prima colonna, quella dell'`idAutore`, in quanto la fa `sqlite3`, dando automaticamente l'identificativo 1 a Manzoni, 2 a Follett e 3 a Steinbeck, nell'ordine con cui li abbiamo inseriti. Dal momento che Ken Follett è vivente, in luogo della data di morte abbiamo inserito il valore `null`. Nella colonna `idPaese` abbiamo inserito gli identificativi di Paese desunti dalla relativa tabella. Notare che, mentre i valori numerici si inseriscono tali e quali, i testi (stringhe) vanno racchiusi tra apici.

Finalmente tocca alle opere, che inseriamo con i comandi

```
insert into titoli (titolo, idAutore) values ('I promessi sposi', 1);
```

```
insert into titoli (titolo, idAutore) values ('Un luogo chiamato libertà', 2);
```

```
insert into titoli (titolo, idAutore) values ('I pilastri della terra', 2);
```

```
insert into titoli (titolo, idAutore) values ('Furore', 3);
```

ed abbiamo così creato il database della nostra biblioteca.

A questo punto vediamo come possiamo utilizzarlo per avere alcune informazioni sulla nostra raccolta letteraria.

Per conoscere le opere scritte da Ken Follett, in questo spezzone di schermata di terminale

```
sqlite> select titolo from titoli, autori where titoli.idAutore = autori.idAutore and autori.nome like 'Follett%';
Un luogo chiamato libertà
I pilastri della terra
```

vediamo la query e il risultato. Nella query, in linguaggio SQL, chiediamo di selezionare il titolo delle opere ricorrendo alle tabelle titoli e autori, relazionate tra loro dall'identificativo dell'autore presente in entrambe e in corrispondenza di un autore il cui nome comincia con Follett. Puntualmente sqlite3 ci restituisce l'elenco delle opere di Ken Follett che abbiamo in biblioteca.

Per conoscere le opere scritte da autori americani, in questo spezzone di schermata di terminale

```
sqlite> select nome, titolo from autori, titoli, paesi where titoli.idAutore = autori.idAutore and autori.idPaese = paesi.idPaese and paesi.paese like 'USA';
Steinbeck, John|Furore
```

abbiamo query e risultato. Nella query, in linguaggio SQL, chiediamo di selezionare autore e titolo delle opere, ricorrendo alle tabelle autori, titoli e paesi, autori e titoli relazionate tra loro dall'identificativo dell'autore presente in entrambe e autori e paesi relazionate tra loro dall'identificativo del Paese presente in entrambe, in corrispondenza di un paese che si chiama USA. E sqlite3 ci indica l'unica opera scritta da un autore americano che abbiamo in biblioteca, come richiesto, con il nome dell'autore e il titolo.

Nel caso che segue abbiamo chiesto autore e titolo di opere scritte da autori nati prima del 1900

```
sqlite> select nome, titolo from autori, titoli where autori.idAutore = titoli.idAutore and autori.nascita < 1900;
Manzoni, Alessandro|I promessi sposi
```

ed abbiamo puntualmente ottenuto la risposta.

Ora abbiamo voluto conoscere autore e titolo di opere scritte da autori viventi

```
sqlite> select nome, titolo from autori, titoli where autori.idAutore = titoli.idAutore and autori.morte is null;
Follett, Ken|Un luogo chiamato libertà
Follett, Ken|I pilastri della terra
```

ed otteniamo l'elenco delle opere di Follett, l'unico vivente che abbiamo in biblioteca.

Finora abbiamo ottenuto risposte sullo schermo.

Se volessimo ottenere risposte su supporto meno volatile, o per conservarle o per stamparle, potremmo far precedere al comando select il comando .output <nome\_file> come fatto in questo esempio

```
sqlite> .output opere_di_Ken_Follett
sqlite> select titolo from titoli, autori where titoli.idAutore = autori.idAutore and autori.nome like 'Follett%';
sqlite> █
```

dove vediamo che sullo schermo non compare alcuna risposta ma, se andiamo a vedere nella directory in cui stiamo lavorando, troviamo un file di testo denominato, come richiesto, «opere\_di\_Ken\_Follett», con l'elenco delle opere di Ken Follett che abbiamo in biblioteca.

Attenzione che il comando `.output <nome_file>` rimane valido per tutte le query che facciamo nel seguito della corrente sessione di lavoro e tutti i risultati che otterremo andranno ad accodarsi al testo contenuto nel file `<nome_file>`, fino a quando, con un altro comando `.output <altro_nome_file>` reindirizzeremo su un altro file i risultati delle nostre ricerche, o, con un comando `.output stdout` li faremo ritornare sullo schermo.

## 4 Interfacce grafiche per SQLite

Nel paragrafo precedente abbiamo esaminato il modo più ruspante di usare SQLite, quello che forse preferiscono gli smanettoni. Da mezzo smanettone quale sono, pur apprezzando la semplicità e la scorrevolezza dell'uso della riga di comando per impostare le tabelle e le query, devo riconoscere che l'inserimento dei dati dalla riga di comando non è il massimo, soprattutto se la biblioteca contiene alcune migliaia di libri e non quattro, come nell'esercizio appena svolto.

Il mondo del software libero ci propone qualche soluzione semplificante, con piccoli programmi abbastanza leggeri e semplici da usare che ci danno modo di utilizzare SQLite senza conoscere i comandi puntati e riducendo la necessità di conoscere il linguaggio SQL al solo comando `select`.

La scelta del programma che presento in questa sede è stata ispirata dal fatto che si tratta di un programma che si comporta egregiamente su tutti i sistemi operativi Linux, Mac OS X e Windows, che ha un'interfaccia in italiano (pur avendo, purtroppo, l'help in inglese) e che, a parte innumerevoli funzionalità anche per me ancora misteriose, è semplicissimo da usare: si chiama **Sqliteman**. Il suo difetto è di essere un po' antiquato e di non avere avuto aggiornamenti da alcuni anni.

Girovagando su Internet possiamo trovare altre cose dello stesso tipo: da **Sqlitebrowser**, pure disponibile per tutti i tre sistemi operativi, a **SQLiteStudio**, disponibile solo per Linux.

Per Android molto bella la app **aSQLiteManager**, che troviamo su Google Play e che ha un funzionamento molto simile a Sqliteman che adesso vedremo.

Sqliteman possiamo scaricarlo dal sito <http://sqliteman.yarpen.cz/>. Chi usa Linux lo può installare dal repository della distro.

La figura 3 alla pagina seguente mostra la schermata di lavoro di Sqliteman.

Vi è caricato il database che abbiamo creato nell'esercizio svolto nel paragrafo precedente e, in particolare, è visibile la tabella autori di quel database.

La finestra verticale a sinistra è dedicata alla navigazione nelle tabelle del database che avviene in modo assolutamente intuitivo. Con doppio click sul nome della tabella rendiamo visibile la tabella stessa nella finestra centrale sulla destra.

La finestra sulla destra in alto è dedicata all'inserimento dei comandi in linguaggio SQL, se desideriamo usarli o se dobbiamo usarli, come nel caso delle ricerche.

Nella finestra sulla destra in basso troviamo i messaggi che SQLite ritiene di inviarci, tipo quelli di OK se tutto va bene o di errore se qualche cosa va male.

Una passeggiata tra le voci del menu ci fa capire immediatamente cosa possiamo fare con questo programma (affinché le voci del menu siano tutte apribili è necessario sia caricato un database) e il come farlo è assolutamente intuitivo.

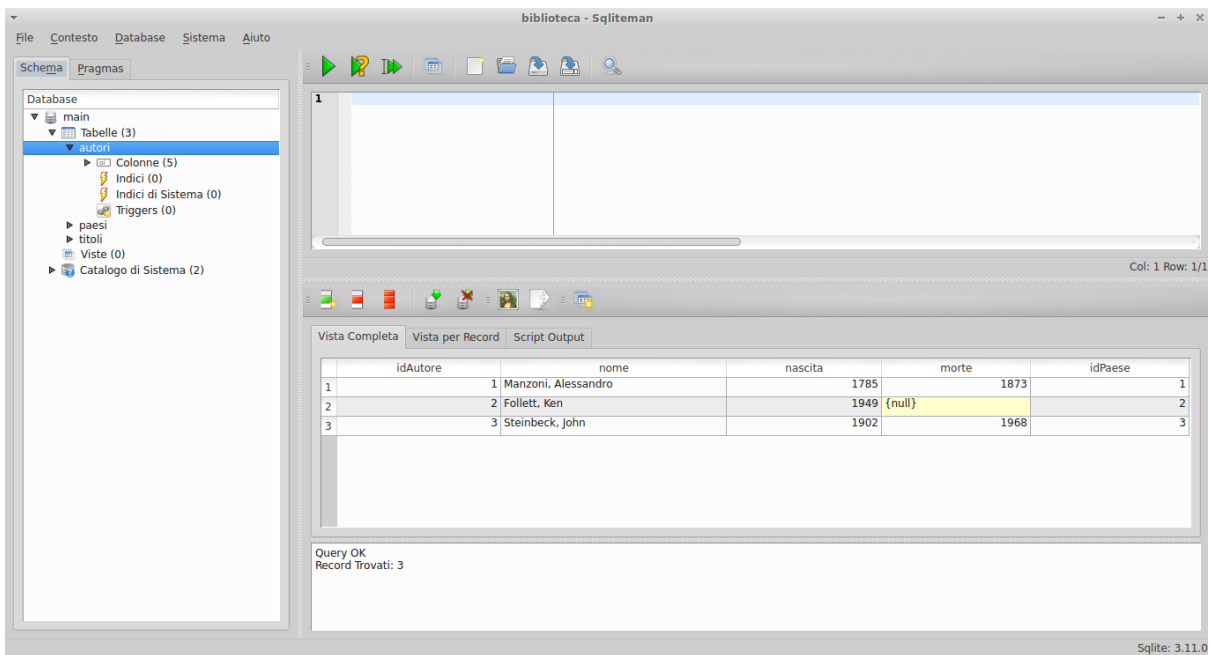


Figura 3: Schermata di Sqliteman

Per creare un nuovo database scegliamo FILE ▷ NUOVO e indichiamo dove salvare e come nominare il database; per creare le tabelle scegliamo DATABASE ▷ CREA TABELLA. Dopo aver letto il precedente paragrafo non dovremmo avere alcun dubbio su come proseguire.

L'inserimento dei dati può avvenire in due modi:

- utilizzando i comandi in linguaggio SQL scrivendoli nella finestra sulla destra in alto e attivandoli cliccando, man mano per ognuno di essi, sul pulsante (possiamo anche scrivere più comandi e attivarli in blocco, posizionandoci sul primo da inserire e cliccando sul pulsante );
- lavorando nella finestra centrale di destra, previa apertura della tabella in cui vogliamo inserire i dati. L'inserimento comincia cliccando sul pulsante (nuova riga): ciò produce una nuova riga vuota nella tabella e, con doppio click sulle varie zone corrispondenti alle varie colonne, apriamo una finestrella di input in cui scrivere il dato da inserire. Una volta scritti tutti i dati per tutte le nuove righe diamo il via all'inserimento nel database cliccando sul pulsante .

Nel primo caso abbiamo tutta la scomodità che avevamo riscontrato nell'inserimento da riga di comando visto nel paragrafo precedente. Nel secondo è un po' meglio, ma non troppo.

Per quanto riguarda le ricerche si deve ricorrere al linguaggio SQL, scrivendo le query nella finestra in alto a destra e eseguendole con click sul pulsante . Il risultato compare nella finestra centrale di destra. Se scegliamo dal menu DATABASE ▷ CREA QUERY ci troviamo proposto un sistema alternativo praticamente inservibile se la ricerca coinvolge più di una tabella.

In poche parole, questi programmini che forniscono una interfaccia grafica per lavorare con SQLite danno un tocco di modernità all'approccio ma non è che semplificano gran che.

## 5 Uso di SQLite con LibreOffice Base

Un modo egregio che ci offre il mondo del software libero per lavorare comodamente con SQLite è di farlo attraverso il modulo Base di LibreOffice (o delle un po' meno libere edizioni OpenOffice). In questo modo, una volta creato il database e impostate le sue tabelle con il software sqlite3 (direttamente da terminale secondo quanto abbiamo visto nel paragrafo 3 o ricorrendo ad una interfaccia grafica secondo quanto abbiamo visto nel paragrafo 4) lo colleghiamo al modulo Base e, da qui in poi, alimentazione del database, ricerche e quant'altro lo facciamo con tutte le potenti funzionalità che ci offre Base: a partire dalla possibilità di alimentare il database con comode maschere di inserimento fino a quella di produrre risultati di ricerche ben impaginati e inseribili in relazioni dall'aspetto professionale.

Ci si può chiedere: ma, dal momento che il modulo Base può lui stesso creare un database, che bisogno c'è di dargli in pasto un database di altra natura come SQLite?

La risposta è semplice: il database creato da Base è utilizzabile solo con Base mentre il database creato con sqlite3 è utilizzabile ovunque, vuoi secondo quanto abbiamo visto nei paragrafi 2 e 3, vuoi secondo quanto vedremo nel prossimo paragrafo. Basta copiare il piccolo file che lo contiene aggiornato nei luoghi dove lo vogliamo utilizzare: altra directory del computer, chiavetta trasportabile, tablet o smartphone (che abbiamo visto essere naturalmente dotati di sqlite3).

L'ottenimento di tutti questi benefici costa ovviamente qualche fatica, che ritengo comunque sia ben compensata dai risultati: prima fra tutte la fatica di predisporre il computer ad instaurare una connessione tra database e altre applicazioni.

### Open DataBase Connectivity (ODBC)

Il sistema ODBC, almeno nel caso di SQLite, ritengo sia relativamente il più semplice ed il più stabile per creare una connettività di database con altre applicazioni: l'alternativa sarebbe la Java DataBase Connectivity (JDBC) che mi sembra più complicata e ritengo più adatta, eventualmente, per database MySQL.

Per la creazione di un contesto di connettività i problemi da superare sono due:

- sul computer devono essere installati i driver per il collegamento ad un certo tipo di database, nel nostro caso SQLite,
- il computer deve darci la possibilità di creare ed evidenziare l'origine dei dati (Data Source Name, DSN) per instaurare il collegamento.

### Sistema operativo Linux Ubuntu e derivate, compreso Linux Mint

Se abbiamo la fortuna di usare uno di questi sistemi operativi entrambi i problemi si risolvono impartendo da terminale, con collegamento Internet attivo, il seguente comando `sudo apt-get install libsqliteodbc unixodbc`, nelle versioni più recenti semplificabile in `sudo apt install libsqliteodbc unixodbc`.

Con questo comando installiamo il driver (package **libsqliteodbc**) e instauriamo la possibilità di indicare i DSN. Ad installazione avvenuta, grazie al package **unixodbc**, nella directory `/etc` troviamo i due file

`odbcinst.ini`,

`odbc.ini`,

che sono la chiave di tutto.

Entrambi in formato testo, il primo (odbcinst.ini) ha il seguente contenuto derivante dal package libsqliteodbc e che teniamo tale e quale:

```
[SQLite]
Description=SQLite ODBC Driver
Driver=libsqliteodbc.so
Setup=libsqliteodbc.so
UsageCount=2
```

```
[SQLite3]
Description=SQLite3 ODBC Driver
Driver=libsqlite3odbc.so
Setup=libsqlite3odbc.so
UsageCount=2
```

ove si dà conto dei driver installati.

Nel secondo (odbc.ini) dobbiamo inserire i nomi e gli indirizzi dei database che desideriamo diventino delle origini dati (DSN), nel caso del nostro database «biblioteca» che io ho archiviato in una directory Database della mia home:

```
[biblioteca]
Driver = SQLite3
Database = /home/vittorio/Database/biblioteca
```

In questo modo abbiamo creato l'origine dati [biblioteca] abbinando ad essa il driver SQLite3 e l'indirizzo del database.

Per poter memorizzare quanto scriviamo ricordiamo di aprire come superuser il file nell'editor di testo che useremo. Se, per esempio, usiamo GEDIT dovremo aprire il file con il comando `sudo gedit /etc/odbc.ini`.

Quanto possiamo fare semplicemente con l'editor di testo potremmo anche farlo con interfaccia grafica installando il package **unixODBC-bin**. La ritengo tuttavia una strada inutilmente complicata e non la consiglio.

## Linux in generale

Quanto detto per Ubuntu e derivate vale ovviamente per tutti i vari sistemi Linux in circolazione, fatti salvi gli eventuali differenti comandi per installare i packages libsqliteodbc e unixodbc.

## Sistema operativo Mac OS X

All'indirizzo <http://www.ch-werner.de/sqliteodbc/> possiamo trovare i driver odbc per sqlite3 destinati al sistema operativo Mac OS X nel file **sqlite3-odbc-x.xx.dmg**, azionando il quale effettuiamo l'installazione.

Fratello di Linux per la comune derivazione da Unix, anche Mac OS X lavora con unixodbc, già installato insieme ad una utilità di interfaccia grafica **ODBCManager**, attraverso la quale possiamo creare i DSN.

Come in Linux, i file odbcinst.ini e odbc.ini, che si trovano nella directory `/Library/ODBC`, possono essere scritti e modificati con un editor di testo.

## Sistema operativo Windows

Sempre all'indirizzo <http://www.ch-werner.de/sqliteodbc/> troviamo i driver odbc per sqlite3 destinati al sistema operativo Windows nei file installatori **sqliteodbc.exe** e **sqliteodbc\_w64.exe**, rispettivamente per sistemi a 32 o a 64 bit.

Una volta installati i driver dobbiamo per forza ricorrere ad un'applicazione per nominare i DSN: in Windows il file odbc.ini non è così a portata di mano e di editor di testo come avviene per i sistemi Linux e OS X in quanto si trova in una sottochiave del Registro di Sistema che, in Windows, è il sancta sanctorum che è meglio il poco esperto non tocchi mai, pena il rischio di dover buttare tutto.

L'applicazione fa parte del sistema operativo e si chiama **Amministrazione origini dati ODBC**: se non la troviamo nei menu la possiamo far partire con il comando odbcad32 (l'eseguibile si chiama, infatti, odbcad32.exe). Funziona in maniera del tutto analoga all'ODBCManager del Mac.

Se, una volta lanciato, apriamo la scheda DRIVER vediamo l'elenco di tutti i driver che abbiamo installati e, se li abbiamo installati con uno degli eseguibili sopra indicati, l'elenco comprenderà SQLite ODBC Driver e SQLite3 ODBC Driver.

Per creare i DSN apriamo la scheda DSN UTENTE e clicchiamo sul pulsante AGGIUNGI.

Dall'elenco dei driver che compare nella finestra successiva selezioniamo SQLITE3 ODBC DRIVER e clicchiamo sul pulsante FINE.

Si apre così una nuova finestra intitolata SQLITE3 ODBC DSN CONFIGURATION. Nella prima zona di immissione, intitolata DATA SOURCE NAME, scriviamo il nome che vogliamo dare alla nostra sorgente dati (se vogliamo riferirci al database «biblioteca» del precedente esercizio scriviamo biblioteca). Nella successiva zona di immissione, intitolata DATABASE NAME, inseriamo l'indirizzo del database, aiutandoci con il pulsante BROWSE. Il resto possiamo fare a meno di inserirlo.

Con OK su questa finestra e sulla successiva usciamo dal programma.

## Colleghiamo il database

Lanciamo LibreOffice Base e, nella finestra che si apre, intitolata CREAZIONE GUIDATA DATABASE, scegliamo di procedere con COLLEGA A UN DATABASE ESISTENTE. Clicchiamo in fondo a destra della finestrella, scegliamo ODBC dall'elenco che ci viene proposto e clicchiamo su AVANTI.

Nella successiva finestra, inseriamo l'origine dati (il DSN) nella zona di inserimento intitolata NOME DELLA SORGENTE DATI ODBC NEL SISTEMA, andando a sceglierlo nell'elenco che compare cliccando sul pulsante SFOGLIA. Se abbiamo fatto bene tutto quanto previsto nel capitoletto precedente, l'elenco contiene il nome «biblioteca», lo selezioniamo, clicchiamo su OK in modo che il nome si veda nella zona di inserimento e clicchiamo su AVANTI.

Passiamo senza fare nulla oltre la prossima finestra, dedicata alla configurazione di un eventuale sistema di autenticazione, che qui non ci interessa, premendo su AVANTI.

Arriviamo così alla finestra finale dove ci viene proposto di registrare o meno il database in LibreOffice (consiglio di scegliere SÌ e dirò subito il perché). Possiamo anche scegliere se aprire subito il database o no. Cliccando su FINE creiamo un database di LibreOffice Base, che sarà un file con estensione .odb e che sceglieremo dove memorizzare e con quale nome nella successiva finestra (nel nostro caso consiglio il nome biblioteca nella stessa directory dove abbiamo il file sqlite3).

La registrazione del database in LibreOffice consente di collegare il database non solo al modulo Base ma anche agli altri moduli di LibreOffice, con utilità che vedre-

mo nel seguito. Se non lo facciamo subito abbiamo comunque modo di registrare il database anche in un momento successivo.

Se abbiamo operato come suggerito, ora troveremo memorizzati due file biblioteca: uno, senza estensione, è il file del database sqlite3 (alcuni suggeriscono di attribuirgli l'estensione .db per sapere di che cosa si tratta) e l'altro, con estensione .odb, è l'impalcatura che gli creiamo attorno con LibreOffice, impalcatura che conterrà tutte le belle cose che adesso andremo a fare. Importante sapere che i dati sono sempre e soltanto nel file del database, quello senza estensione o con estensione .db se vogliamo dargliela, e che i dati che vediamo usando Base sono quelli e li vediamo grazie alla connessione ODBC che lega Base al database SQLite.

## Una prima esplorazione

Con un doppio click sul file biblioteca.odb compare sullo schermo quanto vediamo nella seguente figura 4.

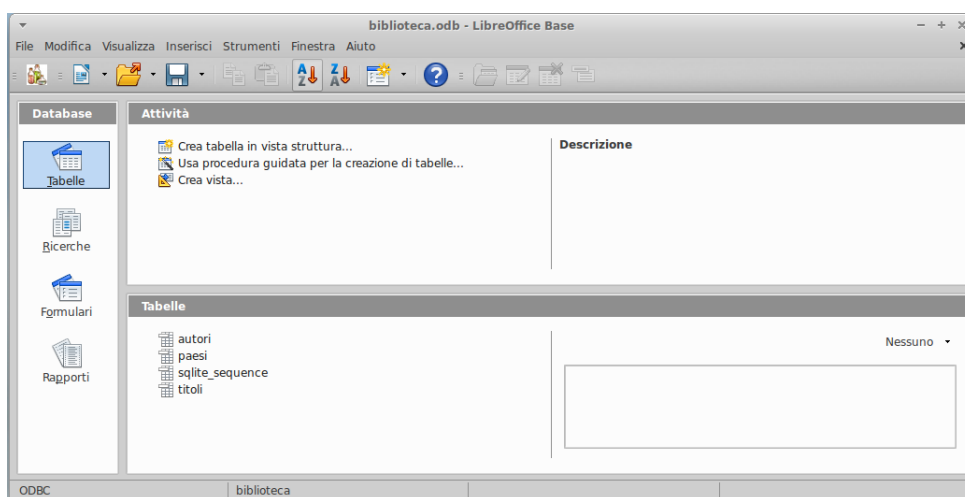
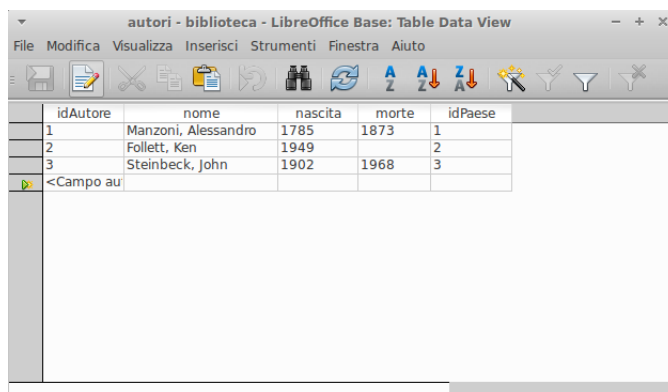


Figura 4: Schermata di LibreOffice Base

Essendo selezionata, nella zona verticale di sinistra DATABASE, l'icona TABELLE, nella finestra in basso a destra TABELLE, a parte la tabella di servizio sqlite\_sequence, abbiamo l'elenco delle tre tabelle del nostro database, creato nell'esercitazione proposta a fine Paragrafo 3.

Con doppio click sulla dicitura di tabella AUTORI apriamo la tabella che compare come nella seguente figura 5.

The screenshot shows the "autori - biblioteca - LibreOffice Base: Table Data View" window. It displays a table with the following data:

idAutore	nome	nascita	morte	idPaese
1	Manzoni, Alessandro	1785	1873	1
2	Follett, Ken	1949		2
3	Steinbeck, John	1902	1968	3

Figura 5: Visualizzazione di una tabella in LibreOffice Base



Se selezioniamo, nella zona verticale di sinistra DATABASE, una delle altre icone RICERCHE, FORMULARI, RAPPORTI non troviamo nulla, in quanto il nostro database di partenza non contiene altro che tabelle, e solo quelle ci ritroviamo. Per avere qualche cosa in queste zone del database dobbiamo costruircelo e vedremo che cosa e come nei prossimi capitoletti.

Rimanendo nella zona TABELLE, dobbiamo dire che la tabella che visualizziamo con doppio click sul suo nome si presta abbastanza comodamente all'inserimento di altri dati.

Supponiamo di aver acquistato altri due libri per la nostra biblioteca: Il codice Rebecca di Ken Follett e Caos calmo di Sandro Veronesi.

Per inserirli nel nostro database, visto che Ken Follett è già un autore catalogato, dobbiamo innanzi tutto inserire nella tabella degli autori soltanto Sandro Veronesi, Italiano, nato nel 1959 e vivente. Nella tabella visualizzata nella figura 5 ci posizioniamo con il puntatore del mouse nella prima casella libera della colonna NOME<sup>4</sup>, sulla destra della casella dedicata al campo automatico e scriviamo Veronesi, Sandro. Ci spostiamo nella casella successiva, nella colonna NASCITA e scriviamo 1959. Non scriviamo nulla nella successiva casella, in quanto Sandro Veronesi è vivente in ottima salute (non scrivendo nulla, nel database viene inserito il valore null). Infine, nella casella successiva della colonna IDPAESE scriviamo 1, che è il codice identificativo del Paese Italia. Diamo INVIO e vediamo che i nostri dati vengono acquisiti con comparsa del numero 4 nel campo IDAUTORE, quello che si incrementa automaticamente.

Ora chiudiamo la tabella AUTORI (click sulla crocetta x in alto a destra oppure da menu FILE ▷ CHIUDI) e, con doppio click sul nome della tabella TITOLI apriamo questa.

Inseriamo Il codice Rebecca nel campo TITOLO e 2 nel campo IDAUTORE, che corrisponde al codice di Ken Follett, diamo INVIO e poi inseriamo Caos calmo e 4 come IDAUTORE di Sandro Veronesi.

Tutto sommato la noiosa operazione del data entry si è abbastanza semplificata rispetto ai sistemi sperimentati prima: rimane il fastidio di dover ricordare o di avere sottomano un prontuario per i codici identificativi degli autori e dei paesi. Ma con la potenza di Base risolveremo anche questo problema.

Dacché siamo nella zona TABELLE, diamo una scorsa alla finestra in alto a destra, denominata ATTIVITÀ, dove ci vengono proposte tre procedure, due con le quali possiamo creare nuove tabelle e una con cui possiamo creare una vista.

Se lavoriamo con sqlite3 è bene ignorare le due procedure per la creazione di nuove tabelle, in quanto ci troveremmo a lavorare con terminologie del motore di database del modulo Base di LibreOffice e faremmo una tremenda confusione. Interventi di questa natura per modificare la struttura del database SQLite dobbiamo farle con sqlite3.

Quanto alla creazione di viste, la vista non è altro che una tabella creata utilizzando dati di un'altra tabella o di altre tabelle e, in realtà, si tratta di una ricerca della quale si memorizza il risultato. Per cui il discorso lo rimando al prossimo capitoletto.

## Ricerche

Se clicchiamo sull'icona RICERCHE nella zona DATABASE, si apre la finestra delle ricerche, nella quale non abbiamo nulla di evidenziato.

---

<sup>4</sup>Se i campi sono visualizzati in spazi troppo ristretti, li possiamo allargare agendo nella barra dove compaiono le intestazioni di colonna, posizionando il mouse sulla barretta di delimitazione destra della colonna da allargare fino a che il cursore si trasformi in doppia freccia e, mantenendo premuto il pulsante sinistro del mouse, spostare il cursore fino a che la colonna assume la larghezza che ci va bene.


Nella finestra sovrastante, quella delle ATTIVITÀ, ci vengono proposte tre modalità per creare una ricerca. Prescindiamo da quella che propone una procedura guidata in quanto non è adatta per una ricerca su un database relazionale e vediamo le altre due.

## Crea ricerca in vista SQL

Partiamo dall'ultima, attraverso la quale possiamo creare una ricerca utilizzando il linguaggio SQL, cioè il comando SELECT, che già conosciamo.

Se scegliamo questa modalità, cliccandoci sopra, ci troviamo di fronte un editor nel quale scriviamo il comando; se siamo interessati ad elencare le opere di Ken Follett, per esempio, scriviamo

```
select titolo from titoli, autori where titoli.idAutore=autori.idAutore and autori.nome like 'Follett%'.
```

Ora premiamo sul pulsante  nella barra degli strumenti o selezioniamo da menu MODIFICA > ESEGUI RICERCA o premiamo il tasto F5 e si aprirà la finestra riprodotta nella figura 6, dove, sopra la finestra dell'editor con scritto il nostro comando SQL, compare una finestra con il risultato della ricerca.

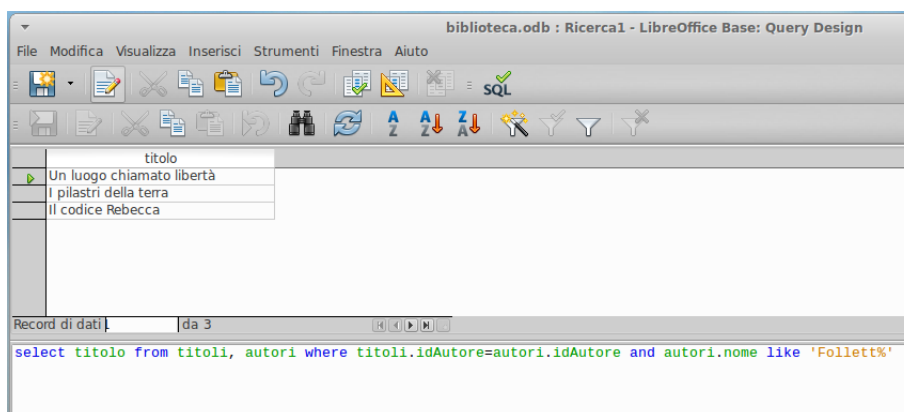


Figura 6: Finestra di ricerca in vista SQL

Come esercizio istruttivo consiglio di passare il mouse su tutte le icone dei numerosi pulsanti per acquisire conoscenza di tutte le belle cose che possiamo fare in questa finestra.

Dal momento che una ricerca destinata ad elencare i libri di un certo autore che abbiamo in biblioteca ci può essere utile sempre, ci conviene memorizzarla con un nome descrittivo, ad esempio «titoli\_per\_autore». Se abbiamo fatto l'esercizio prima suggerito sappiamo quale pulsante utilizzare per la memorizzazione, possibile anche ricorrendo al menu FILE > SALVA CON NOME....

## Crea ricerca in vista struttura

Per chi non conosce il linguaggio SQL, questa è la modalità alternativa. Se la scegliamo, cliccandoci sopra, ci troviamo di fronte una finestra come quella di figura 7 alla pagina seguente.

Proponiamoci di creare una ricerca che ci fornisca nomi dell'autore e titolo di tutte le opere di autori italiani che abbiamo in biblioteca.

Nella finestrella che vediamo in primo piano abbiamo l'elenco delle tabelle che compongono il nostro database. Per la nostra ricerca ci interessano le tabelle dei paesi, degli autori e dei titoli in quanto i dati che vogliamo estrarre coinvolgono tutte e tre.

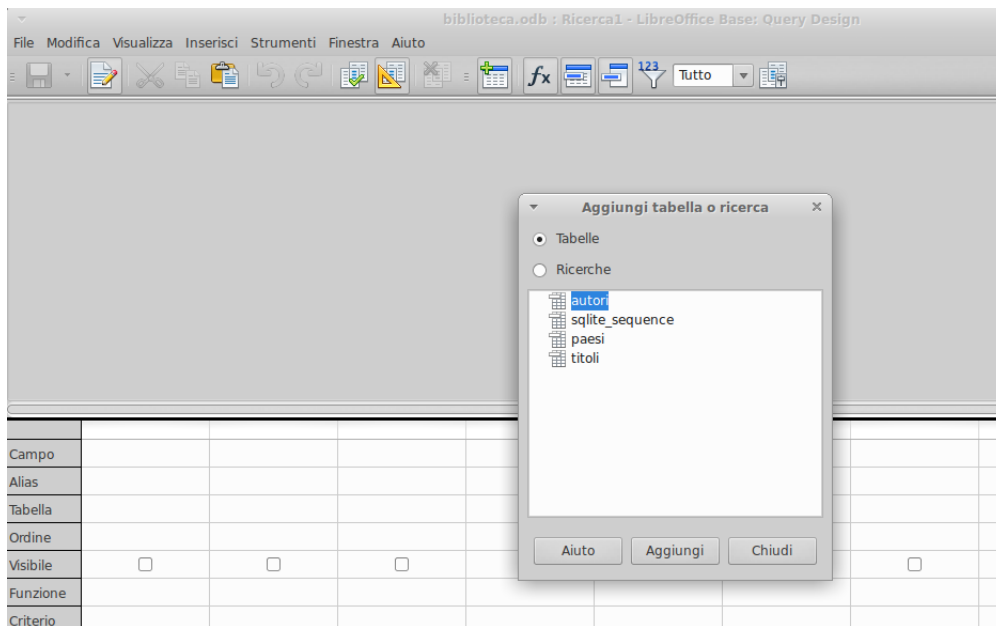


Figura 7: Finestra di creazione ricerca in vista struttura

Selezioniamo la tabella paesi e clicchiamo su AGGIUNGI e facciamo così per le tabelle autori e titoli. Fatto questo clicchiamo su CHIUDI.

Nella zona superiore della finestra ci ritroviamo gli schemi delle tre tabelle



Questi schemi ci servono per indicare le relazioni esistenti tra le tabelle che siano utili alla ricerca che vogliamo effettuare.

Nel nostro caso, dovendo estrarre il nome degli autori italiani, la prima relazione utile è quella tra idPaese della tabella dei paesi e l'idPaese della tabella degli autori. Con il mouse andiamo pertanto a collegare, per trascinamento, queste due voci negli schemi delle prime due tabelle (puntiamo con il mouse su idPaese della tabella dei paesi e, con premuto il tasto sinistro del mouse, lo trasciniamo su idPaese della tabella degli autori).

L'altra relazione utile, dovendo estrarre i titoli delle opere di questi autori, è quella tra idAutore della tabella degli autori e idAutore della tabella dei titoli. Con il mouse andiamo pertanto a collegare queste due voci tra la seconda e la terza tabella.

Con questi interventi gli schemi delle tre tabelle appaiono ora con le relazioni evidenziate, così



Ora dobbiamo stabilire che cosa vogliamo e farne richiesta nella parte inferiore della finestra di figura 7.

Il risultato della ricerca dovrebbe comparire come una tabella con due colonne, la prima contenente il nome dell'autore, la seconda contenente il titolo dell'opera, il tutto riferito ad autori italiani.

Andiamo allora nella prima colonna della finestra, con un click sulla riga CAMPO apriamo il menu a caduta e selezioniamo AUTORI.NOME.

Nella riga ALIAS possiamo indicare l'intestazione della colonna nella tabella che mostrerà il risultato della ricerca: se non lo facciamo, questa intestazione corrisponderà al nome della colonna nella tabella del database. Nel nostro caso indichiamo Autore, in quanto più esplicativo di Nome.


Cliccando sul campo libero nella riga TABELLA esce la conferma della tabella degli autori.

Nella riga ORDINE possiamo scegliere che gli autori compaiano in ordine alfabetico crescente, quindi clicchiamo su crescente nel menu a tendina.

Nella riga VISIBILE, visto che il nome dell'autore vogliamo che compaia nella tabella, selezioniamo la casellina.

Andiamo ora nella seconda colonna e selezioniamo come CAMPO TITOLI.TITOLO. Lasciamo perdere l'alias in quanto l'intestazione titolo è adatta. Lasciamo perdere l'ordinamento e scegliamo anche in questo caso VISIBILE.

Nella terza colonna inseriamo la condizione della ricerca che consiste nel fatto che il campo paese della tabella paesi deve contenere la stringa 'Italia'. Selezioniamo, nel solito modo, PAESI.PAESE nella riga del CAMPO. Facciamo in modo che il quadratino VISIBILE non sia selezionato, in quanto non ci interessa vedere il paese nel risultato della ricerca. Nella riga CRITERIO inseriamo la dizione like 'Italia' (alcune versioni di OpenOffice o di LibreOffice capiscono anche l'italiano e potremmo inserire come 'Italia').

Con click sul pulsante  eseguiamo la ricerca, il cui risultato compare nella zona superiore della finestra, come in figura 8.

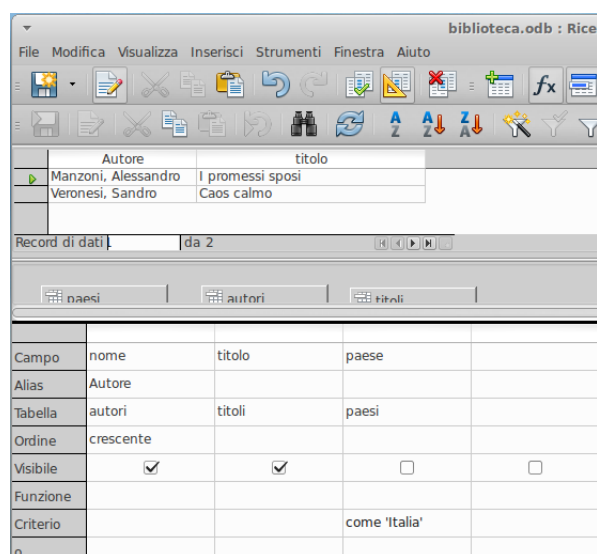


Figura 8: Query in vista struttura e relativo risultato

Se riteniamo che ci possa servire ancora, memorizziamo questa ricerca dandole un nome che ne ricordi lo scopo, per esempio «opere\_per\_paese\_autore».

\* \* \*

Le ricerche che abbiamo memorizzato, da qui in poi compariranno nel quadrante RICERCHE della finestra delle ricerche ed il loro risultato comparirà semplicemente facendo doppio click su di esse. Così facendo, tuttavia, il risultato rispecchierà l'impostazione che aveva la ricerca al momento della memorizzazione: l'ultima prodotta, per

esempio, relativa all'elencazione di autori e opere per autori italiani, produrrà sempre questo risultato.

Essa, tuttavia, ci può essere utile per estrarre autori e opere per autori statunitensi: per arrivare a questo risultato basta che clicchiamo destro sul nome della ricerca e scegliamo MODIFICA nel menu a discesa che si apre. Nella successiva finestra sostituiamo USA a Italia nella condizione COME, clicchiamo sul pulsante di esecuzione e otteniamo l'elenco degli autori e delle opere per autori statunitensi. Al momento dell'uscita dalla finestra ci viene richiesto se memorizzare: possiamo farne a meno e tenerci il modello relativo all'Italia.

Stessa cosa se vogliamo utilizzare la ricerca che avevamo impostato per estrarre i titoli delle opere di Ken Follett per estrarre le opere di Steinbeck.

## Formulari

Il formulario è uno strumento utile per gestire i dati contenuti in una tabella del database: in particolare per scorrere i record, per ricercare un particolare record, semplicemente per vederlo o per modificarlo. Di grande utilità può risultare quando ne facciamo una maschera per l'inserimento dei dati. Gli esempi che vediamo sono destinati soprattutto a questo uso.

Nella zona ATTIVITÀ della finestra che si apre cliccando sull'icona FORMULARI, ci vengono proposte due modalità per la creazione di formulari: in vista struttura oppure seguendo una procedura guidata.

### Creazione del formulario in vista struttura

Abbiamo visto che le tabelle si possono visualizzare, modificare e alimentare con nuovi dati semplicemente aprendole con doppio click sul loro nome nella finestra TABELLE. Nella finestra compare la tabella con tutti gli strumenti per cercare un particolare record, per ordinamenti e, semplicemente scrivendoci dentro, possiamo modificare il contenuto di certi campi o aggiungere nuovi record, come peraltro abbiamo già fatto.

Un modo più elegante può essere quello di ricorrere a un formulario.



Ora ne creiamo uno per la tabella degli autori del nostro database «biblioteca».

Nella zona ATTIVITÀ della finestra FORMULARI clicchiamo su CREA FORMULARIO IN VISTA STRUTTURA....

Apriamo così l'editor dei formulari, come compare nella figura 9 nella pagina successiva.

Passando il mouse sulle icone dei numerosi pulsanti che circondano l'area di lavoro bianca vediamo quali funzioni essi possono attivare e ci accorgeremo che quelli localizzati nelle barre orizzontali, superiore e inferiore, attivano funzioni da modulo di scrittura (Writer) o di disegno (Draw) mentre quelli localizzati nella barra verticale di sinistra, per chi conosce tutti i moduli di LibreOffice, rappresentano delle novità: si tratta, infatti, di funzioni di controllo create apposta per lavorare su un formulario che si colleghi ad un database.

In questa sede non ci occuperemo di quelle che sottintendono l'esistenza di macro per attivare strane diavolerie programmate con il basic di LibreOffice (vedi pulsante, pulsante di scelta, ecc.) ma ci limitiamo a quelle di semplice interfacciamento con un database, che sono, in definitiva due:

- la casella di testo, contrassegnata dall'icona ,
- la casella di riepilogo, contrassegnata dall'icona .

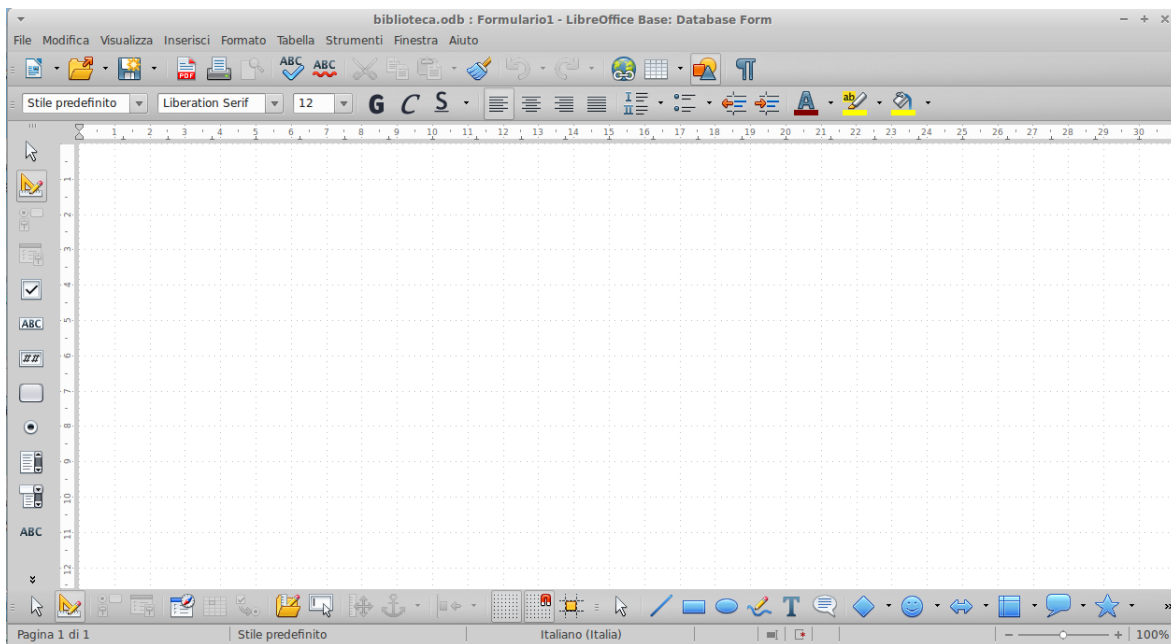


Figura 9: Finestra per la creazione di un formulario in vista struttura.

Dal momento che vorremmo utilizzare il formulario, oltre che per scorrere la tabella degli autori, anche per inserire nuovi autori, dobbiamo inserire nel formulario stesso tutti i campi della tabella oggetto di inserimento, salvo il primo, contenente l'idAutore, che, ricordiamo, è inserito automaticamente.

Dobbiamo allora inserire nell'area di lavoro, nelle posizioni che più ci aggradano o che più siano comode per la consultazione o per l'inserimento di nuovi dati, tante caselle di testo quanti sono i campi del database che ci interessa evidenziare.

Dal momento che in uno di questi campi, quello dedicato al codice identificativo del paese, idPaese, è inserito un dato che relaziona la tabella autori con un'altra tabella, quella dei paesi, per questo campo non usiamo una semplice casella di testo ma usiamo una casella di riepilogo, in modo che ci venga mostrato non già il codice del paese che c'è nella tabella degli autori ma la dicitura del paese che c'è nella tabella dei paesi.

Per fare questo dobbiamo però preconstituire una ricerca che produca una tabella dei paesi con il nome dei paesi in chiaro nella prima colonna e l'idPaese nella seconda. In vista struttura questa ricerca si crea così:

paesi		
Campo	paese	idPaese
Alias		
Tabella	paesi	paesi
Ordine	crescente	
Visibile	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Salviamo la ricerca con il nome \_paesi (il trattino di sottolineatura iniziale può essere utile per ricordare che si tratta di una ricerca, direi, di servizio e, applicato a tutte le altre di questa natura, le faccia comparire tutte insieme nell'elenco delle ricerche che vediamo, ordinato, nella relativa finestra; evitiamo inoltre che la ricerca paesi faccia casino con la tabella omonima paesi che abbiamo nel database).

Con un piccolo lavoro di grafica che ho eseguito con competenze che non hanno a che vedere con la gestione di un database e che un creativo potrà certamente fare meglio di me, ho creato il formulario che compare nella figura 10 alla pagina seguente.

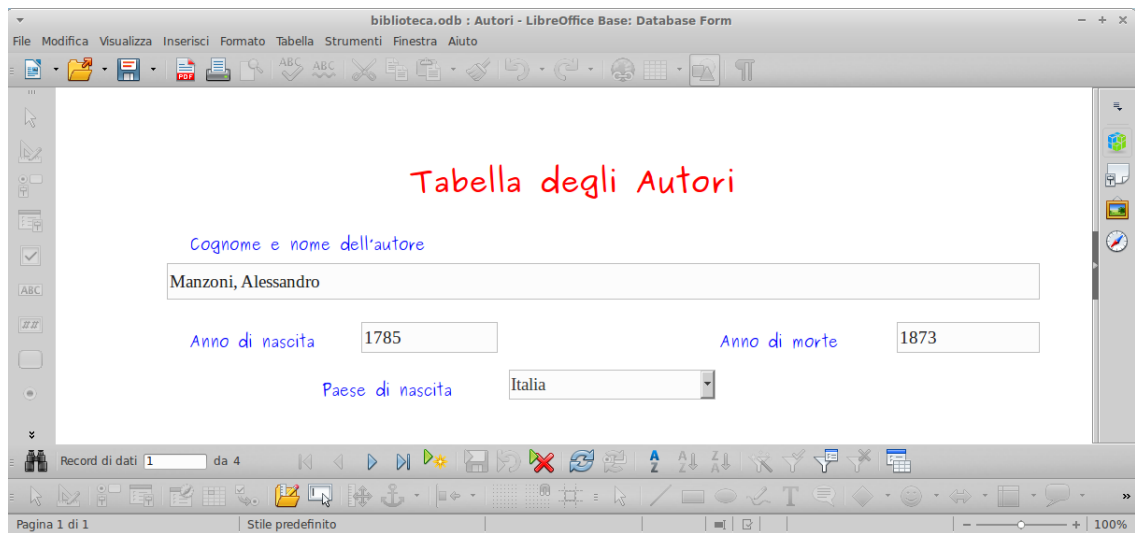



Figura 10: Formulario relativo alla tabella Autori

Vi compare il primo record della nostra tabella degli autori.

Nella barra degli strumenti appena sotto l'area dei dati abbiamo i pulsanti che attivano le varie funzionalità. Basta passare il mouse sui vari pulsanti per vedere la descrizione di queste funzionalità.

Devo invece dire qualche cosa su come si possa ottenere che nelle varie caselle di testo compaiano i dati che stanno nella tabella e nella casella di riepilogo compaia il dato che sta in un'altra tabella, non solo, ma come mai in questa casella di riepilogo, cliccando sul piccolo pulsante nella parte finale destra della casella, si apra un elenco di tutti i paesi contenuti nella tabella dei paesi.

Una volta aperta la finestra per la creazione del formulario, quella che vediamo nella figura 9, la prima cosa da fare è l'inserimento dei controlli che ci interessano. Inserimento che avviene cliccando sull'icona che identifica il controllo nella barra verticale di sinistra, utilizzando poi il mouse, il cui puntatore si è trasformato in una crocetta per disegnare rettangoli, per dare una posizione, un'altezza e una larghezza alla casella del controllo nell'area di lavoro.

Inseriti i controlli, diventa attivabile il pulsante per le proprietà del formulario, che troviamo nella barra degli strumenti inferiore, identificato dall'icona . Cliccando su di esso apriamo la finestra di dialogo PROPRIETÀ FORMULARIO nella quale, utilizzando la prima scheda GENERALE, diamo un nome al formulario scrivendolo nella finestrella NOME (nel nostro caso scriviamo Autori). Utilizzando la scheda DATI selezioniamo «Tabella» nella finestrella TIPO DI CONTENUTO e selezioniamo «autori» nella finestrella CONTENUTO. In entrambe le schede lasciamo invariate le scelte di default. In questo modo abbiamo stabilito che il formulario si chiami «autori», riguardi una tabella e la tabella sia quella degli autori.

Ora dobbiamo collegare le caselle di testo ai campi della tabella. Con doppio click all'interno della casella di testo apriamo la finestra di dialogo PROPRIETÀ: CASELLA DI TESTO. Utilizzando la scheda GENERALE potremmo dare un nome alla casella, cosa che ritengo inutile, andando benissimo quello assegnato per default e tra le tante altre cose possiamo lasciare tutto com'è, salvo, volendo, scegliere il font e la relativa dimensione per i caratteri: scorrendo la finestra si arriva alla riga dedicata al CARATTERE e, nel mio caso, ho scelto Liberation Serif, Normale, 14 avvalendomi della finestra che si apre cliccando sul pulsantino con i tre puntini. Passiamo poi alla scheda DATI e scegliamo il dato che dovrà riguardare la casella interessata, aprendo il menu a discesa nella riga

CAMPO DI DATI. Ovviamente scegliamo «nome» se stiamo lavorando sulla casella di testo destinata al nome, ecc.

Ci rimane da trattare la casella di riepilogo. Con doppio click al suo interno apriamo la finestra di dialogo PROPRIETÀ: CASELLA DI RIEPILOGO. Utilizzando la scheda GENERALE, sempre essendo inutile assegnare un nome diverso da quello proposto per default, possiamo scegliere il font e la dimensione nella riga CARATTERE, come abbiamo fatto prima, e dobbiamo scorrere fino alla riga APRIBILE, scegliendo «Si». In questo modo stabiliamo che la casella si possa aprire in un menu a tendina cliccando sul pulsantino che essa conterrà in fondo a destra. Nella riga successiva, NUMERO DELLE RIGHE, possiamo anche indicare il numero di righe che debba contenere il menu a tendina alla prima apertura, fermo restando che, se le righe sono di più, il menu conterrà una barra di scorrimento per visualizzare anche le altre. Utilizzando la scheda DATI scegliamo il dato che dovrà riguardare la casella, nel nostro caso scegliamo «id-Paese» nel menu a tendina che si apre cliccando sul pulsantino in fondo a destra della finestrella relativa al CAMPO DI DATI. Nella riga TIPO DEL CONTENUTO DELLA LISTA scegliamo «Ricerca» dal menu a tendina (in certe edizioni di OpenOffice o LibreOffice può comparire l'omologa voce «Query»), nella riga CONTENUTO ELENCO scegliamo la ricerca che avevamo creato a questo scopo, che avevamo chiamato «\_paesi» e lasciamo il valore 1 proposto per il Campo collegato.

Salviamo il nostro formulario con il nome Autori.

### **Creazione del formulario usando la procedura guidata**

Se scegliamo di creare il formulario seguendo la procedura guidata ci si semplifica tutto il lavoro, innanzi tutto quello di grafica, necessario per dare un aspetto accettabile al formulario stesso: l'aspetto lo possiamo scegliere tra alcuni modelli che ci vengono proposti.

Con questa modalità ora creiamo un formulario per la tabella dei titoli.

Cliccando nella zona ATTIVITÀ della finestra FORMULARI su USA PROCEDURA GUIDATA PER LA CREAZIONE DEI FORMULARI..., apriamo la finestra di dialogo CREAZIONE GUIDATA FORMULARIO.

Il primo passo riguarda la selezione dei campi, che faremo previa scelta, nel riquadro TABELLE O RICERCHE, della tabella «titoli». Fatta questa scelta, nella finestrella CAMPI DISPONIBILI compare l'elenco dei tre campi idTitolo, titolo e idAutore. Il primo non ci interessa, pertanto selezioniamo gli altri due e li spostiamo nella finestrella CAMPI NEL FORMULARIO agendo sui pulsanti a freccia che si trovano tra le due finestrelle. Clicchiamo su AVANTI e passiamo al passo 2, CONFIGURA UN FORMULARIO SECONDARIO. Dal momento che non siamo interessati a questo, clicchiamo su AVANTI e saltiamo al passo 5, DISPONI I CAMPI DI CONTROLLO NEL QUESTIONARIO. Dei quattro modelli di disposizione che ci vengono proposti scegliamo, per esempio, il secondo, che disporrà le caselle dei dati con scritta sopra la loro descrizione. Clicchiamo su AVANTI e passiamo al passo 6, IMPOSTA IMMISSIONE DATI. Nel frattempo, sotto la finestra di dialogo sulla quale stiamo lavorando compare il formulario che abbiamo disegnato. Possiamo scegliere se il formulario debba servire solo per l'inserimento di nuovi dati, senza visualizzare i dati esistenti, o se debba visualizzare tutti i dati: scegliamo pure questa seconda modalità, che dovrebbe essere quella selezionata per default. Clicchiamo su AVANTI e affrontiamo il passo 7, APPLICA LO STILE DI FORMATO DEL QUESTIONARIO. Se non abbiamo esigenze particolari accettiamo la proposta di default BEIGE e ASPETTO 3D e clicchiamo su AVANTI. Nel passo 8, IMPOSTA IL NOME DEL QUESTIONARIO, accettiamo la proposta di default di nominarlo con lo stesso nome



della tabella cui si riferisce, «titoli» e, visto che ancora dobbiamo lavorare sul formulario, alla domanda COME VUOI PROCEDERE UNA VOLTA CREATO IL FORMULARIO? rispondiamo scegliendo MODIFICA IL FORMULARIO e clicchiamo su FINE.

A questo punto ci troviamo di fronte la stessa finestra riprodotta nella figura 9, relativa alla creazione di un formulario in vista struttura, nella quale ritroviamo il disegno del formulario come lo abbiamo prodotto con la procedura guidata.

A parte modifiche grafiche ancora possibili, per esempio quella di inserire un testo che identifichi la natura del formulario, come «Tabella dei titoli» e quella di rendere più largo e meno alto il riquadro del titolo, oltre che di allungare un po' il campo dell'idAutore, dobbiamo sistemare il funzionamento del controllo relativo a quest'ultimo campo.

Una precisazione per quanto riguarda la selezione dei controlli da modificare. Con doppio click sul controllo si effettua una selezione multipla del controllo in blocco, cioè della finestrella del controllo vero e proprio e della scritta che lo descrive: questo può essere utile, per esempio, per spostare tutto il blocco. Per selezionare una parte del controllo, per esempio solo la casella di testo, occorre puntare il mouse al suo interno e premere il tasto sinistro del mouse con premuto il tasto CTRL della tastiera.

Dicevamo che dobbiamo sistemare il funzionamento del controllo relativo all'idAutore. Ciò in quanto la procedura di creazione guidata inserisce, per tutti i campi della tabella, delle caselle di testo e le collega alla tabella del database. Se lasciassimo il nostro formulario così com'è uscito dalla procedura guidata, nella casella di testo dell'idAutore non vedremmo il nome dell'autore ma il suo codice numerico. A noi interessa invece che in questa casella si veda il nome dell'autore.

Come abbiamo fatto per i paesi nell'esercitazione svolta nel precedente capitoletto, dobbiamo allora innanzi tutto creare una ricerca che elenchi in ordine alfabetico autori e rispettivo codice, con lo stesso procedimento che abbiamo prima visto per i paesi, e salvare questa ricerca con il nome «\_autori».

Dopo di che selezioniamo la casella di testo dell'idAutore, solo la casella di testo, e con la casella selezionata clicchiamo destro su di essa e scegliamo la voce SOSTITUISCI CON: nel menu a tendina che si apre e, nella finestra successiva, scegliamo CASELLA DI RIEPILOGO. In questo modo la nostra casella non è più una casella di testo ma una casella di riepilogo. Sempre con la casella selezionata clicchiamo nuovamente destro su di essa e, dal menu che compare, scegliamo la voce CAMPO DI CONTROLLO. Nella successiva finestra di dialogo facciamo ciò che abbiamo fatto prima per la casella dell'idPaese: cioè attiviamo l'attributo APRIBILE nella scheda GENERALE e nella successiva scheda DATI scegliamo «ricerca» come TIPO DEL CONTENUTO DELLA LISTA e «\_autori» come CONTENUTO ELENCO e salviamo il formulario.

Abbiamo così a disposizione il formulario anche per la tabella dei titoli, e, se l'abbiamo creato come prima descritto, esso avrà l'aspetto riprodotto nella figura 11 alla pagina seguente.

\* \* \*

A questo punto acquistiamo un altro volume, Guerra e pace di Lev Tolstoj e lo dobbiamo inserire nel database. Lev Tolstoj è un autore russo vissuto tra il 1828 e il 1910. Apriamo la ricerca \_paesi e constatiamo che la Russia non vi è contemplata. Allora andiamo nella finestra delle tabelle, apriamo la tabella dei paesi con doppio click sul suo nome nell'elenco e inseriamo Russia nella prima casella vuota della colonna paese. Trattandosi di un inserimento semplice semplice non siamo stati a creare un formulario per farlo e abbiamo inserito direttamente da tabella. Ora verificiamo, aprendo la

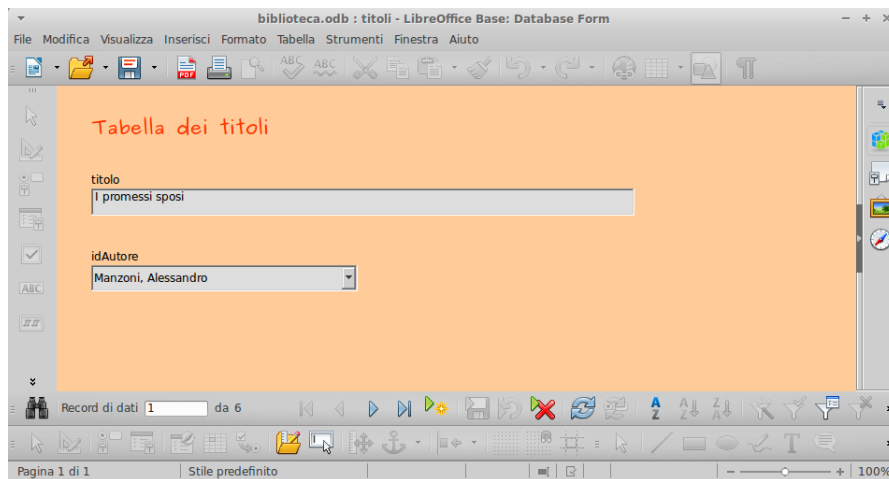




Figura 11: Formulario relativo alla tabella Titoli

ricerca `_autori`, che Tolstoj non c'è. Allora andiamo nella finestra dei formulari, clicchiamo sul formulario Autori e sul pulsante  nella barra degli strumenti appena sotto il formulario, in modo da raggiungere il primo record vuoto della tabella. Inseriamo nelle rispettive caselle il nome dell'autore, scritto Tolstoj, Lev, secondo lo stile con cui abbiamo scritto gli altri autori, l'anno della nascita e quello della morte. Arrivati alla casella `idPaese` clicchiamo sul pulsantino che la chiude sulla destra e scegliamo Russia dal menu a tendina: questo menu ci permette di inserire il codice paese della Russia senza conoscerlo. Memorizziamo il record cliccando sul pulsante  subito a sinistra di quello di prima (se usciamo da formulario senza farlo saremo invitati a decidere se salvare la modifica o meno). Finalmente inseriamo il titolo avvalendoci del formulario per i titoli. Anche qui, il titolo lo scriviamo mentre l'`idAutore` lo inseriamo ricorrendo al menu a tendina collegato con la relativa casella e scegliendo Tolstoj nell'elenco che compare.

## Rapporti

E' disponibile una procedura guidata per la creazione di rapporti, che possiamo scegliere dalle attività aprendo la finestra dei rapporti. Se abbiamo installato il report builder, che è un'estensione di LibreOffice, e Java possiamo anche ricorrere alla creazione del rapporto in vista struttura. Entrambe le procedure sono tuttavia macchinose e portano a risultati deludenti, soprattutto sul piano della libertà di esposizione, per cui le sconsiglio.

Molto più produttivo ritengo sia l'uso degli altri moduli di LibreOffice per inserire dati ricavati da Base in rapporti (Writer), tabelle (Calc), presentazioni (Impress) o disegni (Draw).

Il database prodotto o collegato via ODBC con Base è infatti registrabile in LibreOffice. Si ricorderà che, quando abbiamo collegato il nostro database SQLite «Biblioteca» a Base abbiamo scelto di registrarlo. Se non lo avessimo fatto potremmo farlo ora. Basterebbe aprire un qualsiasi modulo di LibreOffice e cliccare sul tasto F4 o, come accade per le più recenti edizioni di LibreOffice sulla combinazione di tasti CTRL + SHIFT + F4 (in Writer e Calc è disponibile anche la voce di menu `VISUALIZZA > SORGENTE DATI`). In questo modo, appena sopra l'area di lavoro, si apre una finestrella contenente l'elenco dei database registrati. Se quello che ci interessa non c'è e lo vogliamo inserire, clicchiamo destro nella zona dove c'è l'elenco dei database registrati e scegliamo

mo DATABASE REGISTRATI dal menu a tendina. Nella successiva finestra di dialogo clicchiamo su NUOVO e inseriamo il database che vogliamo registrare seguendo i vari passaggi guidati.

Il database inserito compare con la possibilità di visualizzare gli elenchi delle ricerche e delle tabelle che contiene. Per trascinamento con il mouse, a partire da questi elenchi, possiamo inserire i valori di ricerche e tabelle nell'area di lavoro del modulo LibreOffice in cui vogliamo riportare questi valori, con tutte le possibilità di titolazione e commento collaterale, formattazione, colorazione e quant'altro ci consenta il modulo stesso (se siamo in Calc anche elaborazioni). Ovviamente dobbiamo prima costruire e memorizzare le tabelle o le ricerche che ci servono.

## 6 Collegamenti attraverso i principali linguaggi di programmazione.

All'indirizzo <http://www.w3ii.com/it/sqlite/default.html>, oltre che trovare una buona guida a SQLite in generale, possiamo vedere svolte esercitazioni di collegamento a database sqlite3 utilizzando i linguaggi C/C++, Java, PHP, Perl e Python, collegamenti destinati ad interagire con il motore sqlite3 per creare database, tabelle, alimentarle, leggerle.

La guida è nata in lingua inglese e quella che consultiamo all'indirizzo sopra indicato ne è una traduzione evidentemente fatta da un traduttore non umano e contiene qualche strafalcione. Per chi conosce l'inglese è comunque accessibile la versione originaria scegliendo la lingua inglese nella finestrella di scelta in alto a destra della pagina web.

Lasciamo C e C++ ai professionisti in quanto di quelli sopra citati è il più ostico, a partire dalla necessità di procurarsi l'header `sqlite3.h`, sapere dove metterlo, ecc. Per chi voglia provare a svolgere qualche esercizio suggerito nella citata guida in C/C++ segnalo la via diretta che ci fornisce Linux Ubuntu per sistemare il compilatore GNU: basta installare la libreria di sviluppo con il comando `sudo apt-get install libsqlite3-dev`. Trattandosi di libreria esterna, bisogna ricordare, oltre all'inclusione dell'header `sqlite3.h`, di linkare la libreria al momento della compilazione con l'opzione `-l sqlite3`.

Anche per Java l'utilizzo non è direttamente possibile: occorre un binding con il file `sqlitejdbc.jar` che si può trovare su Internet. Per chi usa Netbeans è installabile come plugin.

Quanto a PHP, tradizionalmente legato a MySQL, da quando sono sorti alcuni problemi di licenza l'implementazione nativa del linguaggio non è più quella di MySQL ma è proprio quella di SQLite, attraverso la libreria PDO, integrata all'interno di PHP.

Perl non è direttamente utilizzabile ma va arricchito del modulo Perl DBI, che possiamo procurarci collegandoci al sito <https://dbi.perl.org/>, dove si trova pure la documentazione. L'uso di Perl non è facile.

Fin da Python 2.5, cioè da anni, il modulo SQLite è incluso nella libreria standard, inizialmente `pysqlite2` e, da Python 2.6, `sqlite3`. Ritengo che l'approccio di Python per lavorare con SQLite sia il più semplice di tutti e ho pensato di illustrarlo brevemente, riferendomi alla versione 3 di Python, che è ormai quella corrente e, ovviamente, a `sqlite3`.

### Utilizzo di Python3

Una volta importato il modulo con

```
import sqlite3
```

creiamo una connessione con il database, che chiamiamo con un nome evocativo, per esempio, **db**, con

```
db = sqlite3.connect("<database>")
```

dove <database> sta per percorso e nome del database, messi tra doppi apici come stringa.

Poi, per lavorare sul database con cui siamo collegati, utilizzando il metodo cursor() della connessione creiamo un cursore, che chiamiamo, per esempio, con la sua iniziale **c**, con

```
c = db.cursor().
```

I principali metodi del cursore così creato ci consentono di eseguire sul database comandi SQL o di leggere i risultati di una query, in particolare:

```
c.execute("<comando_SQL>")
```

esegue un comando SQL indicato come stringa o come variabile stringa;

```
risultato = c.fetchall()
```

crea una lista, chiamata risultato, nella quale ogni elemento è una tupla che rappresenta la riga di una astratta tabella con i dati della query per riga e colonna, come estratti dal database su cui è stata fatta la ricerca, dati a loro volta rappresentati sotto forma di stringhe unicode.

Alla fine del nostro lavoro dobbiamo chiudere il database con

```
db.close().
```

In breve penso di aver detto le cose che servono nella stragrande maggioranza dei casi. Una documentazione completa sulla semplicità e la potenza di Python per fare queste cose si trova all'indirizzo <https://docs.python.org/2/library/sqlite3.html>.

Facciamo un piccolo esempio: ci proponiamo di estrarre il contenuto della tabella dei Paesi del database «biblioteca» che abbiamo creato e alimentato per esercizio nei precedenti capitoli.

Lo script Python, secondo quanto abbiamo appena visto, sarà il seguente:

```
import sqlite3
db = sqlite3.connect("/home/vittorio/Database/biblioteca") 5
c = db.cursor()
c.execute("select * from paesi")
risultato = c.fetchall()
print(risultato)
db.close()
```

L'esecuzione dello script fornisce il seguente risultato:

```
[(1, 'Italia'), (2, 'Gran Bretagna'), (3, 'USA'), (4, 'Russia')]
```

Se non ci piace questo modo grezzo di presentare il risultato e lo vogliamo esporre meglio possiamo intervenire inserendo, al posto dell'istruzione print(risultato), il seguente blocchetto:

```
for i in range(len(risultato)):
    ri = risultato[i]
    r = str(ri[0]) + " " + ri[1]
    print r
```

nel quale ogni elemento della lista risultato viene preso come una tupla isolata, ogni elemento della tupla viene accostato all'altro come stringa, con in mezzo un piccolo spazio libero, e ogni stringa risultante viene stampata.

---

<sup>5</sup>Sul mio computer, il database «biblioteca» si trova nella directory Database della mia home; essendo su un sistema Linux, come avverrebbe su un sistema Mac, l'indirizzo al database è indicato con le barre inclinate al contrario di quelle che si usano in Windows.

Il risultato ora si presenta così:

- 1 Italia
- 2 Gran Bretagna
- 3 USA
- 4 Russia

Occorre ricordare che la stringa risultante da quest'ultima operazione è una stringa unicode. Pertanto, se essa, anziché sullo standard output, come avviene per default nel nostro esempio, fosse diretta a un file, la stringa `r` oggetto dell'istruzione `print` andrebbe scritta `r.encode('utf-8')`; in caso contrario avremmo un messaggio di errore e il file non potrebbe essere scritto.

\* \* \*

L'approccio esemplificato utilizzando il linguaggio Python è praticamente lo stesso, magari con qualche complicazione formale in più, per tutti gli altri linguaggi di programmazione che ho prima ricordato.

Ci si collega a un database con una istruzione che, se il database non c'è, lo crea; si crea un cursore attraverso cui si eseguono comandi SQL per creare tabelle, alimentarle, leggerle. E tutto ciò possiamo inserirlo in un programma più ampio, vuoi per predisporre un ambiente grafico attorno al database come tale (ad esempio finestre per l'inserimento dei dati), vuoi per fornire un database ad un software destinato a tutt'altro scopo ma che abbia bisogno, al proprio interno, di una funzione di database.