

OpenOffice.org

&

MySql

Introduzione All'uso Dei
Database Con OpenOffice.org
Parte Prima

retrocopertina

Titolo Originale : *OpenOffice.org & MySql. Introduzione all'uso dei Database con
OpenOffice.org Parte Prima Versione : 1.0*

1a Edizione : *Maggio 2004*

Indice generale

1	Introduzione.....	1
1.1	Falsi problemi e guerre di religione.....	2
2	Riscaldiamo i motori.....	3
2.1	Configurazione utilizzata.....	3
2.2	Installazione di OOo.....	3
2.3	Installazione di MySql.....	4
2.4	Installazione dei Driver ODBC in Windows.....	6
2.5	Installazione del driver ODBC in Linux.....	7
3	Il mondo dei Database.....	9
3.1	Primi passi.....	9
3.2	Progettiamo un Database.....	10
3.3	La Tabella.....	10
3.4	Campi di tipo Stringa.....	11
3.5	Campi di tipo Numerico.....	12
3.6	Campi di Tipo Data/Ora.....	12
3.7	Campi particolari: Intero ad incremento automatico.....	13
3.8	Campi particolari : Timestamp.....	13
3.9	Indici.....	13
3.10	Il nostro Database di esempio.....	14
4	Hands On.....	18
4.1	Struttura e Generazione del Database con MySqlCC	18
4.2	Dettagli.....	23
5	Finalmente OOo.....	25
5.1	Come OpenOffice.org si collega ai Database.....	25
5.2	Generazione del DSN in Windows.....	26
5.3	Generazione del DSN in Linux.....	27
5.4	La sorgente dati di OOo.....	28
5.5	Il buon vecchio DBase.....	30
5.6	Gestione delle Sorgenti Dati.....	31
5.7	Modifica della struttura di un Database con OpenOffice.org.....	33
5.8	Il problema dei valori predefiniti.....	34
5.9	Le Ricerche.....	36
6	La Mediateca.....	39
6.1	Filosofie a confronto.....	39
6.2	Il Formulario.....	39
6.3	Il Formulario per la Mediateca.....	44
6.4	Qualche informazione in più.....	47
7	Gestione dei Report.....	48
7.1	Calc per generare e stampare Report.....	48
7.2	La Ricerca per il Report – Relazioni tra Tabelle.....	48
7.3	Il Report in Calc.....	51
7.4	Report su Dati parziali.....	53
8	Fine del viaggio... per ora.....	54
9	Appendice A - Trasferire le Sorgenti Dati.....	55
10	Appendice B – Schema del Database di Esempio.....	56
11	Appendice C – Tipo di Dati di MySql.....	57
11.1	Valori di Tipo Numerico.....	57

11.2 Range dei valori ammessi per il tipo numerico intero.....	59
11.3 Valori di Tipo Date e Time.....	59
11.4 Valori di Tipo Stringa.....	60
12 Licenza e crediti.....	62
12.1 Diritti d'autore e marchi registrati.....	62
12.2 Sezioni Non Modificabili (no Invariant Sections).....	62
12.3 Modifiche ed aggiunte.....	62
12.4 Gnu Free Documentation License.....	62

1 Introduzione

OpenOffice.org è un potente software di produttività personale composto da moduli dedicati ad aspetti diversi della elaborazione delle informazioni, ma ben integrati tra loro. La versione attualmente disponibile (al momento della stesura di questo documento) è la 1.1, ma lo sviluppo prosegue velocemente. OpenOffice.org (nel seguito **OOo**) comprende i tre moduli principali necessari all'uso “office”, cioè una SW di Videoscrittura (**writer**), un Foglio di Calcolo (**Calc**), un SW di Presentazione (**Impress**). La suite inoltre dispone di un modulo di Disegno (**Draw**) che può servire ad integrare immagini anche mediamente complesse nei propri documenti. I moduli di OOo hanno anche la possibilità di “interagire” con archivi esterni in modo da poter utilizzare “dati” disponibili in molti formati diversi all'interno dei Documenti.

Rispetto ad altre soluzioni, anche molto diffuse, OOo ha due importanti vantaggi, a parità di funzionalità e potenza disponibile:

- è **disponibile a basso costo o gratuitamente**, perché il codice sorgente è “libero” (rilasciato sotto una Licenza che ne permette la variazione, l'integrazione e la distribuzione)
- è **multi piattaforma**, cioè può essere usato in modo esattamente identico su molti sistemi operativi, garantendo sempre e comunque lo scambio senza modifiche dei documenti

Scopo di questa guida è illustrare in modo semplice ed accessibile, anche a chi non ha una grande dimestichezza con gli aspetti più complessi dell'informatica, quali possibilità offre OOo per leggere, modificare, integrare, stampare dati provenienti da “basi di dati” (o Database, abbreviato **Db**) esterni al programma stesso. A tal fine useremo un prodotto anch'esso gratuito, libero e multi piattaforma come **MySQL**, che è appunto un “database server” (e vedremo più avanti che cosa significa). Questa scelta è dovuta principalmente alla volontà di non utilizzare strumenti “proprietary” (cioè disponibili solo a pagamento con licenze non libere), senza peraltro limitare troppo le funzionalità del prodotto (MySQL è un potente motore di Database, utilizzato in moltissime realtà professionali).

Vedremo, quindi, come installare, configurare ed utilizzare OOo e MySQL, in ambienti Windows e Linux. Useremo poi un piccolo “database” di esempio per realizzare documenti esemplificativi delle possibilità disponibili. Ovviamente questa non è (e non vuole essere) una guida ai due applicativi nella loro interezza, ed inoltre volutamente tralascieremo gli aspetti più tecnici, sui quali esiste molta documentazione disponibile, ed alla quale rimandiamo per chi volesse approfondire.

Nel testo saranno evidenziate due sezioni particolari:

- i “**tips**” saranno suggerimenti o scorciatoie utili a velocizzare e razionalizzare il lavoro
- la parte “**tecnica**” servirà ad approfondire alcuni argomenti più “difficili”, e può essere “saltata” senza problemi da chi non è interessato.

OOo è liberamente scaricabile dal Sito www.openoffice.org, dove è anche possibile consultare una ottima sezione in Italiano.

MySQL è disponibile su www.mysql.com, ed, a quanto mi risulta, ad oggi non esiste una traduzione italiana del manuale.

Tutto è perfettibile, quindi mi scuso in anticipo per eventuali inesattezze presenti nel testo; suggerimenti, correzioni, integrazioni saranno benvenuti, e potranno servire a migliorare questa documentazione. Buona lettura.

1.1 Falsi problemi e guerre di religione

Una delle domande più frequenti da parte di chi per la prima volta installa e si accinge ad usare OpenOffice.org è : **“ma il modulo dei Database, dov'è ?”** La risposta è semplice: **“non c'è....”**. La domanda è provocata, secondo me, soprattutto da due fattori “psicologici”:

1. l'inclusione, da parte di Microsoft, di Access come componente della suite Office
2. la tendenza, da parte degli utenti, a paragonare immediatamente OOo a Microsoft Office

Quindi è il caso, prima di continuare, di capire meglio come stanno le cose. Innanzi tutto la scelta di includere Access come componente di Office è soprattutto dovuta a problemi di Marketing, più che a scelte tecniche o logiche. Access, infatti, è un prodotto molto diverso dagli altri componenti della suite. Si tratta, come ben sanno quelli che lo usano in modo intensivo, sostanzialmente di uno strumento RAD (Rapid Application Development) orientato ai Database, tra l'altro poco integrabile ad esempio con Word o Excel. Inoltre, in barba alle molte “auto composizioni”, è piuttosto difficile da utilizzare e comprendere, e risultati accettabili si ottengono solo se usato da professionisti con ampio ricorso al linguaggio di programmazione.

La scelta degli sviluppatori di OpenOffice.org è stata quella invece di fornire uno strumento, le **“Sorgenti Dati”**, integrabile facilmente, come vedremo, nei moduli di Calc e writer. Il risultato è stato piuttosto buono, anche se ancora migliorabile. Abbiamo a disposizione una interfaccia verso MOLTI motori di Database, ed in più multi piattaforma, cioè capace di funzionare senza modifiche su molti sistemi operativi diversi. In definitiva, *un utente medio può fare più o meno le stesse cose con OpenOffice.org e con Access*. Ovviamente OOo NON E' (e non vuole essere) un RAD, quindi in quel campo, se si vogliono ottenere buoni risultati, è meglio rivolgersi ad altri prodotti.

Infine, il termine di paragone di OpenOffice.org non è Ms Office, ma... se stesso. Sono due prodotti diversi, che pur volendo fare più o meno la stessa cosa la fanno con filosofie e modalità diverse. Quindi discutere di quale sia migliore è aria fritta. Ognuno sceglierà quello che ritiene più adatto ai propri scopi, e, poi, noi abbiamo già scelto.....

2 Riscaldiamo i motori

2.1 Configurazione utilizzata

Tutti i documenti di esempio sono stati creati e modificati sulle seguenti configurazioni:

Ambiente **Windows**:

- Pc Amd Athlon 2200+, 512 Mb di Ram, Hd 80 Gb, Matrox Millennium G400
- Sistema Operativo Windows 2000 Professional Service Pack 4
- OpenOffice.org Versione 1.1 Italiano
- MySql Versione 4.0.16-nt
- MySql Control Center 0.9.3 Beta
- MySql Odbc Driver 3.51.06 (MyOdbc)

Ambiente **Linux** :

- Pc Amd Duron 1200, 512 Mb Ram, Hd 40 Gb, Nvidia GeForce
- Sistema Operativo Suse Linux Professional Versione 9.0
- Ambiente Desktop KDE 3.1.4
- MyOdbc Linux Driver 3.51.06

La maggior parte dei concetti e delle soluzioni presentati nel seguito sono indipendenti dal motore di Database utilizzato. Questo significa che, in generale, la sola presenza di un driver Odbc affidabile permette l'utilizzo di un qualsiasi prodotto alternativo (come Ms Access, oppure Sybase etc.). Anzi, in generale, OOo dispone di strumenti che permettono anche di evitare l'uso di Odbc con alcuni tipi di Database (Ado, Jdbc etc.). Inoltre, per mia comodità, ho usato il Server MySql in ambiente Windows e solo il Client in Linux, ma le funzionalità sono in generale identiche invertendo i Sistemi Operativi.

2.2 Installazione di OOo

L'installazione di OOo in ambiente Windows è piuttosto semplice e ben descritta nel documento [SETUP_GUIDE.pdf](#) disponibile nella Dir creata scompattando il file ZIP scaricato dal sito, evito quindi di tediarevi oltre.

Tips



Raccomando comunque di eseguire SEMPRE una installazione di rete (**setup -net**) ANCHE se il PC non è collegato in Rete, perché questo permette di avere impostazioni personalizzate per ogni utente attivo sul PC, cosa molto saggia in Windows 2000 e XP (oltre che in Linux). Dopo il “setup -net”, il programma di installazione creerà una cartella di nome `c:\programmi\Openoffice.org1.1.0`. Da questo percorso, **ogni utente del PC** deve lanciare il File “Openoffice.org Setup” e scegliere, quando richiesto, il tipo di installazione “workstation”. Avremo così profili indipendenti per ogni utente abilitato.

2.3 Installazione di MySql

Avere in pochi minuti MySql attivo e funzionante in Windows è molto semplice, anche se l'utente medio alla fine potrebbe avvertire un certo senso di insoddisfazione....

I Files da scaricare dal sito www.mysql.com sono:

mysql-4.X.X-win.zip	il server , per circa 24 Mb (X.X è il numero di versione; quella che useremo è mysql-4.0.16-win.exe)
myODBC-3.X.X.exe	Driver ODBC , per circa 750 Kb (versione attuale 3.51.06)
MySqlcc-X.X.X-win32.zip	MySql Control Center , per circa 3 Mb, un SW che permette la creazione e la manutenzione dei Database MySql (versione attuale 0.9.3)

Tutte le operazioni di installazione vanno fatte da un utente con privilegi di amministratore. Scompattiamo il primo file in una dir a nostra scelta, e lanciamo il classico setup. La cartella di destinazione proposta è `c:\mysql` e va confermata, salvo casi particolari (che ai nostri scopi interessano poco). Vi consiglio di scegliere una installazione “typical”, che carica tutto il necessario senza problemi. Ora riavviate, ed all'apparire del vostro desktop realizzerete che... non è cambiato niente...

Già, perché, anche cercando, non troverete alcuna voce di menù che fa riferimento a MySql, neppure un misero readme.txt, tanto da dubitare che l'installazione sia andata a buon fine. Basta però avviare il task manager e consultare l'elenco dei task attivi: troveremo `mysqld-nt.exe`, che indica il funzionamento corretto del server.

Volendo possiamo poi lanciare il file `C:\mysql\bin\winmysqladmin.exe`, un programma che si installa in memoria, con una piccola icona a forma di semaforo sulla task bar, e che fornisce preziose informazioni sul server attivo.

Tips



Vi consiglio di creare un collegamento (shortcut) al file **winmysqladmin.exe** sul desktop o nel menù avvio, perché torna utile in molte occasioni

La documentazione di MySql comprende un completo manuale in formato HTML (`c:\mysql\docs>manual_toc`), purtroppo in inglese. La parte relativa all'installazione in Windows è molto ben fatta, e spiega in dettaglio le varie opzioni disponibili, quindi chi vuole

approfondire può farlo benissimo senza sforzo. Il manuale tratta bene anche il setup del server in **Linux**, ed a questa sezione rimandiamo quelli di voi che sono amanti del Pinguino. Per i nostri scopi, a questo punto il server è installato e funzionante, e questo basta.

Tecnica

MySQL si installa in Windows 2000 e XP come “servizio” ad esecuzione automatica. Se vogliamo che il server **non** parta in automatico all'avvio, basta richiamare il pannello di controllo e nella sezione “strumenti di amministrazione”, scegliere la voce “servizi”. Avremo una lista dei servizi installati sul PC, quindi doppio click su MySQL e selezionare nella casella a discesa “Tipo di Avvio” la voce “Manuale”. A questo punto MySQL non si avvierà più in automatico alla partenza. All'occorrenza potremo lanciare il server avviando il già citato **C:\mysql\bin\winmysqladmin.exe**.

Bene, il server è attivo e funzionante, ma ora come procediamo ? Potremmo gestire i nostri DB con i potenti tools da linea di comando presenti in c:\mysql\bin (certo dopo una attenta lettura del manuale), e sicuramente c'è chi preferisce questa strada. Noi però siamo pigri, e quindi l'ideale è una bella interfaccia grafica che ci faccia risparmiare tempo. Quindi è ora di installare MySQL Control Center.

Scompattiamo il file **MySQLcc-0.9.3-win32.zip** in una dir qualsiasi, ed ancora una volta lanciamo il setup. Se volete, installate pure i language packs, tra cui l'italiano. Alla fine ritroverete sul Desktop l'icona dell'applicazione. Proviamo ad eseguire il programma e ci troveremo di fronte ad un ambiente a finestre multiple (illustrazione 1). Se l'interfaccia non è in Italiano, nel menù “principale” (quello che comincia per “Console”) scegliamo **Option->General** e selezioniamo “Italian” nell'opzione **Language**. Chiudiamo e riavviamo il Control Center per cambiare lingua.

Ora dobbiamo per prima cosa creare i parametri per il nostro server, quindi, nella Finestra del Console Manager, scegliamo **File -> Nuovo** ottenendo la finestra riportata a fianco.

E' necessario inserire il nome del server (quello che volete...) ed il nome di Host, che può essere l'indirizzo IP (oppure il “nome qualificato”) della macchina dove gira il server. Nell'e-

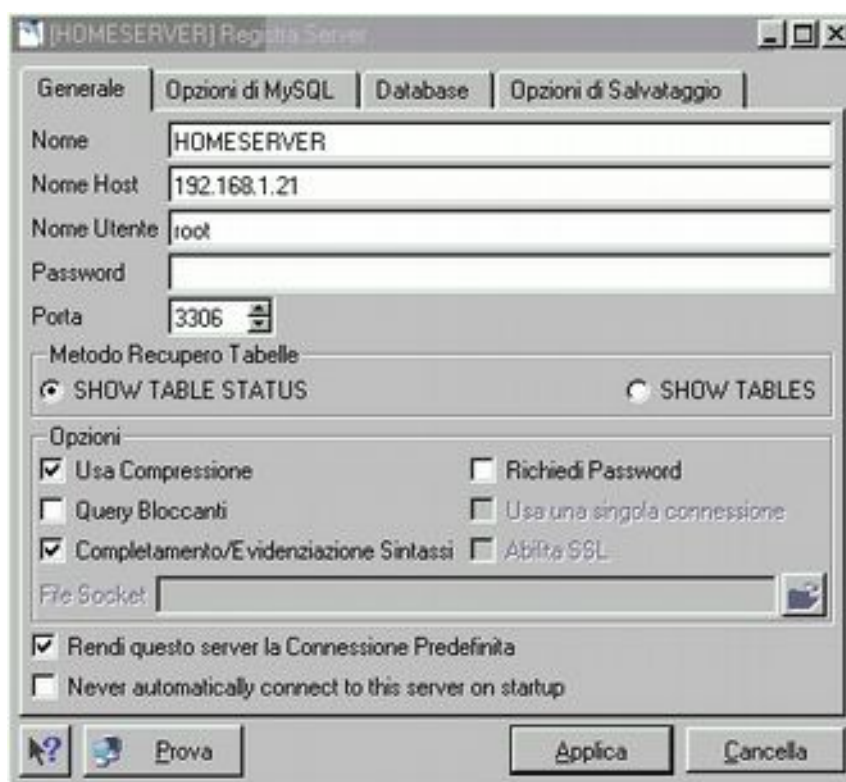


Illustrazione 1 Finestra principale di MySQLcc

sempio, visto che il server è installato sulla stessa macchina, avrei potuto più semplicemente scrivere, nella casella “Nome Host” invece dell'indirizzo Ip, solo “localhost”.

Tips



Il supporto al Tcp/Ip deve essere installato sul Pc perché la cosa funzioni bene. Per quanto MySQL possa anche usare altri sistemi di comunicazione (named pipes), il Tcp/Ip va senz'altro preferito.

Nella installazione standard che abbiamo eseguito, l'amministratore del server MySQL (che si chiama **root**, come in Linux) non ha password, ma possiamo facilmente assegnarne una. Un click sul pulsante “prova” ci permette di testare la connessione, con “applica” chiudiamo il dialogo, ottenendo nel Console Manager, le informazioni sul nostro server. Torneremo dopo sul Console Manager, quindi per ora possiamo chiudere il SW.

Tecnica



Con il Control Center si possono gestire contemporaneamente più server MySQL, anche installati su PC diversi dal proprio. Basta “registrare” il nuovo server con i parametri corretti (nome, indirizzo Ip e password di root). Un altro ottimo prodotto, tra le molte gui grafiche per la gestione dei Db MySQL, è **DbManager Professional** scaricabile dal sito www.db-tools.com.br; è freeware, ma è disponibile solo in inglese.

2.4 Installazione dei Driver ODBC in Windows

La sigla **ODBC** sta per Open Database Connectivity ed indica un layer (strato) software di interfaccia tra le applicazioni ed in motori di Database. In pratica un driver ODBC, proprio come un driver di periferica, si occupa di “tra-ghettare” le informazioni tra ad es. OOo e MySQL. In questo modo si ha la possibilità di accedere ai dati in formati anche molto diversi tra loro senza preoccuparsi eccessivamente delle particolarità implementative di ogni Server Db.

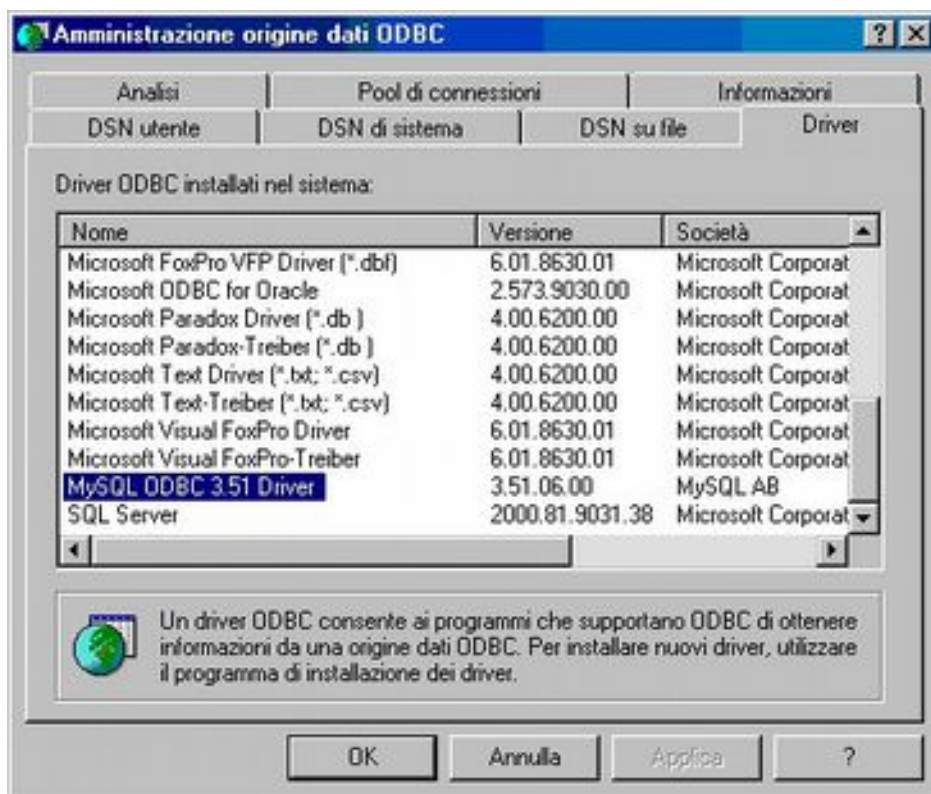



Illustrazione 2 Strumento di gestione driver ODBC per Windows

Esistono driver ODBC per quasi tutti i più importanti motori di Db, e MySQL non fa eccezione. Dobbiamo solo lanciare il file **myODBC-3.51.06.exe**, ed avremo il nostro driver pronto all'azione. Per verificare che tutto sia andato bene, aprendo il Pannello di Controllo -> Strumenti di Amministrazione -> Origine dati (ODBC) -> Driver dovrebbe apparire la finestra riportata sopra:

Tecnica



Quando si parla genericamente di Database, si può fare una gran confusione. Spesso si scambia il **formato** (ad esempio .mdb per Ms Access) con il “**motore**” (che in questo caso si chiama “Jet” e non Access). Infatti “.mdb” è un formato di dati a cui si può accedere tramite il motore di Microsoft “Jet”, anche se Ms Access non è installato sul PC. Allo stesso modo tramite Ms Access possiamo manipolare Dati ad es. in formato MySQL senza usare il Jet, utilizzando invece il driver MyODBC. Nel caso di Ms Access la confusione sorge dal fatto che il programma usa il formato “.mdb” anche per le proprie necessità (maschere, report etc.) oltre che per i dati. Altra importante differenza da considerare è quella tra i “server” Db veri e propri ed i “motori” Db generici. Un server Db di solito parte come un Servizio residente in memoria e rimane in ascolto su una porta specifica (la 3306 per MySQL). Un motore Db (come Jet) è un insieme di DLL (librerie) richiamate all'occorrenza dall'applicazione che deve far uso di dati.

2.5 Installazione del driver ODBC in Linux

In **Linux** il driver è incluso in molte distribuzioni, e quindi può essere installato semplicemente usando il tool specifico di ogni distribuzione (YAST per SUSE, Control Center per Mandrake etc.). Se non lo trovate, o desiderate l'ultima versione, su www.mysql.com è possibile scaricare un comodo RPM. Per il corretto funzionamento, è però necessario installare il pacchetto LinuxODBC. I passi da fare, quindi, in SUSE 9.0, sono:

1. autenticarsi come root, e con YAST installare i pacchetti **unixODBC**, **qt3-unixODBC**, **unixODBC-gui-qt**
2. scaricare da MySQL.com l'ultima versione del driver **MyODBC-3.51.06-1.i386.rpm**, ed installarla
3. Selezionare dal menu la voce “Sistema->Configuration->unixODBC Configuration”
4. Scegliere dal Tab “Drivers” la voce “Add” e configurare il driver come in figura

Notate che l'RPM di MySQL carica il **driver** in **/usr/local/lib/**, mentre il **setup**, che permette di usare l'interfaccia grafica per i DSN, è una aggiunta di unixODBC, e si trova in **/usr/lib/unixODBC/**. Sarebbe anche possibile installare il driver modificando con un editor il file `odbc.ini`, ma questo direi che esula dallo scopo di questa documentazione.



Illustrazione 3 Gestione ODBC driver sotto Linux

3 Il mondo dei Database

3.1 Primi passi....

Questa sezione serve ad introdurre alcuni concetti indispensabili per le persone che non sono molto esperte di Database. Se la cosa vi annoia, saltatela pure.

La traduzione letterale di Database è “Base di Dati”, e serve almeno per farsi un'idea dell'argomento che andiamo ad affrontare. Più semplicemente, potremo indicare un Database con la generica parola “**archivio**”. Ora, un Archivio è una idea semplice, nulla di astratto, abbiamo archivi per le mani molto più spesso di quanto si creda. Basta un elenco telefonico per capire cos'è un archivio, basta guardare la rubrica del cellulare....

Potremmo definire un Archivio come un insieme di informazioni, organizzate in una struttura logica, spesso ordinate secondo una propria necessità, con uno o più caratteristiche in comune. Guardiamo appunto una rubrica telefonica: la caratteristica in comune è “archivio dei numeri della rubrica di Giuseppe”, la struttura comprende due informazioni (il nome ed il numero), l'ordine è di solito quello alfabetico.

Nome	Numero
Carla	340 1234
Elisa	06 54678
Giuditta	02 987456

Tabella 1 Esempio di rubrica semplice

Questa, nel gergo dei Database, è una **Tabella (Table)**, che comprende due **Campi (Fields)**, “Nome” e “Numero”. La Tabella comprende tre **Righe** chiamate anche **Schede (Rows, Records)**. Un **Database** è di solito un **insieme di Tabelle** che possono essere messe in **relazione** tra loro tramite connessioni logiche (presenza in più tabelle della stessa informazione). Da questa definizione deriva il nome di Database Relazionale (RDBMS, Relational Database Management System) assegnato alla tipologia di prodotti che stiamo esaminando. Attenzione, spesso si indica con lo stesso nome (Database) sia il “motore” cioè il software che mi permette di gestire le informazioni, sia le informazioni stesse. Questo, in generale, non è corretto.

Tecnica



Esiste una “definizione” molto precisa dei database relazionali. Questo significa che un Database, per poter essere correttamente definito “relazionale”, deve soddisfare una serie di regole che qui sarebbe lungo ed improduttivo descrivere. Secondo alcuni MySQL, siccome non possiede TUTTE queste caratteristiche, NON E' a rigore un Database relazionale. Questo però a noi interessa davvero poco....

3.2 Progettiamo un Database

Abbiamo quindi definito la **Tabella** come l'**unità logica di un Database**. Ovviamente possiamo avere Database composti da una sola Tabella, ma direi che sono casi particolari. Ora siamo perciò di fronte al nostro amato PC e vogliamo cominciare a creare il nostro primo Database; la prima cosa da fare è quindi... un bel passo indietro.

Infatti l'approccio più sbagliato che possa esistere è partire con la struttura di un Db senza averci prima ben riflettuto... magari davanti ad un bel foglio bianco e con una cara e vecchia penna in mano. Voglio dire che in generale è fondamentale “progettare” il Database PRIMA di mettere la manina sulla tastiera, ed allora il saggio consiglia che bisogna porsi in anticipo le seguenti domande:

1. di quante e quali unità logiche (tabelle) deve essere composto il mio Db ?
2. per ogni tabella, quali informazioni (campi) devo comprendere ?
3. per ogni campo di ogni tabella, che tipo di informazioni devo archiviare ?
4. quali sono i campi su cui sarà necessario eseguire un ordinamento ?
5. che relazioni ci sono tra le varie tabelle ?
6. che “operazioni” desidero eseguire su ogni tabella ?

Avrete quindi capito che, siccome “chi ben comincia etc. etc.”, rispondere a queste domande all'inizio del lavoro ci eviterà problemi nel seguito. Questo non significa che in corso d'opera non si potranno fare variazioni, ma cambiare la struttura di Db complessi quando già è iniziato il caricamento dei dati è davvero complicato. Inoltre, non prevedere qualche piccolo dettaglio può portare a risultati pericolosi, vi ricordate l'affare dell'anno 2000 ?

3.3 La Tabella

Una Tabella, abbiamo visto, è composta da Campi. Possiamo pensare al **Campo** come l'installazione di colonna di una lista, ma in realtà è molto di più di una descrizione. In genere, infatti un Campo incorpora numerose “proprietà”, cioè “caratteristiche” che una volta impostate, determineranno in modo preciso il tipo di informazione che quel Campo può contenere.

Esaminiamo un po' più in dettaglio queste caratteristiche:

- Il “**nome**” è in genere la descrizione dell'informazione (ad es. “numero di Telefono”). Non ci sono molti vincoli sul nome, vi consiglio però di non sceglierlo troppo lungo, ma allo stesso tempo abbastanza esplicativo. Nel nostro caso andrebbe ad es. bene *NumTel*. Se il Db comprende più Tabelle con lo stesso campo (se ho, ad esempio, la Tabella Clienti e quella Fornitori) e non sono in relazione tra loro, può essere utile anteporre al nome del campo un indicativo della Tabella, ad es. “CNumTel” e “FNumTel”. In genere si possono usare gli “underscore” (Num_Tel) ma io lo trovo poco pratico.
- Il “**tipo**” indica la tipologia della informazione da archiviare nel campo. Le tre categorie principali sono **stringa**, **numero** e **data/ora**. All'interno di ogni categoria possiamo scegliere molte ulteriori tipologie, ed inoltre esistono “tipi” particolari (come “binario” o “timestamp”) non riconducibili facilmente agli altri.

- La “**lunghezza**” misura l'occupazione fisica massima (in byte, di solito) che vogliamo assegnare all'informazione contenuta nel campo. Per le stringhe si indicano i caratteri (ad es max 30 caratteri per il campo “nome”), per i valori numerici il discorso è un po' più complesso (lo vedremo in seguito), per data/ora esistono formati standard con diverse occupazioni di memoria a seconda dell'intervallo dei valori che si vuole comprendere.
- “**Ammetti null**” è un segnalatore che indica se è possibile archiviare nel campo anche valori nulli (a prima vista può sembrare inutile, invece la scelta è assai importante)
- “**Valore predefinito**” è il valore che dovrà assumere in automatico il campo all'immissione di una nuova riga. Ad esempio, se la maggior parte dei nostri clienti è della provincia di Milano, potremmo volere che il campo Provincia sia riempito dalla stringa “MI”. Se non si specifica niente, il campo assume il valore null. Per i campi numerici è particolarmente importante assegnare come valore predefinito lo zero (come vedremo tra poco).
- “**Chiave**” oppure “**Indice**” stabilisce se il campo debba essere indicizzato e se l'indice è univoco (vedere il paragrafo successivo “Indici”)

Queste *proprietà del campo* si ritrovano quasi identiche in praticamente tutte le varietà di Db Server. Ogni prodotto, poi, implementa “aggiunte” magari non standard, ma egualmente importanti. Ora descriveremo più in dettaglio come MySQL gestisce le tipologie di informazione (i tipi di campo) che possono interessarci in funzione dei nostri scopi, rimandando al manuale dell'applicazione chi volesse approfondire l'argomento.

Tips

E' possibile cambiare la struttura delle Tabelle di un Db MySQL anche DIRETTAMENTE da OpenOffice.org, anche se vedremo che l'operazione non è particolarmente conveniente rispetto all'uso di MySQL Control Center



3.4 Campi di tipo Stringa

Una **Stringa** è in generale una informazione alfanumerica di lunghezza variabile (un nome, una via, un titolo etc.). Questa informazione può essere archiviata in due modi:

1. posso stabilire un numero massimo di caratteri, e riservare sempre un numero di byte fisici equivalenti nel mio archivio (tipo **char**)
2. posso stabilire un numero massimo di caratteri, ma archiviare solo quelli effettivamente digitati dall'utente (tipo **varchar**)

Nel primo caso saranno aggiunti tanti spazi quanto servono a raggiungere la lunghezza specificata, nel secondo invece lo spazio occupato è in funzione dei caratteri digitati. Ad esempio supponiamo di avere il campo “nome” di lunghezza max 20 caratteri; se scrivo nel campo la stringa “Carlo”, nel primo caso avrò usato 20 byte, nel secondo solo 5.

Allora, vi chiederete, visto che è più conveniente dal lato dell'occupazione di spazio, usare il tipo “varchar”, il “char” che ci sta a fare ? In effetti il “char” si usa quando l'informazione è formata da pochi caratteri e magari deve essere anche indicizzata. Quindi il “char”, per esempio, va bene per archiviare il campo CAP di un indirizzo, oppure un Codice Cliente di 5 caratteri (che sicuramente andrà indicizzato). Il fatto è che i campi a lunghezza fissa come “char”

sono più veloci da manipolare da parte del motore db, e quindi anche più indicati nella creazione di indici.

Una “n” davanti al tipo di campo (“**nchar**”, “**nvarchar**”) significa che il campo usa il set di caratteri internazionali predefinito, ma in MySQL è inutile, il software usa già di default questo set di caratteri.

Tips

La lunghezza di un campo stringa andrebbe attentamente valutata; da una parte, se si imposta a troppo pochi caratteri, si rischia di non poter archiviare informazioni lunghe, dall'altra ogni carattere in più significa spazio sprecato e velocità abbassata. La mia regola (ovviamente opinabile) è che se la lunghezza è inferiore a 10 caratteri uso “char”, altrimenti “varchar”. Per i campi di denominazione (nome e cognome, nome di aziende etc.) direi che 50 caratteri sono sufficienti. Negli altri casi un po' di sperimentazione non guasta.



3.5 Campi di tipo Numerico

I campi di tipo numerico si dividono in due grandi sotto categorie: **interi** e **decimali**. In MySQL per questi campi si può indicare il numero massimo di caratteri di visualizzazione (parametro **M**) e, per i decimali, il numero di cifre dopo il punto (parametro **D**). I parametri M e D hanno valori standard per le varie tipologie, ma possono essere variati dall'utente in determinati intervalli.

I campi di tipo **numerico intero**, a seconda dell'intervallo di valori che possono contenere, si dividono in **TinyInt**, **SmallInt**, **MediumInt**, **Int**, **BigInt**. Ad esempio in MySQL un campo **Int** può contenere da -2147483648 a 2147483647. Non tutti i motori di Db hanno tutte queste categorie, ma il concetto è proprio identico.

Analogamente, i campi di tipo **numerico decimale** in MySQL si possono definire **Float**, **Double**, **Decimal**. In particolare il tipo **Decimal** è indicato per la manipolazione di valori “valutari”, quello che in altri motori di Db viene definito campo “Money” o “Currency”.

Tips

Sarebbe sempre meglio controllare bene il range dei valori memorizzabile nei tipi di dati numerici del motore di database che vogliamo usare. Ad esempio il tipo “Intero” di Access (Jet) può contenere valori tra -32728 e 32767, e quindi equivale a “smallint” di MySQL. Allo stesso modo, il tipo “currency” o “valuta” di Access corrisponde al tipo “Decimal” con M=19 e D=4 di MySQL. La differenza è che mentre il campo di Access occupa sempre 8 Byte, quello di MySQL occupa **M** Byte (nel nostro caso quindi 19). Io per i valori in **euro** in MySQL uso **Decimal(16,4)** cioè 16 cifre di cui quattro decimali. Se si vuole arrotondare i calcoli a due cifre decimali, va bene anche **Decimal(14,2)**.



3.6 Campi di Tipo Data/Ora

Tutti i motori di Db hanno tipologie specifiche di campi per la manipolazione di date ed orari. MySQL usa **Date**, **DateTime**, **Time**, **Year**, dal significato piuttosto intuitivo. Notate che il range di “Date” va dal 01-01-1000 al 31-12-9999. Siccome quasi sempre le date vengono me-

morizzate nel formato AAAAMMGG e visualizzate invece nel formato americano MM-GG-AAAA, bisogna fare molta attenzione ai formati di input che si assegnano.

3.7 Campi particolari: Intero ad incremento automatico

Come vedremo nel paragrafo dedicato agli indici, può essere a volte utile che il valore di un campo sia assegnato in automatico dal motore Db, secondo una progressione numerica. Questo campo può, ad esempio, rappresentare un Codice univoco da assegnare alla scheda senza che noi dobbiamo preoccuparci di fare niente. In MySQL possiamo usare un campo “Int” con la caratteristica “**auto_increment**” settata.

Tips

In Ms Access questo tipo di campo viene indicato come “contatore” e si può stabilire se il valore assegnato sia progressivo oppure casuale. Direi che questa ultima scelta è totalmente ininfluente riguardo tutti gli aspetti da considerare per una buona progettazione del Db.

3.8 Campi particolari : Timestamp

Il **Timestamp** è un tipo di campo Data/Ora aggiornato dal sistema che contiene il riferimento all'ultima modifica ai dati contenuti nella riga. In ambienti multiutente è particolarmente utile per gestire le modifiche concorrenti sulla stessa tabella. Siccome alcuni “front end” per database funzionano male se in ogni tabella non è presente almeno un campo Timestamp, noi lo aggiungeremo sempre, tanto non costa niente. Ad esempio, Ms Access si rifiuta di funzionare bene su Tabelle MySQL che non hanno il Timestamp. Questo, in funzione di OOo potrebbe essere ininfluente, ma certamente non vogliamo che il nostro Db in futuro non sia leggibile da qualsiasi applicazione, giusto ?

3.9 Indici

Alla domanda: “data una Tabella, quali campi è necessario indicizzare?” la risposta è “dipende...”. Cioè in pratica non si risponde, perché non c'è una regola generale universalmente condivisa. Ognuno, in base alla sua esperienza, decide in modo autonomo. Inoltre gestire poche centinaia oppure alcune migliaia di Record sono cose ben diverse, come pure usare da soli il proprio Db oppure far parte di una rete con decine di macchine. Questo non significa che non vi posso dare qualche buon consiglio....

Innanzitutto cerchiamo di capire **cos'è un indice**. Una volta definita la struttura della nostra tabella, ogni volta che aggiungiamo una riga, il nostro motore di Db “accoda” sul disco le informazioni all'archivio aperto. Tornando al nostro semplice esempio della Rubrica del cellulare, quando aggiungiamo i nomi, di certo non lo facciamo in ordine alfabetico; ma comunque, nella consultazione, l'ordine alfabetico ci è molto comodo. Quindi il software del nostro cellulare crea un indice sul campo nome, in modo da poterci fornire le informazioni nell'ordine più logico.

Quindi un indice è, in generale, un “ordinamento” creato su uno o più campi in modo che il reperimento delle informazioni contenute nell'indice stesso sia molto rapido. Non è il caso di

spiegare in dettaglio come questo avvenga a livello di programmazione, accontentiamoci dei risultati. Allora....

Consiglio 1 : i motori di Db funzionano meglio se in ogni tabella esiste un indice univoco, cioè un valore specifico e diverso per ogni riga. Questo campo di solito si chiama **Chiave Primaria (Primary Key)**. Possiamo perciò definire la “chiave primaria” come *un campo che assume un valore diverso per ogni riga della tabella, e quindi identifica univocamente la riga stessa*.

Consiglio 2 : i campi Interi ad incremento automatico sono i candidati ideali per le Chiavi Primarie. Infatti i motori di Db hanno prestazioni ottimali sugli indici numerici interi.

Consiglio 3 : create un indice sui campi che desiderate siano ordinati sui report di stampa. Se ho un archivio clienti, e mi serve una stampa in ordine alfabetico, dovrò creare un indice sulla Denominazione.

Consiglio 4 : create un indice sui campi che userete per cercare delle informazioni. Se ho un elenco di libri, una buona idea è creare un indice sul campo “autore” (cercherò sicuramente i libri per autore)

Consiglio 5 : se avete due tabelle correlate (messe in relazione) tra loro, create indici sui campi comuni (questo, se non vi è chiaro, lo capirete più avanti)

Consiglio 6 : troppi indici rallentano il sistema. Molti utenti ancora inesperti sono portati ad indicizzare tutto l'indicizzabile. Sbagliato. Ogni operazione di modifica dei dati, comporta, se il campo è indicizzato, anche la modifica degli indici. Inoltre gli indici occupano memoria e spazio su disco. Quindi, mi raccomando, parsimonia.

Consiglio 7 : l'efficacia di un indice è inversamente proporzionale alla lunghezza del campo. Più il campo è piccolo, più l'indice è efficiente. Se avete, nella vostra tabella, un campo “note” di 200 caratteri, creare un indice non è una buona idea.

Consiglio 8 : evitate gli indici su campi che assumono solo pochi valori. Se ho un campo che può contenere ad esempio solo “S” per Si e “N” per No, un indice peggiora le prestazioni e non serve a niente.

Consiglio 9 : i campi scelti come indice non dovrebbero contenere il valore “null”, perché i motori Db gestiscono male questa situazione. Ovviamente neppure la chiave primaria deve ammettere valori “null”.

Ora è il momento di passare alla pratica, prima di cadere nelle braccia di morfeo....

3.10 Il nostro Database di esempio

Per meglio illustrare i concetti che andremo ad introdurre, abbiamo bisogno di un piccolo Archivio di esempio, non troppo complesso ma nemmeno banale. Dopo una lunga (e sofferta) riflessione, abbiamo deciso che gestiremo una “**Mediateca**”. Ora vi chiederete che sarà mai questo oggetto sconosciuto: bene, è un impasto di Videoteca, Biblioteca, Discoteca, Emeroteca etc. In pratica tutto quello che finisce in -teca ...

Vogliamo, cioè, creare un Archivio che ci permetta di gestire in modo flessibile qualsiasi tipo di “Media” desideriamo catalogare. Inoltre visto che alle nostre cose ci teniamo, abbiamo stabilito che potremmo anche usare un piccolo promemoria per sapere a chi e quando abbiamo (sconsideratamente) fatto un prestito.

Trattandosi di un esempio didattico, non andremo troppo per il sottile, quindi guru dei Db trattenete commenti inopportuni. Chi vuole poi potrà studiare e migliorare....

Allora, tabelle necessarie:

TbMedia -> l'archivio principale dei nostri Media

Campo	Descrizione
Id	Chiave primaria – Intero ad incremento automatico
Desc	Descrizione
TipoSupp	Tipo del supporto di memorizzazione (CD, DVD, Rivista, File, etc.)
Argom	Argomento o classificazione
Ubicazione	Ubicazione (Scaffale, Scatolo, Num Rivista, Percorso su Hd etc.)
Prezzo	Prezzo (di acquisto, di vendita, di prestito...), forse non necessario, ma sicuramente didattico
Note	Annotazioni libere

Tabella 2 *Tabella dei media*

Bene, dobbiamo ora definire il tipo di campo da assegnare ad ogni informazione della nostra Tabella. La cosa più immediata, ad esempio per il campo “TipoSupp”, sarebbe ovviamente “stringa”, magari di lunghezza 20 caratteri. Così, ad esempio, una parte del nostro archivio potrebbe essere :

Desc	TipoSupp	Argom
Pearl Jam - Binaural	CD Audio	Rock
Fromm – Avere o Essere ?	Libro	Filosofia
Harry Potter e la camera dei segreti	DVD Film	Fantasy
Peter Gabriel – Growing Up Live	DVD Musicale	Rock
Condividere risorse con Samba 3	Rivista	Linux / Samba
Rossini – 10 Ouvertures – Chailly	CD Audio	Classica

Tabella 3 *Esempio di archivio dei media*

Siccome ritengo che prima di progettare qualunque Database sia opportuno preparare uno schema “reale” di quello che la Tabella dovrà contenere, che cosa possono suggerirci queste “righe” di esempio ?

1. I campi “TipoSupp” e “Argom” contengono stringhe ripetitive, cioè molte righe avranno ad esempio in TipoSupp la stringa “DVD Musicale”; è uno spreco enorme di spazio.
2. Se scriviamo “manualmente” in “TipoSupp” le varie classificazioni, sarà facile sbagliare e quindi inserire ad esempio per errore “DVD Musocale”; se devo poi selezionare tutti i DVD Musicali, quella riga sarà scartata.

3. Come facciamo a ricordare a memoria tutte le classificazioni usate ? Se la mediateca è ampia, è quasi impossibile.

Allora come fare ? Semplice, basta creare un'altra Tabella da usare come “classificatore”, del tipo:

TbSupporti -> Classificazione dei Media per Tipo di Supporto

Identificatore	Descrizione
1	DVD Film
2	DVD Musicale
3	CD Audio
4	Rivista
5	Libro

Tabella 4 Tabella per la classificazione dei supporti dei media

Così il nostro Archivio apparirà in questa forma :

Desc	TipoSupp	Argom
Pearl Jam - Binaural	3	Rock
Fromm – Avere o Essere ?	5	Filosofia
Harry Potter e la camera dei segreti	1	Fantasy
Peter Gabriel – Growing Up Live	2	Rock
Condividere risorse con Samba 3	4	Linux / Samba
Rossini – 10 Ouvertures – Chailly	3	Classica

Tabella 5 Archivio con le relazioni alla tabella dei classificazione TbSupporti

I vantaggi di questa organizzazione sono subito evidenti :

1. risparmio di spazio (numeri interi al posto di stringhe)
2. velocità di immissione (devo solo selezionare, nel campo TipoSupp, un valore da una Tabella)
3. velocità operativa (un indice su “TipoSupp” sarà molto più efficiente)
4. riduzione della possibilità di errore

Ho dunque eseguito, magari senza saperlo, quella che si chiama una “**normalizzazione**” del Database. Tutto quanto spiegato vale anche per il campo “Argom”, ed in generale per tutti i campi destinati a contenere informazioni riconducibili ad un insieme ben definito e non troppo ampio. Nella fase operativa vedremo come mettere in “relazione” le due tabelle, e quindi sfruttare al meglio il Db normalizzato.

Tips

In questo esempio abbiamo usato un campo di tipo Intero per collegare le informazioni contenute nelle due Tabelle. Questo non è obbligatorio. Ci sono molti casi in cui i campi collegati sono di tipo stringa. La regola è che comunque tutti i campi usati nelle relazioni DEVONO essere indicizzati (come vedremo più avanti) e quindi gli Interi sono più efficienti



Tecnica

Quella illustrata è una relazione tra Tabelle di tipo “*uno a molti*”, cioè un singolo valore di una riga di TbSupporti (“3”, equivalente a “CD Audio”) appare in molte righe di TbMedia. Altri tipi di relazione sono “uno a uno” e “molti a molti”. Comprendere come funzionano le relazioni tra Tabelle nei Database può sembrare complicato, ma alla fine i risultati sono assai soddisfacenti. Nel nostro caso il campo “TipoSupp” di TbMedia fa riferimento alla chiave primaria di un'altra Tabella, cioè TbSupporti, quindi “TipoSupp” è una “**chiave esterna**” (“**foreign key**”) per TbMedia. Una delle regole che indicano se un motore di Db può definirsi relazionale o meno riguarda proprio le chiavi esterne. Ne ripareremo più avanti nell'esaminare la gestione dell'integrità referenziale con MySQL.



4 Hands On...

4.1 Struttura e Generazione del Database con MySqlCC

Il primo passo è la creazione del nostro Database. Lanciamo MySqlCC (il Control Center). Se l'installazione è andata a buon fine, e se il servizio server di MySQL è attivo sul nostro PC, dopo qualche secondo di “riflessione”, dovremmo avere a video una finestra simile a quella riportata nell'illustrazione 4:

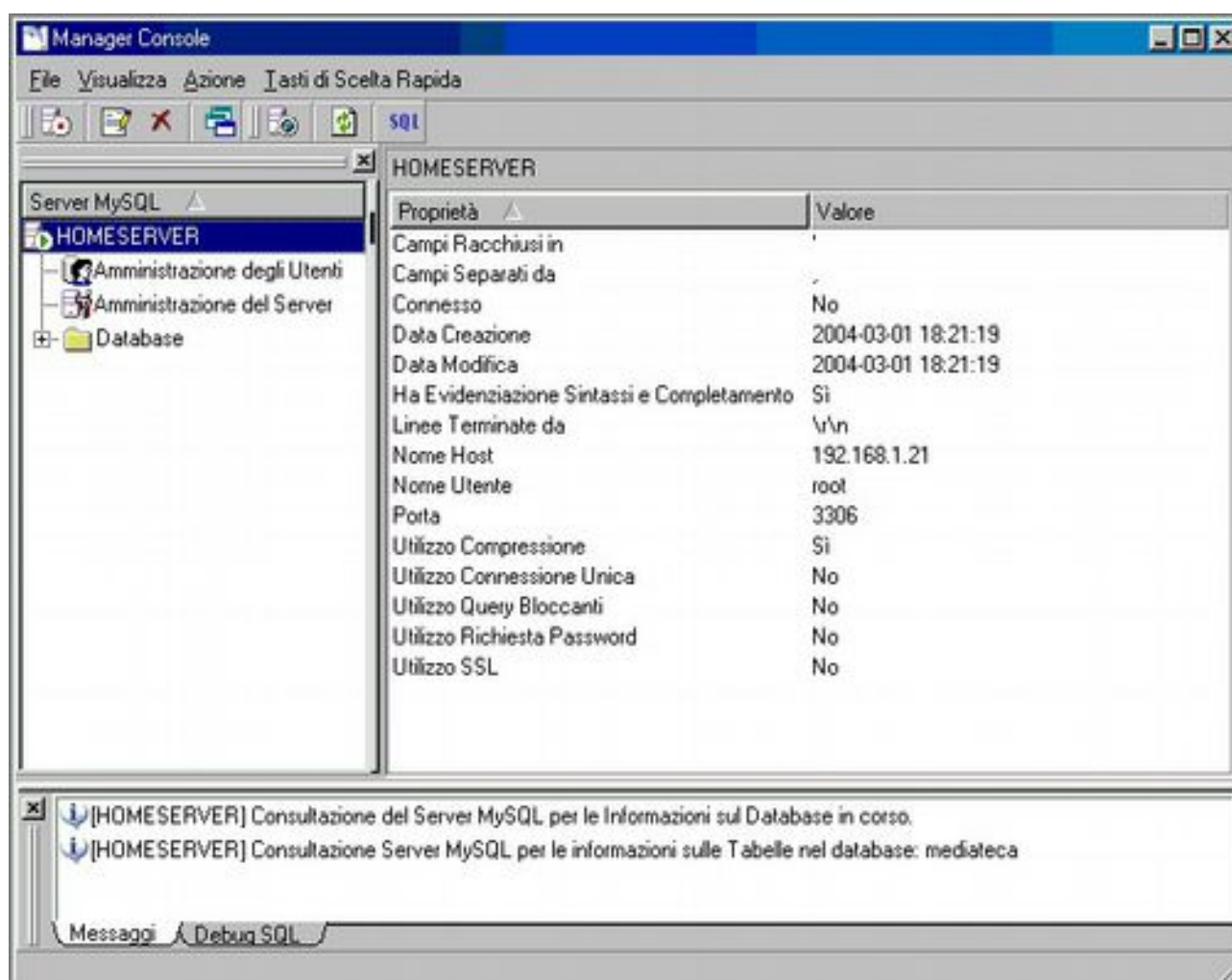


Illustrazione 4 Il console manager di Mysqlcc

Ogni Server MySQL (nel senso sempre di servizio attivo su un computer) può gestire molti Database diversi. Per creare quello che useremo per i nostri esempi, basta fare click sulla voce **Database** della finestra sinistra e, dal menù contestuale attivato dal tasto destro del mouse,

scegliere “**Nuovo Database**”. A questo punto sarà sufficiente inserire il nome che abbiamo scelto, cioè “**mediateca**”, per avere una situazione di questo tipo:



Illustrazione 5 Scelta del database

Ancora il tasto destro, scegliamo “**connetti**”, per avere :



Illustrazione 6 Connessione al database

Tasto destro, “**nuova tabella**” et voilà.... quella che vedete in figura è la finestra di modifica della struttura di una Tabella, già riempita dei campi che useremo nel nostro esempio. Diamo un'occhiata più da vicino. La parte centrale permette la compilazione delle informazioni basilari per ogni campo, cioè il “**nome**”, il segnalatore di “**permetti null**” ed il “**tipo di dato**”. Più in basso possiamo definire altre proprietà del campo. Ricordando cosa abbiamo detto a proposito della struttura dei Db, per la tabella principale, illustrazione 7 (**tbmedia**) sceglieremo come identificatore il Campo **MId** (Media Identifier), di tipo Intero (**Int**), che ha la lunghezza standard di **undici byte**, è ad incremento automatico (“**auto_increment**”), ed inoltre può contenere solo valori positivi (“**unsigned**”). La parte relativa alle proprietà cambia a seconda del tipo di campo scelto. Ad esempio per i campi di tipo “**varchar**”, possiamo definire solo la lunghezza ed il valore predefinito.

Tips

La proprietà “**valore predefinito**” è importante. Se definiamo un campo numerico qualsiasi, non è saggio, per i motivi spiegati in precedenza, permettere l'archiviazione di valori nulli. Nel caso di aggiunta di una riga, però, dovremmo manualmente scrivere il valore 0 nel campo numerico. Basta quindi assegnare al campo il valore predefinito di 0 per toglierci il fastidio. *Quindi ai campi numerici è sempre opportuno assegnare il valore iniziale uguale a*



zero. I Campi stringa possono anche “ammettere” il null, a meno che non si tratti di un indice, dove il null è sconsigliato. Anche in questo caso se il “null” non è ammesso, può essere comodo assegnare un valore iniziale, magari uno spazio “ ”.

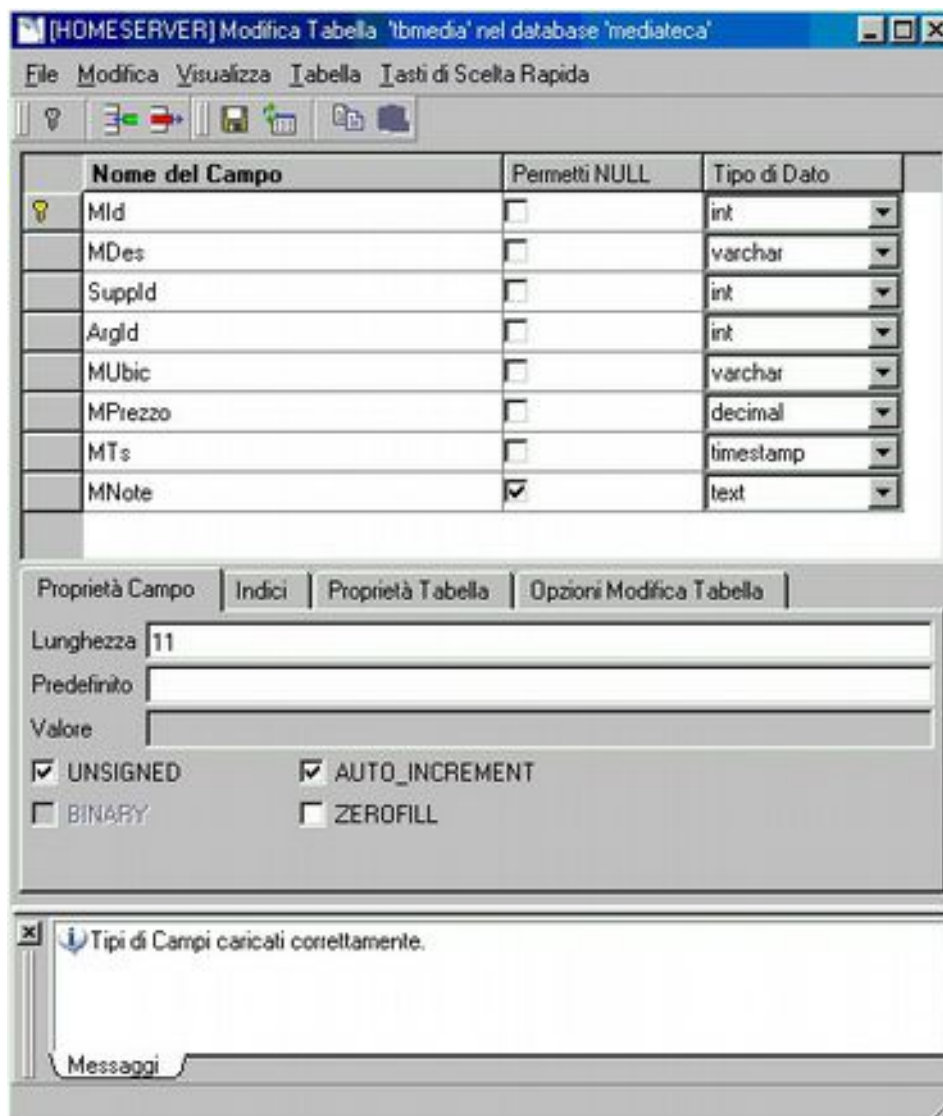


Illustrazione 7 Proprietà dei campi della tabella tbmedia

Per i Campi di tipo Decimal, la lunghezza va impostata secondo la forma “M,D”, dove M è il numero complessivo di cifre desiderato e D il numero di Decimali. Il campo “**Mprezzo**” è stato definito come “14,2”. Ho aggiunto alla fine un campo “**Note**” di tipo “**text**”, cioè una variante di “varchar” di lunghezza massima di 65.535 caratteri, che può contenere sia testo in chiaro che valori binari, infatti è equivalente al tipo “blob” (che sta per “Binary Long Object”).

Tecnica

Per chi non ha mai approfondito l'argomento, il concetto di “**null**” può non essere chiaro. In effetti “null” significa “vuoto” cioè letteralmente “senza alcun valore di alcun tipo”, quindi “privo di valore”. Questo significa che un valore “null” è cosa diversa dallo zero, ed anche



diverso dalla stringa vuota, cioè “”. Per definizione un valore “null” non è confrontabile, e non può essere usato in operazioni aritmetiche. Perciò, ad esempio $0 + \text{null}$ è scorretto (può dare un errore di sistema o ancora un null), e neppure ha senso dire che “pippo” viene prima o dopo “null”. Quindi occhio, amici miei, che questo è un punto fondamentale per ottenere, alla fine, dal nostro archivio valori sensati.

Per scegliere una **chiave primaria**, selezioniamo il campo MId, e premiamo il pulsantino a forma di chiave sulla barra dei pulsanti. Dovremmo ora creare gli altri indici da associare alla tabella, ma è necessario prima salvare ed uscire dalla maschera di modifica, perché un bug di questa versione di MySQLCC non permette l'aggiunta di indici senza salvare prima la tabella. Quindi salviamo col pulsante a forma di disco, assegnando il nome “tbmedia”, e subito dopo rientriamo in modifica sulla tabella. Selezionando il tab “indici” in basso avremo:



Illustrazione 8 Definizione chiave primaria

Sulla destra notiamo il nome dell'indice, e nella colonna “campi utilizzati” i campi che formano l'indice stesso (possiamo ovviamente usare più campi). La chiave primaria è sempre di tipo “unique”, cioè unica, quindi non possiamo cambiare gli attributi. Per aggiungere un nuovo indice basta premere il pulsante verde con il segno “+”. Dobbiamo quindi assegnare un nome all'indice, e quindi spostare dalla colonna “campi disponibili” a quella “campi utilizzati” i nomi di campo che serviranno all'ordinamento. In questo caso stiamo creando un indice sul campo Mdes (la descrizione).



Illustrazione 9 Definizione dell'indice

Salviamo il risultato per creare l'indice. Con lo stesso sistema possiamo aggiungere tutti gli indici desiderati, cercando di seguire i consigli che abbiamo già visto. Un indice “unique” significa che il valore del campo sarà obbligatoriamente diverso per ogni scheda del nostro archivio. La trattazione degli indici “full text” esula dallo scopo di questo documento.

Per la tabella principale del nostro Db, “**tbmedia**”, alla fine dovremmo avere questo risultato nella finestra di MySQLCC:

Field	Type	Null	Key	Default	Extra
Mid	int(11) unsigned		PRI		auto_increment
MDes	varchar(100)		MUL		
Suppld	int(10) unsigned		MUL	0	
Argld	int(10) unsigned		MUL	0	
MUbic	varchar(100)	YES			
MPrezzo	decimal(14,2) unsigr			0.00	
MTs	datetime	YES			
MNote	text	YES			

Illustrazione 10 Risultato finale tabella TbMedia

Quindi abbiamo scelto, come **campi**:

- Mid - di tipo intero positivo ad incremento automatico, come chiave primaria
- MDes – Stringa di max 100 caratteri, come descrizione del “supporto mediatico”
- Suppld – un intero positivo che si collega alla Tabella dei “Tipi di Supporto”
- Argld – un intero positivo che si collega alla Tabella degli “Argomenti”
- Mubic – stringa di max 100 caratteri che potrà contenere l'ubicazione del supporto
- MPrezzo – campo di tipo Decimal, per la gestione di un importo in valuta
- MNote – campo testo / binario da usare ad esempio per annotazioni (max 65535 car)

Ed invece, come **indici** :

- Mid – Chiave primaria
- MDes, Suppld, Argld come indici aggiuntivi

Ora che la nostra tabella principale è pronta, vi risparmio il dettaglio sulla creazione delle altre, riportando solo la struttura che dovrete inserire. Notate anche la presenza, in ogni tabella, di un campo di tipo “Timestamp”, per i motivi esposti in precedenza.

Tabella **TbSupporti** – Descrizione dei Supporti

Field	Type	Null	Key	Default	Extra
Suppld	int(10) unsigned		PRI		auto_increment
SuppDes	varchar(30)		MUL		
SuppTs	timestamp(14)	YES			

Illustrazione 11 Tabella TbSupporti

Tabella **TbArgomenti** – Argomenti contenuti nell'archivio

Field	Type	Null	Key	Default	Extra
ArgId	int(10) unsigned		PRI		auto_increment
ArgDes	varchar(30)		MUL		
ArgTs	timestamp(14)	YES			

Illustrazione 12 Tabella TbArgomenti

Tabella **TbUtenti** – Archivio degli utenti dei prestiti

Field	Type	Null	Key	Default	Extra
UId	int(10) unsigned		PRI		auto_increment
UIDen	varchar(50)		MUL		
UITel	varchar(30)				
UICit	varchar(100)				
UITs	timestamp(14)	YES			

Illustrazione 13 Tabella TbUtenti

Tabella **TbPrestiti** – Lista dei prestiti e delle restituzioni

Field	Type	Null	Key	Default	Extra
PreId	int(10) unsigned		PRI		auto_increment
PreData	date		MUL	0000-00-00	
UId	int(11)		MUL		0
MId	int(11)		MUL		0
PreDataR	date			0000-00-00	
PreTs	timestamp(14)	YES			

Illustrazione 14 Tabella TbPrestiti

4.2 Dettagli....

Sarà meglio, prima di continuare, spiegare un po' meglio le scelte fatte nella creazione della struttura degli archivi. Per prima cosa notate la modalità con cui MySQLCC mostra la struttura di una Tabella. Accanto al nome di campo, una “chiave” indica la *Chiave Primaria* (Primary Key), un pallino gli altri campi indicizzati (di colore verde per gli indici multipli, rosso per gli indici univoci). La colonna “*Type*” indica il tipo di campo; “*Null*” informa sulla possibilità di memorizzare valori nulli nel campo. “*Key*” indica il tipo di Indice, “*Default*” il valore predefinito, “*Extra*” altre informazioni sul campo.

TbMedia è la nostra tabella principale; possiede un campo “*MId*” (intero ad incremento automatico) che rappresenta la **chiave primaria**; i campi “*MDes*” (Descrizione) e “*MUbic*” (Ubi-

cazione) sono di tipo `varchar`; abbiamo scelto una lunghezza massima di 100 caratteri, ma anche 50 magari poteva andare bene. “*MDes*” è indicizzato, perché probabilmente vorremo eseguire ricerche ed ordinamenti sulla Descrizione. Per quanto non sia il massimo dell'efficienza avere un indice “`varchar`” di 100 caratteri, in questo caso è indispensabile. I campi “*ArgId*” e “*SuppId*” sono le **chiavi esterne** per le Tabelle *TbArgomenti* e *TbSupporti*, cioè i riferimenti in *TbMedia* ai valori di descrizione presenti “fuori” di *TbMedia*. Nella tabella principale sono associati ad indice multiplo, nelle tabelle esterne rappresentano la chiave primaria. Notate la scelta dello stesso nome di campo: in database complessi, anche il solo nome di campo può indicare che esiste una connessione logica tra due tabelle, quindi è buona norma usare questa abitudine.

TbArgomenti e **TbSupporti** sono le tabelle che contengono le descrizioni degli argomenti e dei supporti del nostro Archivio; entrambe hanno una chiave primaria di tipo “intero ad incremento automatico” ed abbiamo scelto campi “`varchar`” non troppo lunghi. Questi campi (“*ArgDes*” e “*SuppDes*”) sono indicizzati, perché nell'immissione dei valori nell'archivio useremo delle caselle a discesa ordinate appunto per descrizione (questo sarà più chiaro più avanti).

TbUtenti è la tabella dei nostri utenti dei prestiti. Anche qui un campo “intero ad incremento automatico” per chiave primaria (“*UtId*”), ed un campo “Denominazione” di tipo “`varchar`” indicizzato. Ho di proposito usato pochi campi, ma questa tabella è ampliabile a piacere con tutte le informazioni che vi servono (numero di telefono, posta elettronica etc.).

Infine **TbPrestiti**, l'elenco dei prestiti effettuati, con un campo “*DataPr*” che indica la data del prestito, ed uno “*PreDataR*” che indica la data di restituzione. Anche qui abbiamo degli indici sulle chiavi esterne, cioè “*MId*” per la tabella dei media, e “*UtId*” per la tabella degli Utenti.

Notate infine che in tutte le tabelle è presente un campo di tipo “**timestamp**”, che viene aggiornato automaticamente dal motore di Db.

Questa struttura di archivio è volutamente semplificata, e non pretende affatto essere un modello per una reale gestione, ad esempio, di una Biblioteca. Piuttosto i principi generali sono applicabili nella pratica a TUTTE le procedure di gestione archivi, e quindi su queste basi ognuno potrà ampliare la struttura delle tabelle secondo le proprie esigenze.

Ma è arrivato il momento di passare (finalmente...) ad OpenOffice.org.

5 Finalmente OOo....

5.1 Come OpenOffice.org si collega ai Database

OOo permette la consultazione, la modifica e l'integrazione nei documenti dei Db attraverso apposite interfacce che il programma chiama “**sorgenti dati**”. In pratica, quando vogliamo usare i dati di un Db, dobbiamo prima creare una apposita “sorgente dati”, che indica ad OpenOffice.org come “dialogare” col Db stesso. Una volta stabilita questa connessione, il programma integra alcuni interessanti strumenti che permettono di ottenere risultati notevoli anche senza conoscere (quasi) nulla del motore di Db utilizzato.

Si possono creare vari tipi di “sorgenti dati”, ma in questa sede ci occuperemo in dettaglio di quelle che usano **ODBC** per l'accesso ai Db. Infatti, come abbiamo già detto, questa modalità di interfaccia è in definitiva quella che ci permette di usare MySQL, ma anche qualunque Db abbia un driver ODBC. Dunque è in generale la più flessibile ed usata.

Abbiamo già visto come installare in Windows e Linux il driver ODBC per MySQL, ma altri drivers possono già essere installati e funzionanti nel sistema. Prima però di creare una sorgente dati, è necessario un ulteriore passaggio. Lo standard ODBC prevede, per l'accesso ad uno specifico Database, la creazione di un “**DSN**”, o “**Data Source Name**”. In pratica si tratta di una interfaccia di configurazione che contiene informazioni sull'Archivio a cui bisogna collegarsi. Infatti i server di Db, come MySQL, possono gestire contemporaneamente molti Database diversi, ed è quindi necessario indicare, oltre al tipo di “motore” che vogliamo usare, anche esattamente “a cosa” vogliamo avere accesso. Se volessimo schematizzare questa situazione, potremmo ad esempio usare il diagramma in figura:

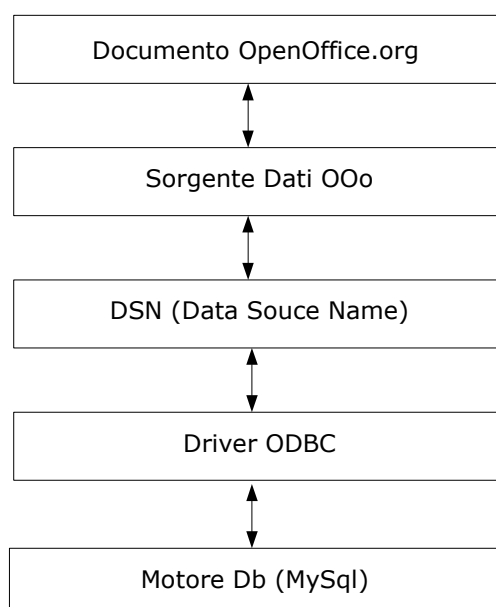


Illustrazione 15 Struttura di accesso al database di OpenOffice.org

5.2 Generazione del DSN in Windows

In **Windows**, i DSN vengono gestiti da un apposito programma presente nel Pannello di Controllo, nella voce “*Strumenti di Amministrazione*”: “**Origine Dati ODBC**”. Alla partenza, il modulo mostra una finestra come quella in figura 16:

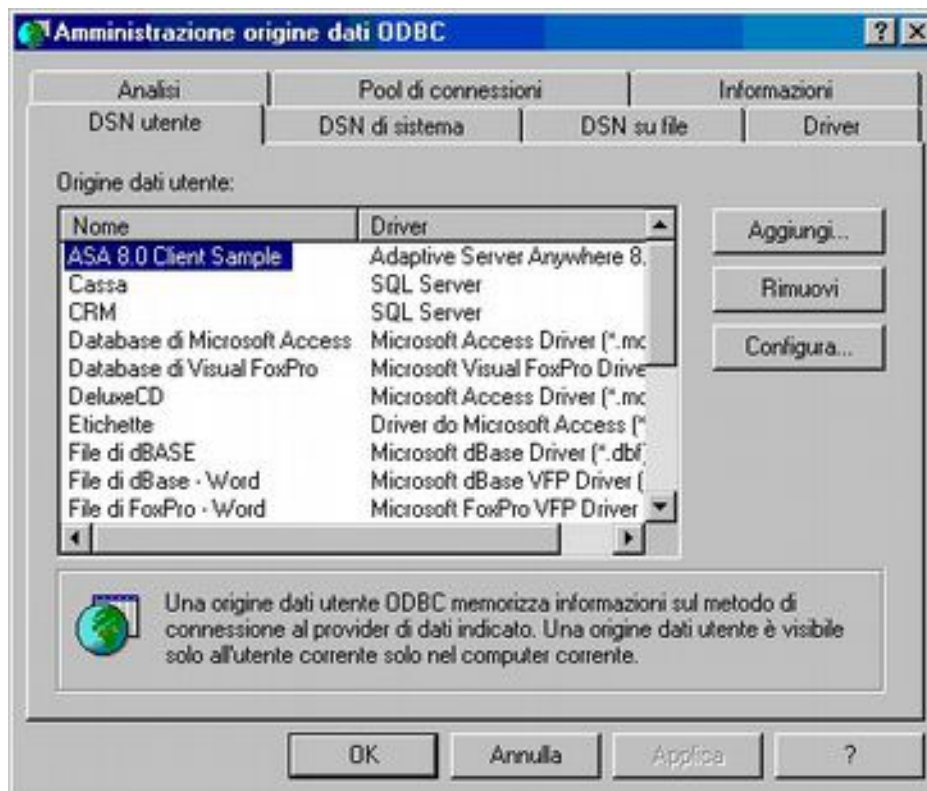


Illustrazione 16 Gestione dei DSN in ambiente Windows

Come si vede, esistono tre tipi di DSN: “**utente**”, “**di sistema**” e “**su file**”. Tralasciando l'ultimo (poco importante ai nostri fini), è importante notare la differenza tra i primi due. Il “*DSN Utente*” ha valore solo per l'utente che lo crea, mentre quello “*di sistema*” vale per tutti gli utenti del Computer. Per il resto la procedura da seguire è assolutamente identica. Assicuriamoci che il Server MySQL sia attivo e premiamo il pulsante “Aggiungi...”. Ci verrà chiesto di selezionare un Driver tra quelli disponibili sul sistema; scegliamo “*MySQL ODBC 3.51 Driver*” e premiamo il pulsante “Fine”. Otterremo una finestra come quella in figura 17.

Assegniamo il nome “**Mediateca**” al nostro DSN, ed una descrizione a piacere al campo “Description”. Nei parametri di connessione dobbiamo specificare il **nome o l'indirizzo IP del Server** (“localhost” se gira sulla nostra stessa macchina), il **nome del Database** (che deve essere lo stesso usato nella creazione eseguita con MySQLCC), l'**utente** (nel nostro caso root) e l'eventuale **password**, la **porta** (3306 è quella di default e di solito non va modificata).

Il pulsante “options” permette di specificare molti ulteriori parametri per la connessione, ma di solito il DSN funziona egregiamente con i valori di default. Per ultimo possiamo testare la nostra connessione con l'apposito pulsante: un messaggio di conferma ci informa che va tutto

per il meglio. Confermiamo, ed ora il nostro DSN dovrebbe comparire, col nome di Mediateca, nell'elenco dei DSN utente.

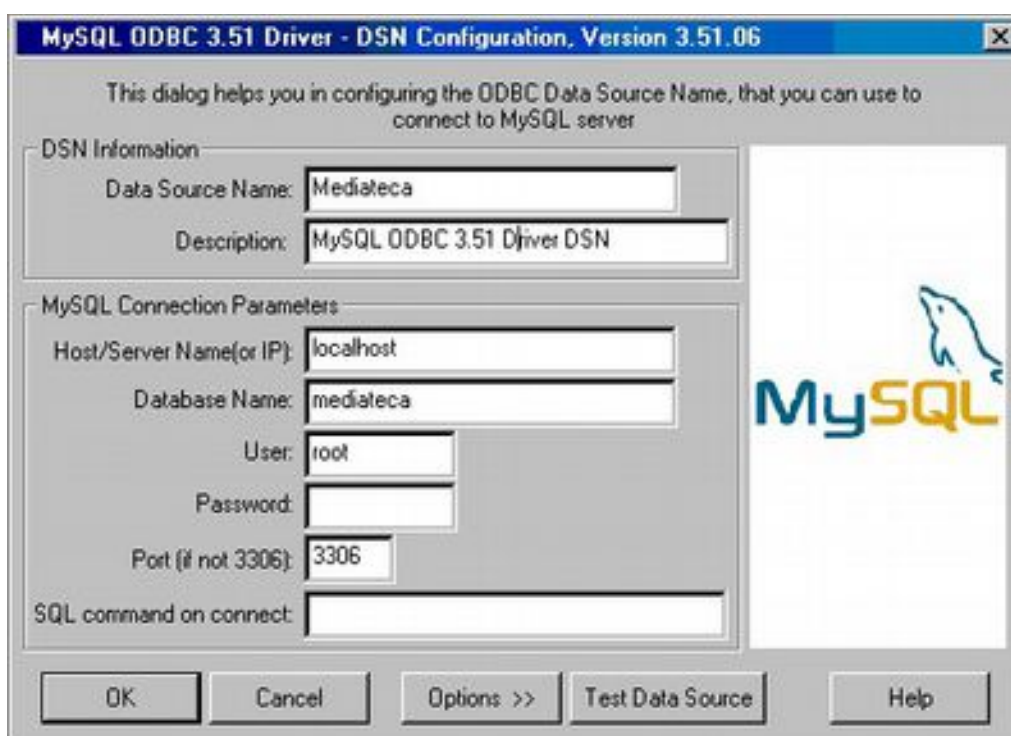


Illustrazione 17 Configurazione del DSN del database su MySQL

5.3 Generazione del DSN in Linux

In **Linux** è disponibile un programma abbastanza simile per la gestione dei DSN. Posto che anche il pinguino fa la stessa differenza tra DSN “utente” e “di sistema”, in SUSE 9.0 il programma può essere richiamato con “Sistema->Configuration->unixODBC Configuration”. Dal Tab “User DSN” si sceglie il pulsante “Add...” e si aggiungono i parametri come in figura 18.

Nella casella “server” va immesso l'indirizzo



Illustrazione 18 Definizione del DSN sotto Linux

IP (oppure il nome) del server MySQL, e nella casella successiva, “Database”, il nome del Db creato sul server.

A questo punto il DSN è disponibile per tutte le applicazioni che dialogano con i Db tramite ODBC.

Tips



Per quanto sia possibile creare sorgenti dati MySQL di sistema, non mi è riuscito, per qualche misterioso motivo, di collegare poi questo tipo di DSN ad Openoffice. Consiglio quindi di utilizzare solo DSN di tipo “utente”.

5.4 La sorgente dati di OOo

Dopo tanta fatica, è arrivato il momento di “vedere” il frutto del nostro lavoro “dentro” OpenOffice.org. Abbiamo già detto che per la connessione ad un Database è necessario creare in OOo una “sorgente dati”. Apriamo ad esempio un documento di testo vuoto, la voce di menù che ci interessa è presente nel menù “**Strumenti**” e si chiama appunto “**Sorgente Dati..**”; una volta selezionata appare una finestra come questa (figura 19):

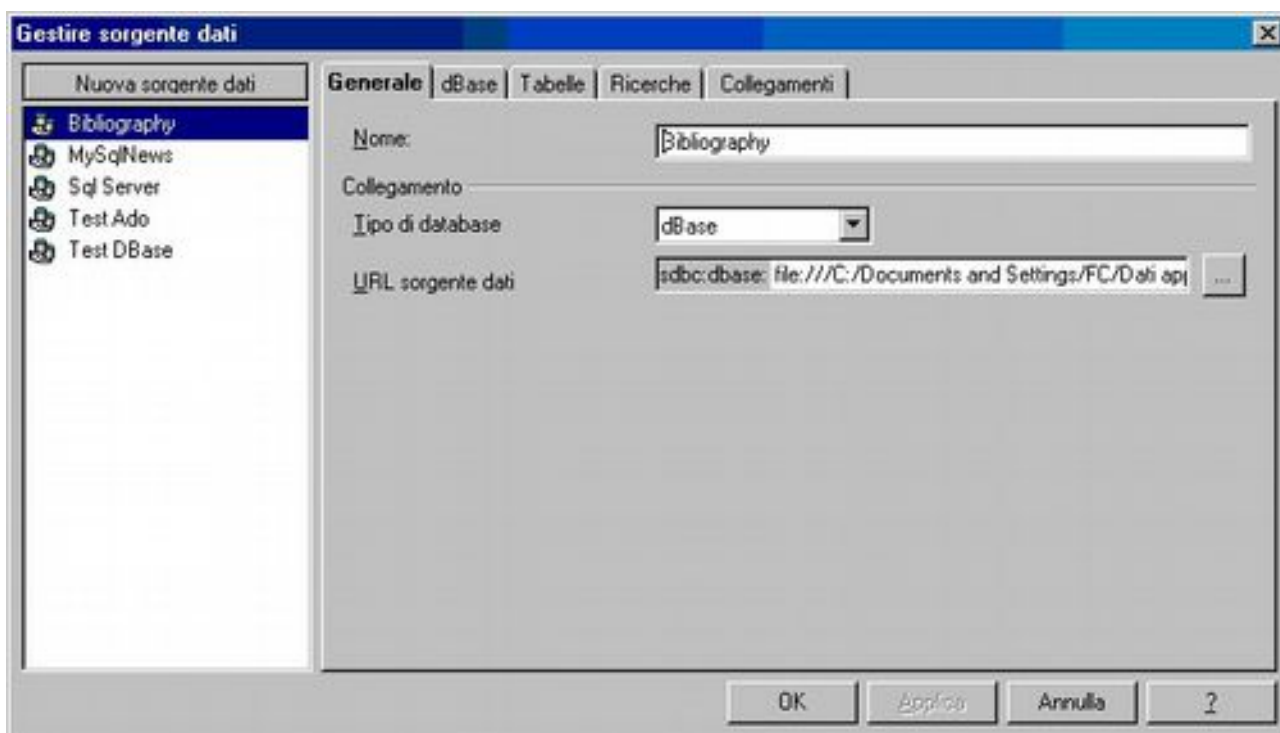


Illustrazione 19 Creazione di una sorgente dati

Sulla sinistra sono elencate le sorgenti dati già create, e sulla destra sono evidenziati i parametri di connessione per ogni sorgente. OOo può gestire molti tipi diversi di Sorgenti Dati, ed ogni tipo ha parametri di connessione differenti. Premiamo il pulsante “**nuova sorgente dati**”, assegnamo il nome “**Mediateca**” e scegliamo dalla casella a discesa il tipo “**MySQL**”. Un click

su “**applica**” salverà il nome di connessione. Passando quindi alla **Tab “MySql”** avremo una situazione di questo tipo:



Illustrazione 20 URL della sorgente dati (database) MySQL

Dobbiamo ora specificare qual'è il nostro DSN, e quindi premiamo il pulsante per la selezione dell'*URL sorgente dati* :

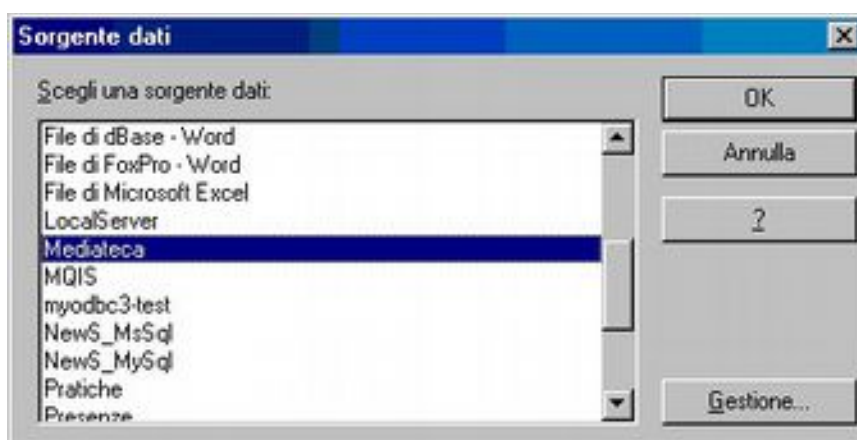


Illustrazione 21 Uso del browser dei DSN

Scegliamo “**Mediateca**”, e non dimentichiamo di scrivere “**root**” nella casella del nome utente. Se ora passiamo alla tab “**Tabelle**”; dovremmo vedere elencate tutte le nostre tabelle. Confermiamo con “**OK**”, ed ora la nostra sorgente dati è pronta per essere utilizzata con tutti i documenti di OOo.

Tecnica

Nella Tab “Tabelle” delle proprietà della sorgente dati è possibile selezionare anche **solo alcune** delle Tabelle che compongono il Database. Questo può essere particolarmente utile nel caso ci collegassimo a Db molto ampi (pensate ad una gestione aziendale) e ci servissero



poche tabelle (magari solo quelle relative ai clienti). In generale è buona norma specificare nella sorgente dati SOLO le tabelle effettivamente utilizzate.

Tips



La casella a discesa “**Tipo di Database**” elenca tutte le tipologie di DB che possono essere incluse in una Sorgente Dati di OOo. La più generica, come abbiamo visto, è ODBC, ed infatti avremmo potuto collegare il nostro Db MySQL anche selezionando questa opzione. Stranamente, però, i Db di MySQL collegati con l'opzione “ODBC” della casella di controllo risultano di sola lettura. Come per MySQL, esistono anche tipologie specifiche per alcuni motori Db molto diffusi. Ad esempio per i Db *Access* si consiglia di usare il tipo *ADO* anche se è poi necessario impostare manualmente la stringa di connessione. Esiste anche una voce per *Adabas*, il motore di Db fornito assieme alla versione “commerciale” di OOo, *StarOffice* di *Sun Microsystems*. Ed infine, se questi motori di Db ci sembrano troppo complessi, potremmo sempre usare il buon vecchio Dbase...

5.5 Il buon vecchio DBase...

Perdonatemi una breve (ma forse utile) digressione, amici miei. E' infatti con gli occhi lucidi ed un groppo alla gola che mi accingo a parlarvi di un vecchio amico, che molti di voi neppure conosceranno, il DBase. **DBase II** è stato il primo SW di gestione DB che ha avuto successo nell'arcano mondo di Ms-Dos, ai tempi remoti che Berta filava... Stiamo parlando della metà degli anni ottanta, quando ogni PC che voleva definirsi tale doveva avere a bordo un SW di elaborazione testi (WordStar), un foglio elettronico (Lotus 123) ed appunto DBase II. La leggenda dice che DBase I non sia mai esistito e che gli sviluppatori partirono dal 2 per dare un'aria di “vissuto” al programma. Fatto sta che lo standard “DBase” ha conosciuto da allora un enorme successo, e ce lo ritroviamo pari pari nel nostro aggiornatissimo OOo.

DBase è un semplice formato di archiviazione che prevede un file dati con estensione **.dbf** per ogni tabella, associato ad uno o più file con estensione **.ndx** per gli indici. Non si tratta quindi di un server Db, quanto piuttosto di una specifica di formato che ognuno può utilizzare come vuole. Nonostante l'età, le prestazioni sono più che dignitose, ed ancora può essere usato per scopi non troppo complessi. Per farla breve, non vi parlerò delle evoluzioni che DBase ha avuto nel tempo (DBase III, poi IV, poi il Clipper, uno dei compilatori più diffusi in passato), ma solo di come usarlo con OOo, perché può tornare utile in più di una occasione.

Bisogna, innanzi tutto, notare che la sorgente dati di esempio di OOo (Bibliography) è appunto di tipo DBase, e questo significa che questa tipologia di archivi può essere usata in OpenOffice.org senza l'ausilio di alcuno strumento esterno. Quello che ci serve, infatti, è solo una directory vuota del nostro Hard Disk. La creazione di una nuova Sorgente Dati di tipo DBase in OOo infatti richiede solo di specificare nella casella “*URL Sorgente Dati*” un percorso valido. Fatto questo, tutti gli strumenti per la creazione e modifica di tabelle ed indici sono disponibili in OOo. Vi chiederete allora perché non abbiamo usato DBase, e ci siamo rivolti invece ad uno strumento sicuramente più complesso come MySQL. La risposta è semplice: DBase non ha alcune delle caratteristiche che ci torneranno utili nel seguito, in particolare le chiavi primarie e gli interi ad incremento automatico. Comunque per compiti non troppo complessi è sicuramente utilizzabile, e nel seguito torneremo sull'argomento.

5.6 Gestione delle Sorgenti Dati

Se tutto è andato bene, da un qualsiasi documento OOo, alla pressione del tasto F4 (che attiva il “panel” delle sorgenti dati, dovremmo avere circa la situazione in figura 22:

The screenshot shows the OpenOffice.org database interface. On the left, a tree view displays the database structure under 'Mediateca'. The 'Tabelle' (Tables) folder is expanded, showing several tables: 'medioteca.tbargomenti', 'medioteca.tbmedia', 'medioteca.tbprestiti', 'medioteca.tbsupporti' (which is selected and highlighted in blue), and 'medioteca.tbautenti'. Below this, 'MySQLNews' and 'Sol Server' are also listed. On the right, a data table is displayed with three columns: 'Suppld', 'SuppDes', and 'SuppTs'. The table contains seven rows of data, with the last row being a header row for a new record. The status bar at the bottom indicates 'Record di dati 1 da 7'.

Suppld	SuppDes	SuppTs
1	DVD Video	01/04/04 18.38
2	DVD Audio	01/04/04 18.38
3	CD Audio	01/04/04 18.38
4	CD Software	01/04/04 18.39
5	Libro	01/04/04 18.39
6	Articolo	01/04/04 18.39
7	Rivista	01/04/04 18.39
* <Campo au		

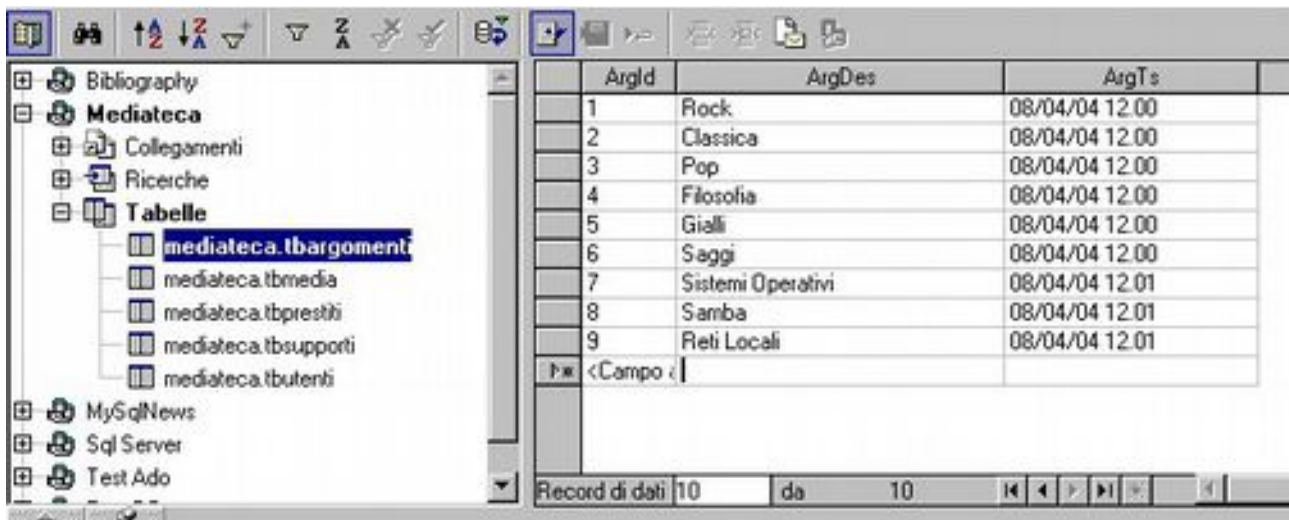
Illustrazione 22 Accesso ad una tabella attraverso la sorgente dati

Sulla sinistra abbiamo, in una struttura ad albero, l'elenco delle nostre sorgenti dati. Aprendo, con le caselle col segno “+”, il dettaglio di ogni struttura, si possono esplorare le componenti della sorgente dati selezionata. In generale, col tasto destro del mouse è anche possibile accedere ad un menù contestuale che permette ulteriori elaborazioni sull'elemento selezionato. Notate che, una volta selezionata una tabella, sulla destra ci appare il contenuto della Tabella stessa. Questo contenuto è dinamico, nel senso che si può intervenire direttamente sui dati, con modifiche, aggiunte e cancellazioni. Infatti vediamo in figura i campi definiti per la tabella **tbsupporti**, che, ricordiamo, descrive le tipologie di supporti presenti. Queste modifiche però avvengono senza alcun controllo e quindi in generale è più comodo, per la gestione della tabella crearsi un modulo, come vedremo tra poco.

Tips

La tabella **tbsupporti** prevede un solo campo modificabile dall'utente, cioè “**SuppDes**”, in quanto gli altri sono gestiti direttamente dal motore di Db. Si tratta, infatti, di “**Suppld**”, cioè un intero ad incremento automatico che abbiamo usato come chiave primaria, e di “**SuppTs**”, un campo di tipo Timestamp aggiornato dal sistema, che serve a rendere compatibile il nostro archivio con altri front end di Db. Nessuno di questi campi dovrebbe essere modificato dall'utente, ma nella tabella mostrata da OOo tutti e due sono variabili a piacere. La cosa può portare a risultati imprevedibili, ed è questo uno dei motivi per cui non è saggio fare modifiche direttamente nel pannello destro di una sorgente dati.

Per i nostri scopi, però, dobbiamo “popolare” con qualche dato le nostre tabelle, e quindi ho immesso delle informazioni di prova che ci aiuteranno nel seguito. In particolare, per la tabella **tbargomenti**:



ArgId	ArgDes	ArgTs
1	Rock	08/04/04 12.00
2	Classica	08/04/04 12.00
3	Pop	08/04/04 12.00
4	Filosofia	08/04/04 12.00
5	Gialli	08/04/04 12.00
6	Saggi	08/04/04 12.00
7	Sistemi Operativi	08/04/04 12.01
8	Samba	08/04/04 12.01
9	Reti Locali	08/04/04 12.01

Illustrazione 23 Inserimento di un record in tabella TbArgomenti

Notate come, man mano che si aggiungono delle righe, il sistema assegni il numero progressivo di “**ArgId**”, ed aggiorni con la data e l'ora corrente il timestamp “**ArgTs**”. Questi campi non devono essere immessi manualmente, mi raccomando.

Abbiamo già detto che, una volta creata la sorgente dati, è possibile modificare la struttura dei nostri archivi direttamente da OOo, e quindi senza ricorrere ad esempio a MySQLCC. Se infatti apro il menù contestuale (tasto destro del mouse) ad esempio su “mediateca.tbargomenti” e scelgo la voce “modifica tabella”, avrò a video questa finestra (figura 24):



Dovendo prevedere la gestione di molti Database diversi, OOo cerca di adattarsi, ed il tipo di dati disponibile da assegnare ad un campo varia molto in funzione della sorgente dati. Con MySql, inoltre non c'è una corrispondenza esatta, e molti tipi di dati potrebbero portare a pericolosi equivoci. Notate anche che la “casella di formato” in MySql non è utilizzabile.

Per sorgenti dati DBase, ad esempio, il tipo di dati selezionabile per un campo è assai ridotto, come ridotte sono le opzioni disponibili. Il tipo numerico è uno solo (Decimal) e non c'è traccia di assegnazione automatica del valore (incremento automatico).

Una pressione sull'apposito pulsante della barra sotto il Menu apre la finestra di gestione degli indici (figura 25):

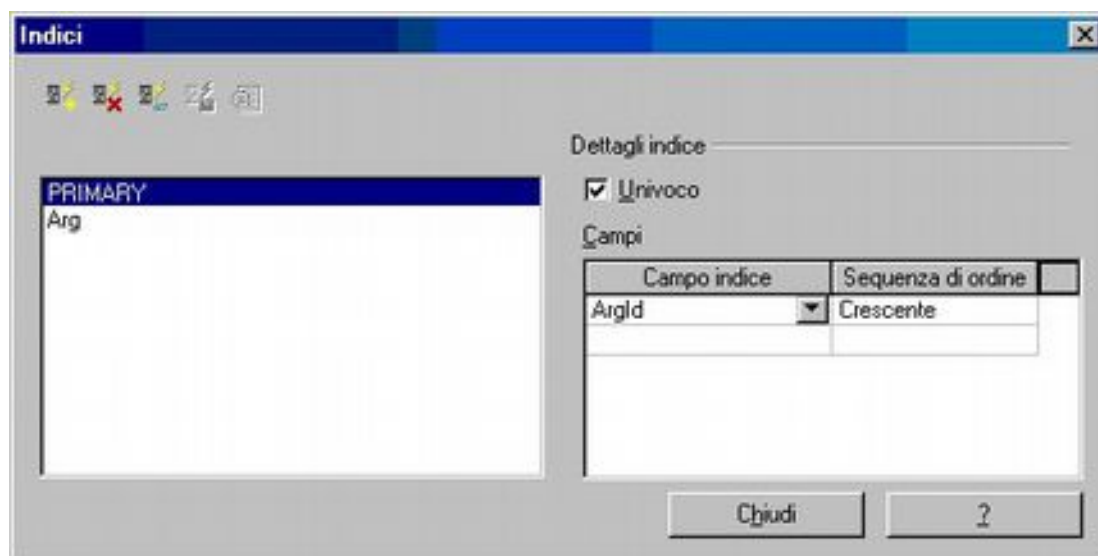


Illustrazione 25 Accesso agli indici

Il funzionamento è piuttosto intuitivo, quindi non ci dilungheremo sull'argomento. Tenete però presente che da questa finestra non è possibile assegnare una chiave primaria. Anzi esiste un piccolo bug nella creazione delle tabelle di MySQL che di fatto impedisce di assegnare “da soli” una chiave primaria del tipo integer auto increment, ed obbliga ad una procedura contorta, che non descriverò.

Infatti, SALVO EMERGENZE, è assolutamente sconsigliato creare o modificare tabelle MySQL direttamente da OOo: la strada migliore è senza dubbio MySQLCC o la riga di comando. Diverso è il caso di sorgenti dati DBase, in quanto l'unico strumento disponibile è appunto quello di OpenOffice.org (o avete da qualche parte un vecchio DBase Dos ?).

Tecnica

Uno dei problemi principali nella creazione di tabelle MySQL da OOo è l'impossibilità di specificare il tipo di tabella da generare. Come vedremo in seguito, oltre al tipo di tabella “**MyIsam**” (che è quello predefinito) in MySQL si possono usare altri “formati” di archiviazione. Quello più interessante si appoggia ad un prodotto Open Source di terze parti, di nome “**InnoDB**”. Questo formato supera alcune delle limitazioni intrinseche delle tabelle MyIsam. Notate che nello stesso Db possono coesistere tabelle in formato diverso, e che si può sempre trasformare una tabella MyIsam in InnoDB e viceversa.



5.8 Il problema dei valori predefiniti

Vorrei ora descrivere un problema che OOo ha con la gestione delle tabelle MySQL, e che, se trascurato, può essere molto fastidioso. Nella tabella **tbmedia** abbiamo definito alcuni campi numerici che servono sia a collegarsi con altre tabelle (SuppId, ArgId), sia a specificare dei valori numerici propri (MPrezzo). Abbiamo già detto che di solito non è buona norma permettere l'immissione di un valore *null* nei campi numerici, e quindi abbiamo, a livello di definizione di tabella, assegnato a questi campi il valore predefinito di zero. Questo significa che all'aggiunta di un nuovo record, il sistema stesso si occupa di fare in modo che il valore di questi campi, se non specificato, sia zero.

Bene, come vedremo, nei “**formulari**” di OOo basati sulle tabelle (cioè sui documenti collegati in cui è possibile impostare maschere e tabelle per la gestione del Db) questo non accade ed il programma riporta un messaggio di errore. In pratica OOo non si accorge che per il campo è stato definito un valore “standard”, e richiede l'immissione manuale di un dato, in quanto non è permesso lasciare il campo a null. Stranamente, invece, va tutto bene se il record si aggiunge direttamente dal pannello di gestione delle sorgenti dati.

Una prima soluzione può essere richiamare la modifica della struttura della tabella in OOo, e reimpostare il valore predefinito per il campo, come in figura 26:

Infatti OOo non lo riporta in automatico. Incidentalmente questo significa anche, come vedremo, che OpenOffice.org salva numerose informazioni aggiuntive sulla struttura della sorgente dati in qualche suo file di configurazione, e bisogna tenerne conto.

Se però così si bypassa il problema con i campi numerici, lo stesso persiste con i campi “**timestamp**”. Abbiamo già definito questa tipologia di campo come *un valore di data/ora che il motore di Db assegna in automatico*

per permettere una migliore gestione degli accessi concorrenti. Pur non avendo una utilità immediata con OOo, il timestamp è indispensabile se desideriamo che il nostro Db sia poi usabile anche con altri front-end, come Ms Access. Ora, se per un campo numerico è possibile assegnare un valore predefinito all'interno di OOo, lo stesso non si può fare con un campo timestamp. Anzi, anche se nella casella a discesa della definizione del campo di OOo il tipo “Data/Ora (timestamp)” esiste, in pratica il programma rifiuta di assegnarlo ed, al salvataggio della tabella, si ritorna al classico “Data/Ora (datetime)”.

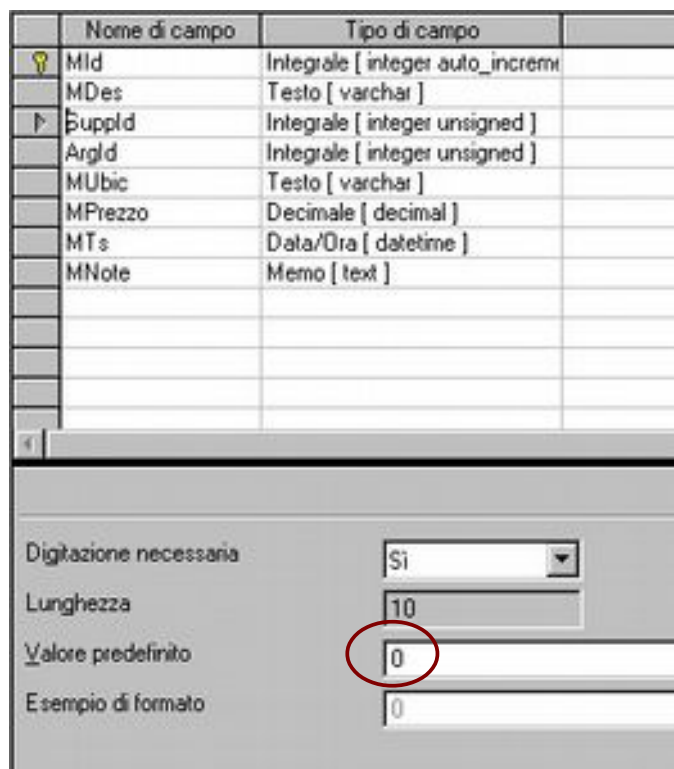


Illustrazione 26 Valore predefinito di un campo

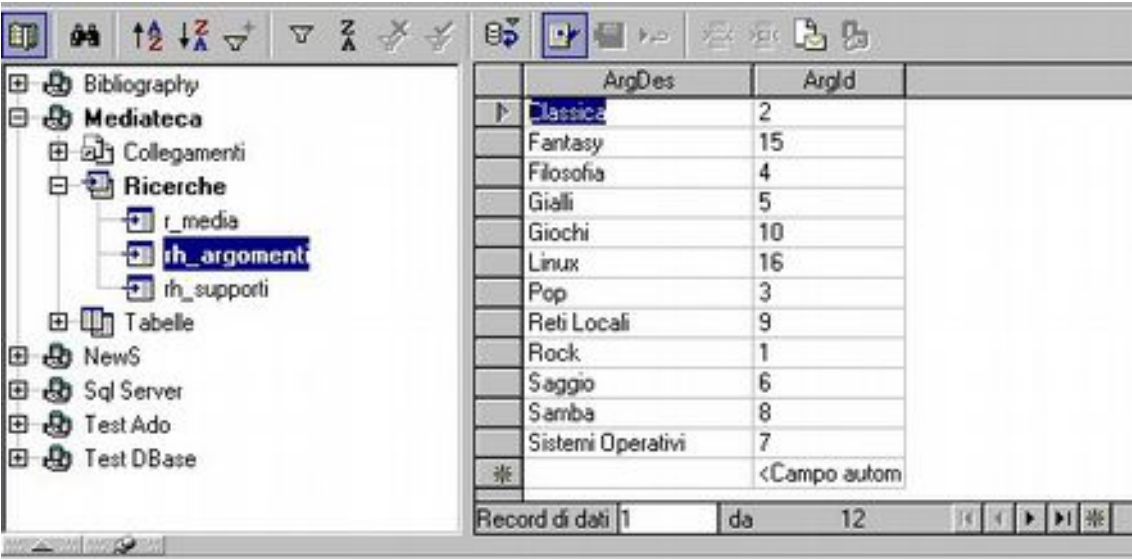
Siccome il “formulario” è la modalità principale di interazione con i Db di OpenOffice.org, siamo in un vicolo cieco; dobbiamo evitare di usare i campi “timestamp” ??

Fortunatamente abbiamo una soluzione di riserva che ha anche, come vedremo, dei vantaggi tecnici, ed è, diciamo, piuttosto *elegante*: l'uso delle “**ricerche**” come base per i formulari. Infatti, in questo caso, il problema miracolosamente scompare...

Ma che cos'è una “ricerca” ?

5.9 Le Ricerche

Il termine inglese sarebbe “**query**”, ed infatti in Ms Access sono chiamate così. Però “query” è un po' generico, in quanto viene usato in molti contesti diversi, con significati non proprio identici. Quindi “ricerca” ci va bene.



The screenshot shows the OpenOffice.org database interface. On the left, a tree view displays the database structure, including a folder named 'Ricerche' which contains the query 'rh_argomenti'. The main window displays the results of this query as a table with two columns: 'ArgDes' (Argument Description) and 'ArgId' (Argument ID). The data is sorted alphabetically by 'ArgDes'.

ArgDes	ArgId
Classica	2
Fantasy	15
Filosofia	4
Gialli	5
Giochi	10
Linux	16
Pop	3
Reti Locali	9
Rock	1
Saggio	6
Samba	8
Sistemi Operativi	7
<Campo autom	

At the bottom of the window, a status bar indicates 'Record di dati 1 da 12'.

Illustrazione 27 Query o ricerche

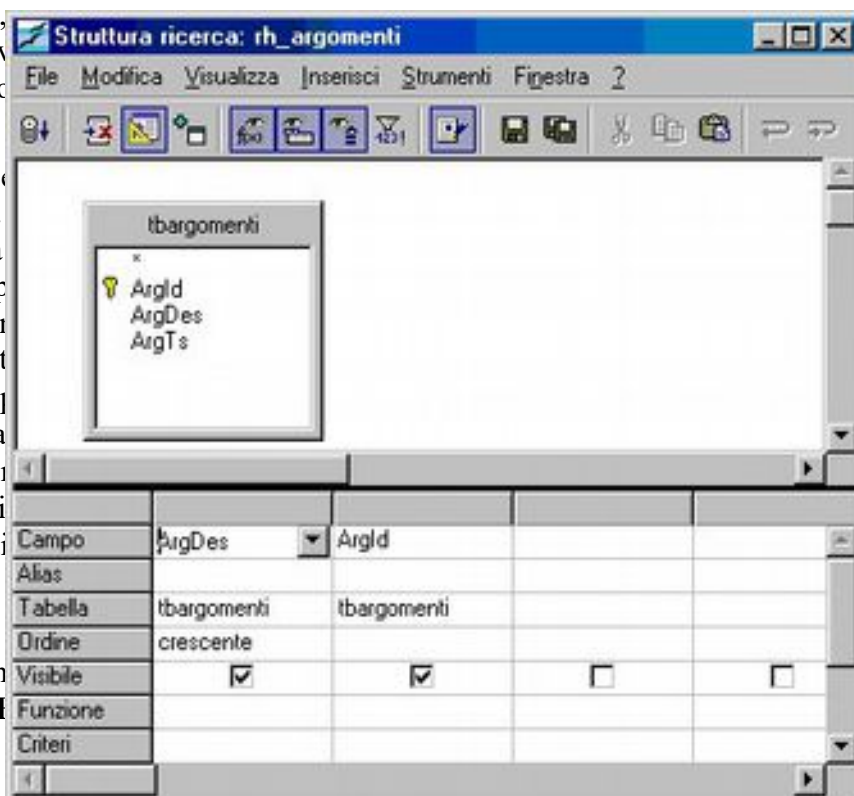
Una “ricerca” è una particolare “vista” dei dati presenti in una o più tabelle del nostro Db. Si presenta come una nuova tabella, ma non deve necessariamente rispettare una struttura “fisica” esistente. Infatti alcune colonne possono essere frutto di calcoli, oppure l'ordine non è quello naturale della tabella di origine. Per fare un esempio, per motivi di comodità potremmo volere un elenco degli elementi compresi nella tabella **tbargomenti**, in ordine alfabetico. Uno dei modi è creare una “ricerca”. Questa struttura è anche utile per scartare dalle elaborazioni quei campi che non hanno una utilità immediata, come i timestamp. Alle ricerche è dedicato uno dei rami dell’“albero” a sinistra nel pannello delle sorgenti di dati. Per ogni sorgente possiamo creare una o più ricerche, a seconda delle nostre esigenze. Nella figura 27 possiamo vedere una ricerca, chiamata “rh_argomenti”, che ci mostra appunto il contenuto di **tbargomenti** ordinato per descrizione.

Per creare una nuova ricerca, si seleziona “nuova ricerca” (vedi la tabella (o le tabelle) che deve essere quella mostrata in figura 28.

La parte superiore contiene la visualizzazione delle relazioni tra le tabelle, che dovranno “comporre” la query. Un doppio click sul nome, nel pannello a griglia della parte bassa. Infine, i campi possono essere disposti e visualizzati. Per ogni campo è possibile definire: un **alias**, cioè una intestazione; uno o più criteri di ordinamento; se deve essere visibile; uno o più campi (su questo torneremo in seguito).

Tips

Siccome nel seguito useremo la funzione dell'argomento, è O.K. (vedi figura 28).



Dei pulsanti sulla barra sotto *Illustrazione 28 Definizione di una ricerca*

prima del risultato delle nostre impostazioni ed è utile per verificare al volo la correttezza delle nostre scelte. Il terzo da sinistra, “**attiva/disattiva vista disegno**” è invece particolarmente importante, perché permette di visualizzare la ricerca con la sintassi del linguaggio SQL. In questo modo è anche possibile “spedire” ordini direttamente in SQL al motore di Database, per chi di voi è in grado di farlo. Nel nostro caso, sarebbe:

```
SELECT `ArgId` AS `Id`, `ArgDes` AS `Descrizione`
FROM `mediateca`.`tbargomenti` `tbargomenti`
ORDER BY `Descrizione` ASC
```

cioè in pratica:

```
SELEZIONA (elenco dei campi) DA (elenco delle tabelle) ORDINATO PER (nome di campo).
```

Facile, no ?

Tecnica

SQL sta per Structured Query Language, ed è una sintassi definita anche da uno standard ANSI per la creazione, la modifica e l'interrogazione di basi di dati. Si basa su relativamente poche parole chiave o “istruzioni” (come SELECT) con una sintassi di solito semplice e



comprensibile. Gli standard ANSI per SQL sono ANSI 92 e ANSI 99. Ovviamente ANSI 99 è molto più ampio del predecessore. MySql è sostanzialmente conforme ad ANSI 92, ma non implementa completamente ANSI 99. Inoltre esistono “estensioni” a SQL caratteristiche di MySql, che potreste non ritrovare in altri prodotti di Database. Questo degli standard è un problema annoso, ancora in parte non risolto, per cui esistono molti “dialetti” SQL, anzi tutti i produttori di Server di Db hanno una propria “versione” del linguaggio, ovviamente parzialmente incompatibile con le altre. Per quanto riguarda OOo, il programma “traduce”, come abbiamo visto, le ricerche create col tool visuale in istruzioni SQL. Purtroppo non sempre queste istruzioni sono corrette per il Db a cui sono destinate, e quindi alcune ricerche potrebbero non funzionare come ci si aspetta. Fortunatamente è anche possibile “spedire” istruzioni SQL direttamente al motore Db, senza “traduzione” di OpenOffice.org, risolvendo così in parte i problemi.

6 La Mediateca

6.1 Filosofie a confronto

Quando si parla di interfacce utente ognuno ha una opinione diversa dagli altri, e le discussioni si sprecano. C'è chi ad esempio ritiene che, per un programma gestionale, il mouse sia solo di impaccio e la tastiera rende il lavoro migliore; c'è chi preferisce avere il desktop ingombro di icone e programmi, chi sceglie la pulizia assoluta e quindi focalizza un compito alla volta. Siccome il nostro argomento sono i Database, fortunatamente c'è poco da scegliere: *tabella o scheda* ?

Se scegliamo un archivio qualsiasi, la vista a **tabella** (tabellare, nella terminologia di OOo) è simile a quella di un foglio elettronico, con a video una lista di records (cioè righe) e la possibilità di scorrere l'archivio di molti records alla volta. La vista a **scheda** (a righe o a colonne), più tradizionalmente, mostra appunto una scheda alla volta ed i campi sono sistemati sul video in modo da facilitare l'immissione dei dati. Scegliere una o l'altra modalità di interazione con l'utente dipende, appunto, dai gusti personali, ed anche dalla quantità e qualità dei dati da gestire. La tabella consente di avere a schermo un numero elevato di records, e facilita gli spostamenti nell'Archivio e l'individuazione di specifici records, se ad esempio l'archivio è ordinato. La scheda rende agevole l'immissione e la variazione dei dati di un *singolo* record.

Io uso solo la vista tabellare se i record da gestire sono pochi; invece, come vedremo, una combinazione tra le due per archivi complessi. Ma andiamo con ordine.

6.2 Il Formulario

Cominceremo col creare dei piccoli moduli per la gestione delle tabelle “di contorno” del nostro Database, cioè **tbargomenti** e **tbsupporti**. Nonostante si potrebbe facilmente “popolare” di dati queste tabelle, vista la loro semplice struttura, direttamente dal pannello delle sorgenti (che si attiva, lo ricordo, con la pressione del tasto **F4**), costruiremo dei formulari appositi per esse. Dovremo aver già creato delle “ricerche” per le due tabelle, secondo quanto già spiegato nei paragrafi precedenti. In particolare, le due ricerche comprenderanno solo i campi Id e Descrizione, e saranno ordinate appunto per Descrizione.

Il “formulario” è un documento di OOo (testo o foglio elettronico) collegato ad una sorgente dati; attraverso un formulario è possibile visualizzare e modificare le tabelle che fanno parte del nostro DataBase. In OOo esiste una funzione di auto composizione dei formulari, ma qui preferiremo la procedura manuale, che è anche più didattica.

Cominciamo con un documento di testo vuoto, e visualizziamo con **F4** le sorgenti dati. Per prima cosa visualizziamo la *barra mobile per le funzioni del formulario* (figura 29), richiamabile con il pulsante presente nella *barra degli strumenti* sulla sinistra della finestra di OpenOffice.org.



Questa barra mobile comprende tutti gli strumenti dedicati alla gestione dell'interfaccia verso le sorgenti dati, e durante il lavoro è opportuno che rimanga visualizzata.



Illustrazione 29 Funzioni di formulario

Non vogliamo in questa fase aiuti automatici, quindi **disattiviamo il “pilota”** (ultimo pulsante a destra della seconda riga). Siccome intendiamo usare un'interfaccia di tipo tabellare, scegliamo lo strumento “Campo di controllo Tabella” (che è l'ultimo a destra sulla prima riga) e tracciamo un bel rettangolo sul nostro foglio bianco. Otterremo più o meno questo (figura 30):

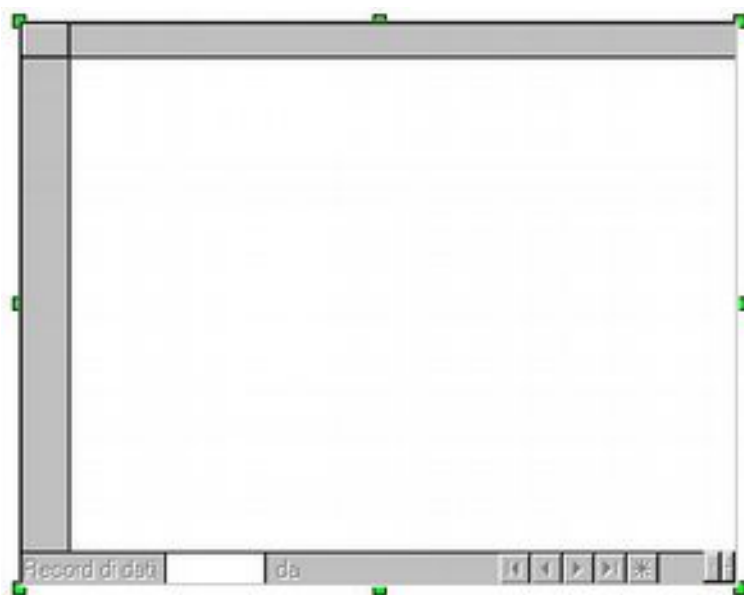


Illustrazione 30 Definizione del campo di controllo tabella

Bene, abbiamo creato la base per la nostra tabella. Siamo in modalità “**bozza**” e ciò è anche indicato dall'attivazione dell'apposito pulsante sulla barra delle funzioni del formulario:



In modalità “bozza” è possibile cambiare l'aspetto e le caratteristiche del nostro documento. Ogni **elemento** del formulario ha delle “**proprietà**” che possono essere modificate secondo le nostre necessità. Ad esempio, selezionando la tabella creata, col tasto destro del mouse possiamo scegliere la voce “**formulario...**” per stabilire per prima cosa quali dati vogliamo gestire. Nella finestra delle proprietà, nella tab “**Generale**” assegnamo il nome del formulario (“argomenti”); nella tab “**dati**” dobbiamo specificare i parametri di collegamento come in figura 31:

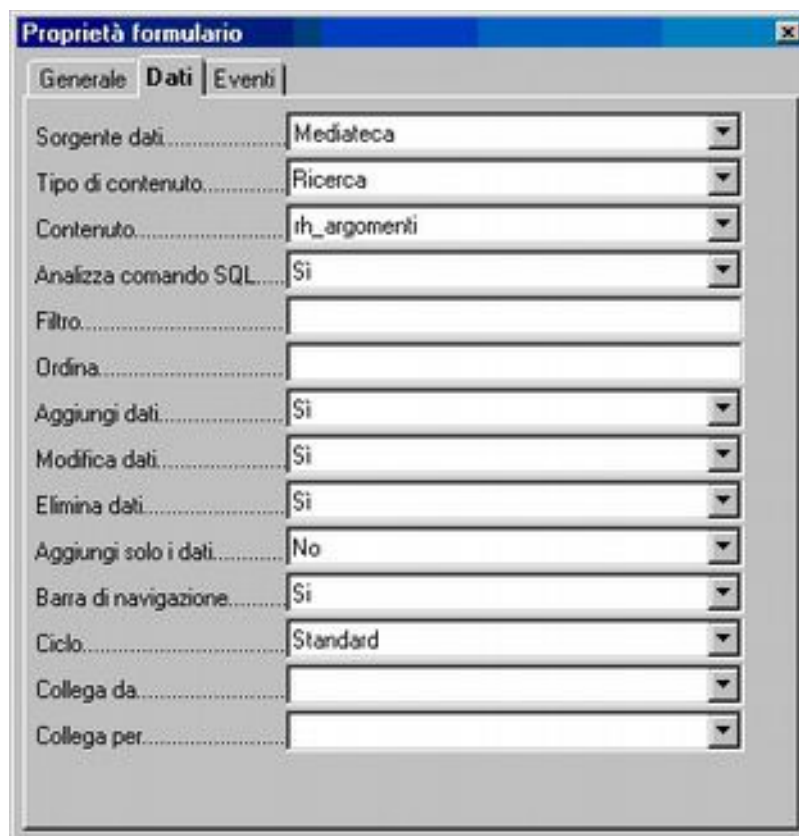


Illustrazione 31 Definizione dei parametri di collegamento

Quindi: *sorgente dati* “**Mediateca**”, *Tipo di contenuto* “**ricerca**”, *contenuto* “**rh_argomenti**” (che è il nome della ricerca). Bene, ora dobbiamo “popolare” la nostra tabella, quindi posizioniamo il cursore sul bordo superiore della tabella stessa e col tasto destro del mouse scegliamo “Inserisci colonna” e quindi “campo di testo”. Siccome ci servono due colonne, ripetiamo l'operazione due volte, ottenendo questo risultato (figura):

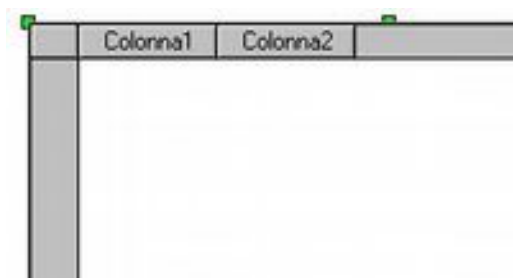


Illustrazione 32 Definizione delle colonne

Dobbiamo ora “personalizzare” le colonne, cioè fare in modo che contengano i dati che ci servono. Selezioniamo “colonna1” ed ancora col tasto destro del mouse scegliamo la voce “**Colonna...**”. In questo modo si apre la finestra delle proprietà della colonna. Il primo campo da visualizzare per la nostra tabella è l'identificatore, valore numerico intero assegnato dal motore di Db, quindi nel Tab “**Generale**” possiamo immettere questi valori (figura 33):

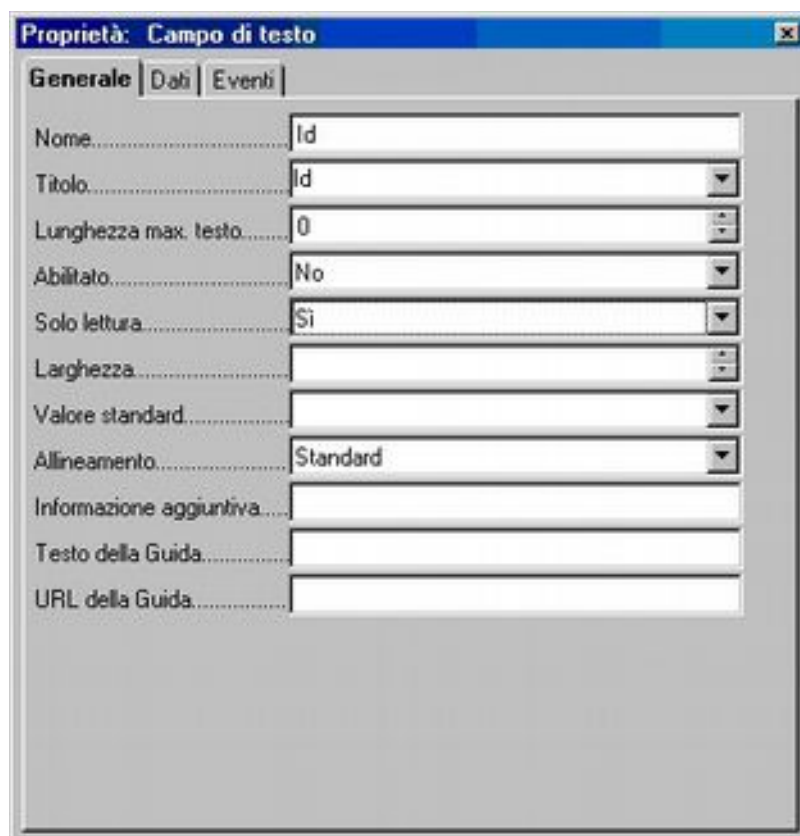


Illustrazione 33 Definizione della proprietà della colonna

Quindi: *Nome della colonna “Id”, Titolo visualizzato “Id”, Abilitato “No”, Sola lettura “Si”*. Questi due ultimi valori fanno in modo che il campo non sia modificabile. Se volete informazioni sulle altre opzioni presenti, la pressione del tasto **F1** vi porta all'ottima guida in linea di OOo. Nella Tab “**Dati**” dobbiamo indicare quale campo desideriamo sia visualizzato sulla colonna, quindi nel *Campo di Dati* selezioniamo **ArgId** dalla casella a discesa (figura 34).



Illustrazione 34 Definizione della visualizzazione dei dati sulla colonna

Passiamo ora alla colonna di descrizione. Quindi Colonna2, tasto destro, “**Colonna...**” ed impostiamo, nella tab “**Generale**”, *nome “Descrizione”, Titolo “Descrizione”, Abilitato “Si”, Sola Lettura “No”,* perché il campo deve essere modificabile. Nella Tab “**Dati**”, Campo di Dati “**ArgDes**”. Siccome la colonna di Descrizione è un po' stretta, allargiamola trascinando-la per il bordo, proprio come faremmo con una colonna di un foglio elettronico. Dovremmo avere (figura 35):



Id	Descrizione
----	-------------

Illustrazione 35 Definizione seconda colonna

Benissimo, siamo alla fine, una bella pressione sul pulsante “**bozza**”, et voilà (figura 36):



Id	Descrizione
2	Classica
15	Fantasy
4	Filosofia
5	Gialli
10	Giochi
16	Linux
3	Pop
9	Reti Locali
1	Rock
6	Saggio
8	Samba
7	Sistemi Operativi
* > automatico >	

Record di dati 1 da 12

Illustrazione 36 Risultato finale

Abbiamo creato il nostro primo formulario. Tutto qui ? Ma non è la stessa cosa del pannello destro delle sorgenti dati ?? In effetti così sembra, ma NON E' AFFATTO la stessa cosa. Ci sono molte importanti differenze, e ne cito ora solo due, tanto per cominciare. Primo, i formulari sono completamente personalizzabili per quanto riguarda la formattazione, e questa non viene persa una volta chiuso il documento. Secondo, in un formulario si possono specificare una serie di parametri (come il “sola lettura”) che nel pannello destro delle sorgenti dati non sono applicabili.

Tips



Date uno sguardo alla colonna “Id” del nostro formulario. Abbiamo definito l'Id come un intero ad incremento automatico. Allora, perché alcuni numeri (ad es. 11) non compaiono ? Perché se si cancella una riga, il numero assegnato non viene più riutilizzato; quindi se cancello il record 11, il motore riparte dal 12.

Tornando in modalità “bozza”, è possibile variare molti parametri relativi alla visualizzazione dei dati, quindi possiamo abbellire il formulario secondo i nostri gusti. Trattandosi di un documento di testo, ad inizio pagina possiamo anche assegnare un bel titolo, magari “Tabella degli Argomenti”, e salvare il documento. Non ci resta che preparare un bel formulario anche per la Tabella “Supporti”, che dovrebbe apparire, alla fine, più o meno così (figura 37):



	Id	Descrizione
▶	6	Articolo
	3	CD Audio
	4	CD Software
	2	DVD Audio
	1	DVD Video
	5	Libro
	7	Rivista
✱	automatico>	

Illustrazione 37 Formulario supporti

In questo caso ho usato il **pilota automatico**, che è raggiungibile alla voce *File -> Pilota Automatico -> Formulario*; questa auto composizione rende abbastanza lineare la creazione di formulari non molto complessi, e può essere usata senza problemi se si ha fretta. Inoltre nulla vieta di modificare in seguito i formulari generati dall'auto composizione secondo le nostre preferenze.

Tips



Il passaggio dalla modalità di modifica della struttura del formulario alla modalità dei gestione dei dati avviene sempre con la pressione del pulsante “bozza”. E' l'equivalente del passaggio in Ms Access da vista “struttura” a vista “scheda”, per quelli di voi che hanno confidenza col prodotto Microsoft.

Forse è il caso di ricapitolare i vari passaggi che ci hanno portato fin qui:

- Abbiamo creato la struttura dei nostri archivi con **MySqlCC**, nell'Archivio Mediateca
- Abbiamo generato un **DSN** (Data Source Name) per la gestione dell'archivio tramite ODBC
- Abbiamo collegato OOo con il DSN mediante una “**sorgente dati**”
- Abbiamo creato due “**ricerche**” relative alle tabelle *tbargomenti* e *tbsupporti*, scartando il campo di tipo “timestamp” ed ordinate per Descrizione (*rh_argomenti* e *rh_supporti*)
- infine abbiamo creato due documenti di testo forniti di “**formulario**” per la gestione delle due tabelle.

6.3 Il Formulario per la Mediateca

Siamo arrivati al momento di costruire il modulo di gestione del nostro archivio principale, cioè **tbmedia**. Per spiegare meglio le varie possibilità disponibili, potremmo usare **due tipi di formulario diversi**: il primo più semplice, basato su una struttura a tabella simile a quelle già viste. Il secondo più complesso, che mostrerebbe una struttura mista scheda / tabella, più adatta all'immissione dei dati. In questa pubblicazione descriveremo solo il primo, rimandando gli approfondimenti alla seconda parte.

Cominciamo col creare una nuova ricerca (**r_media**) che sarà la base del nostro formulario. Includiamo ovviamente solo i campi che ci interessano, quindi (figura 38):

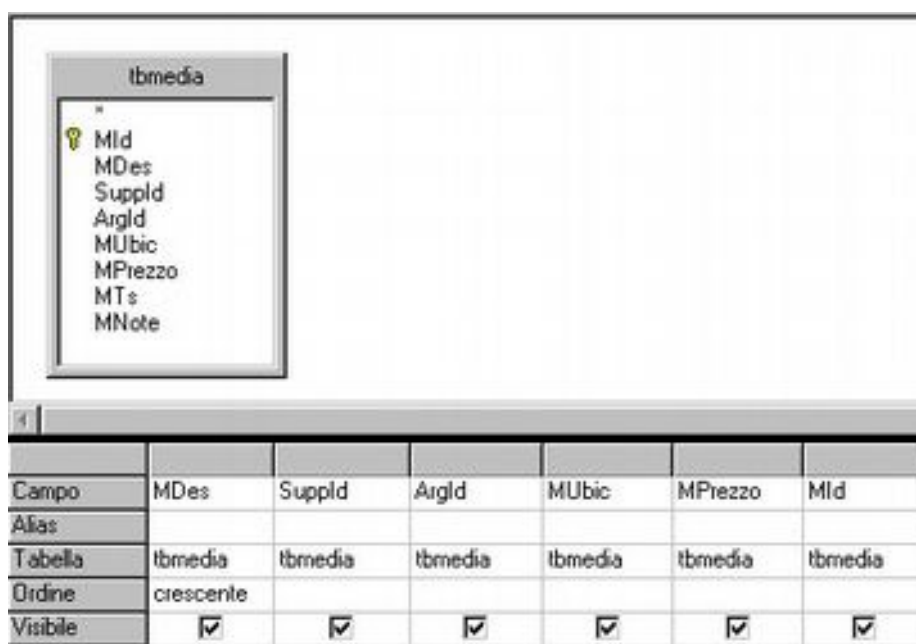


Illustrazione 38 Definizione ricerca mediateca

Abbiamo scelto di ordinare la ricerca per descrizione, e di includere anche il campo MIId (la chiave primaria). A rigore questo campo (intero ad incremento automatico) non era necessario, però **se non si include la chiave primaria le ricerche di OOo risultano di sola lettura**, cioè non si possono modificare i dati. I Campi MTs (cioè il timestamp) e MNote in questa fase non ci interessano. Crea la ricerca, in un nuovo documento di testo vuoto generiamo il formulario seguendo le procedure che abbiamo già visto (manualmente o col pilota automatico, fate voi). Dovremmo ottenere più o meno questo risultato (figura 39):

	Descrizione	Tipo Supporto	Argomento	Ubicazione	Prezzo
	Bad Boys II	0	0		€ 0,00
	Benni - La compagnia dei celestini	0	0		€ 0,00
	Guccini - Ritratti	0	0		€ 0,00
	Joss Stone - The Soul Session	0	0		€ 0,00
	KDE 3.2 - Novità Major Release	0	0		€ 0,00
	Montalbano - Il Ladro di merendine	0	0		€ 0,00
		0	0		€ 0,00

Illustrazione 39 Risultato della ricerca

Per un corretto funzionamento del formulario è necessario impostare, nelle proprietà delle colonne “Tipo Supporto”, “Argomento”, “Prezzo”, la proprietà “**Valore Standard**” a zero (0), per fare in modo che venga assegnato in automatico un valore. La Colonna “Prezzo” è stata definita come “**Campo Formattato**”, modo da poter prevedere l'indicazione del simbolo dell'euro. Comunque ancora non ci siamo.

Infatti immettere a mano il valore numerico di corrispondenza per la Tabella “Tipo Supporto” e “Argomento” certo non è accettabile. Sarebbe meglio usare una bella casella a discesa, da cui selezionare in campo che ci interessa. Dobbiamo perciò cambiare ancora qualcosa. Torniamo in modo “bozza”, selezioniamo ad esempio la colonna “Argomento” e con il tasto destro del mouse scegliamo “Sostituisci con...” e poi “Casella di riepilogo”. Sempre col tasto destro selezioniamo “Colonna...” e nella tab dati impostiamo questi valori (figura 40):



Illustrazione 40 Modifica delle proprietà della ricerca

Con questa impostazione, stiamo dicendo ad OOo che desideriamo compilare il Campo Dati “ArgId” attraverso il riferimento ad una ricerca (“rh_argomenti”), ed il campo collegato (cioè il valore da trasferire al campo dati) è il **SECONDO** della ricerca (cioè, appunto, ArgId). Infatti i campi della ricerca che desideriamo collegare sono **numerati a partire da zero**, e quindi 0 rappresenta “ArgDes”. In verità questa impostazione non è molto logica, ed anche un po' limitata, perché ad esempio nella casella a discesa potrò mostrare sempre solo una colonna. La regola è dunque che, a seconda del valore impostato in “campo collegato”, nella casella a discesa sarà mostrata la colonna A SINISTRA del campo scelto, cominciando a contare le colonne da zero. Per cui siccome nella mia ricerca ho solo due colonne “ArgDes” ed “ArgId”, sarà mostrata “ArgDes” e sarà trasferito il valore di “ArgId”, cioè appunto quello che volevamo ottenere. Adesso dovrebbe essere anche chiaro perché abbiamo posizionato ArgDes come prima colonna della ricerca. Questo meccanismo non è molto ben spiegato nella guida in linea, e si ci arriva con un po' di tentativi. Ricordatevi anche di salvare il documento dopo aver fatto le modifiche, in caso contrario le caselle a discesa sembreranno non funzionare bene. Comunque alla fine tutto si risolve, e, applicando la stessa procedura anche a “Tipo Supporto”, avremo (figura 41)

	Descrizione	Tipo Supporto	Argomento	Ubicazione	Prezzo
	Bad Boys II	DVD Video	Gialli		€ 0,00
	Benri - La compagnia dei celestini	Libro	Romanzo		€ 0,00
	Guccini - Ritratti	CD Audio	Pop		€ 0,00
	Il Signore degli Anelli	DVD Video	Fantasy		€ 0,00
	Joss Stone - The Soul Session	CD Audio	Rock		€ 0,00
	KDE 3.2 - Novità Major Release	Articolo	Linux	Linux & C. - 38	€ 0,00
	Montalbano - Il Ladro di merendine	Libro	Gialli		€ 0,00

Illustrazione 41 Risultato finale del formulario relativo la mediateca

Abbiamo costruito un formulario per la nostra mediateca che:

- è ordinato per descrizione, quindi la ricerca di un titolo è agevole
- ci permette di caricare i dati in modo comodo, con caselle a discesa per i campi che fanno riferimenti ad altre tabelle
- è facilmente navigabile

Possiamo, anche in questo caso, scrivere un bel titolo per il documento (ad es. **MEDIATECA**) e salvare. **Se desideriamo che il documento di testo non sia modificabile, cioè vogliamo che l'utente si limiti ad interagire col database, la soluzione migliore è assegnare, da sistema operativo, l'attributo di “sola lettura” al file.** In questo modo i dati del formulario saranno accessibili, ma tutto il resto (compresa la possibilità di passare in modo bozza) risulterà bloccato.

Tips



ATTENZIONE: un **bug** in OOo fa “sparire” a volte i valori contenuti nelle caselle a discesa, sia quando si cambia lo ZOOM, sia quando si riapre il documento. Si tratta di un bug molto fastidioso, che si risolve utilizzando il pulsante “**aggiorna**” nella barra degli strumenti del Database, posizionata in basso.

6.4 Qualche informazione in più



OOo ha altre interessanti caratteristiche quando adoperiamo un formulario. Innanzi tutto vi sarete accorti che il “Campo di Controllo Tabella” ha alla base una piccola barra degli strumenti.



Questa barra è del tutto simile a quella che compare nelle schede di Ms Access, e serve a navigare nell'archivio. L'ultimo pulsante sulla destra permette anche l'immissione immediata di un nuovo record. Quella più interessante è, però, la barra dei pulsanti che compare alla base della finestra di OpenOffice.org quando usiamo il formulario (figura 42).



Illustrazione 42 Barra degli strumenti del database

Oltre ai pulsanti già visti, questa barra offre una serie di altre possibilità. In primo luogo il pulsante “**aggiorna**”  che permette di “ricaricare” i dati del formulario, e ci è utile quando OOo non mostra, come abbiamo visto, le nostre caselle a discesa. Poi i pulsanti di ordinamento, non molto sfruttati: con i formulari basati su ricerche ordinate, infatti, non funzionano. Il più utile è senz'altro quello di “**Filtro Automatico**”,  che permette di selezionare molto velocemente i dati sulla base del contenuto di un campo. Funziona così: basta portare il cursore sul campo che contiene il valore da selezionare, e premere il pulsante. Ad esempio, se voglio selezionare tutti i CD Audio del mio archivio, posiziono il cursore sul campo “supporto” in una riga che contiene “Cd Audio” ed applico il filtro automatico. OOo mi mostrerà tutti e solo i Cd Audio. Per tornare alla situazione precedente, basta usare il pulsante “**Rimuovi fil-**

tro/ordine”, che si trova tre posizioni più a destra. Questa barra prevede molte altre possibilità, e vi consiglio di approfondire l'argomento con l'aiuto della Guida di OpenOffice.org.

7 Gestione dei Report

Con la parola **Report** intendiamo la produzione di “riassunti” del contenuto di un archivio, in forma di solito non modificabile, e formattati per la stampa. Quindi una cosa diversa dal formulario, che invece è orientato alla modifica dei dati. Dalla versione 1.1, OpenOffice.org include una auto composizione dei Report, ma il modulo è immaturo e necessita di un'ampia revisione. Il report è un aspetto molto interessante ed utile della gestione di un Archivio; nell'ottica di OOo, direi che è lo strumento più utile per accedere a dati che non sono gestiti direttamente dal programma. Supponiamo infatti di avere un Database esterno, proveniente dalla procedura gestionale dell'Azienda; potrei semplicemente voler stampare tutti i Clienti di un certo Agente con un fatturato minore di una soglia arbitraria. E' una tipica situazione dove usare OOo, senza scomodare il sistemista che si occupa del gestionale. Inoltre, disponendo di un accesso in sola lettura, sicuramente non posso fare danni.

Dunque la possibilità di generare Report è importante, ma come fare, visto che il modulo di OOo funziona piuttosto male ? Semplice, basta usare **Calc**.

7.1 Calc per generare e stampare Report

Calc non è molto utile nella gestione dei formulari, infatti il Writer si presta meglio. Calc però possiede una utile caratteristica: è possibile collegare alle celle del foglio una Tabella o una Ricerca, e fare in modo che ogni modifica ai dati si rifletta nel foglio stesso. Spieghiamoci meglio: non è possibile modificare i dati direttamente dalle celle di Calc (o meglio le modifiche non vengono trasferite all'archivio), ma ogni modifica eseguita “a monte” può essere riportata nel foglio elettronico. Questo significa che, una volta creato il “link”, anche se alla tabella sono state aggiunte 100 righe, le ritroveremo pari pari nel nostro report. Inoltre l'area di “collegamento” può essere formattata a nostro piacimento, e nel report è possibile aggiungere intestazioni e piè di pagina esattamente come in un normale foglio elettronico.

Quello che faremo sarà dunque creare una “base” per i dati che intendiamo comprendere nel report, e collegare un foglio elettronico alla “base”. A titolo di esempio genereremo un report per il contenuto della nostra mediateca, e come base useremo una ricerca. Al lavoro.

7.2 La Ricerca per il Report – Relazioni tra Tabelle

Vogliamo quindi una bella lista in ordine alfabetico di tutti i contenuti della nostra Mediateca, e vogliamo che sia dinamica, cioè che ad ogni apertura del Report i dati risultino sempre aggiornati. Generiamo una nuova Ricerca, contenente le Tabelle `tbmedia`, `tbargomenti` e `tbsupporti`. Fino ad ora abbiamo usato solo Ricerche con una sola Tabella, e quindi è arrivato il momento di vedere come si usano le **Relazioni tra Tabelle**. La Ricerca (che chiamiamo `st_media`) dovrebbe contenere la struttura ed in dati presenti in figura 43.

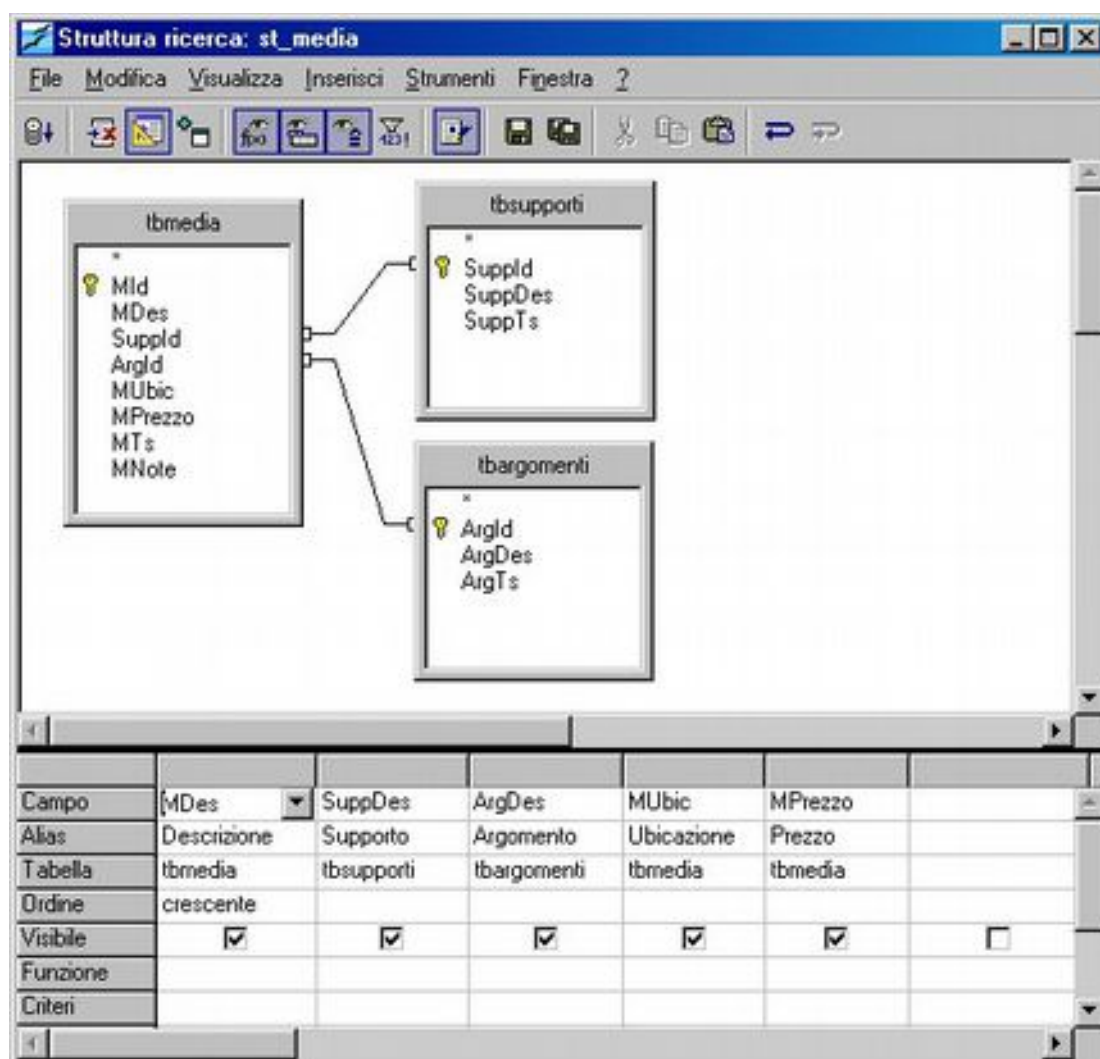


Illustrazione 43 Definizione di una ricerca complessa

Come vedete la Ricerca contiene campi provenienti da più tabelle, per fare in modo che ad ogni riga corrisponda la Descrizione corretta dei Supporti e degli Argomenti. Per ottenere questo risultato è necessario dire ad OOo come “collegare” le varie tabelle, cioè indicare quali campi in comune il programma deve considerare per costruire la ricerca. In pratica devo spiegare ad OpenOffice.org che *il campo SuppId della Tabella tbmedia corrisponde al campo SuppId della tabella tbsupporti, quindi, se il valore è ad esempio 3, la descrizione da riportare è 'Cd Audio'*. Il nome tecnico di questa operazione è “**join**”: sto eseguendo un **join tra tabelle**.

In OOo ottengo il risultato “**trascinando**” con il mouse il campo *SuppId* di *tbmedia* sul campo *SuppId* di *tbsupporti*. La stessa operazione devo fare per il campo *ArgId*. Quindi sceglierò i campi che dovranno apparire nella ricerca; notate che assegno anche degli “**alias**”, cioè delle intestazioni di colonna più leggibili, ed, infine, che il campo *Mdes* è ordinato.

Il risultato della Ricerca è più o meno quello in figura 44:



	Descrizione	Supporto	Argomento	Ubicazione	Prezzo
	Bad Boys II	DVD Video	Gialli		0,00
	Benni - La compagnia dei celestini	Libro	Romanzo		0,00
	Cornwell - L'Ultimo Distretto	Libro	Gialli		0,00
	Guccini - Ritratti	CD Audio	Pop		0,00
	Il Signore degli Anelli	DVD Video	Fantasy		0,00
	Joss Stone - The Soul Session	CD Audio	Rock		0,00
	KDE 3.2 - Novità Major Release	Articolo	Linux	Linux & C. - 38	0,00
	Montalbano - Il Ladro di merendine	Libro	Gialli		0,00
	Montalbano - La forma dell'acqua	Libro	Gialli		0,00
	Pearl Jam - Ten	CD Audio	Rock		0,00

Illustrazione 44 Risultato della ricerca

Se, nella struttura ricerca, faccio un doppio click sulla linea che collega i due campi, avrò come risultato (figura 45):

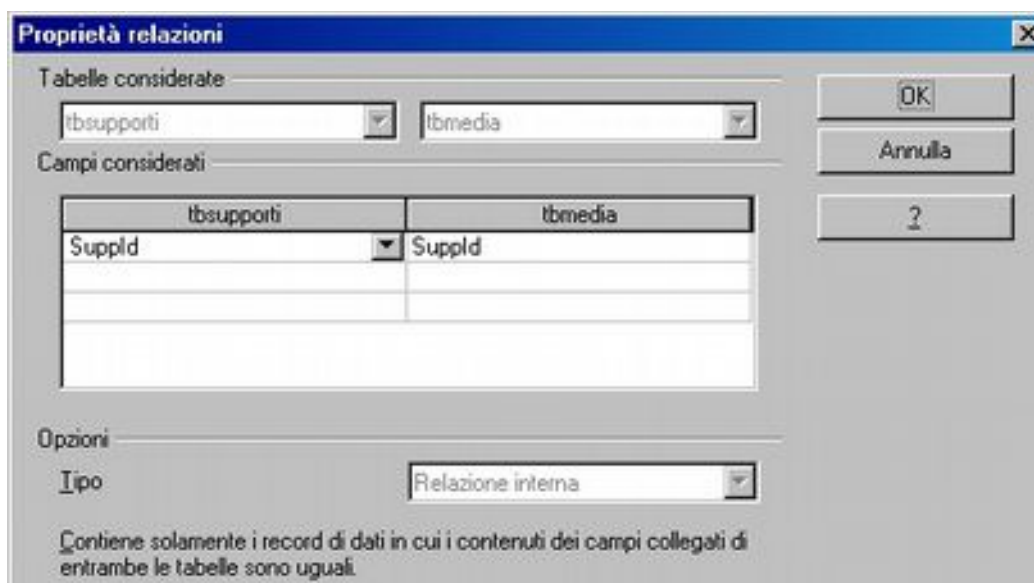


Illustrazione 45 Proprietà della relazione

Cioè una finestra riepilogativa della Relazione creata. Notate che:

1. Una Relazione può contenere anche più Campi
2. Esistono vari tipi di relazione, ma per Tabelle MySQL da questa maschera è possibile avere solo una “relazione interna” (in questo caso la cosa è influente, in altri casi è molto importante, ma è un argomento che non tratteremo in questo contesto)

Infine, diamo una occhiata a come OOo trasforma la nostra ricerca in linguaggio SQL, premendo il pulsante “Attiva/Disattiva vista Disegno”, nella barra superiore:

```

SELECT `tbmedia`.`MDes` AS `Descrizione`, `tbsupporti`.`Supp-
Des` AS `Supporto`, `tbargomenti`.`ArgDes` AS `Argomento`,
`tbmedia`.`MUbic` AS `Ubicazione`, `tbmedia`.`MPrezzo` AS
`Prezzo`

FROM `mediateca`.`tbsupporti` `tbsupporti`, `mediateca`.`tb-
media` `tbmedia`, `mediateca`.`tbargomenti` `tbargomenti`

WHERE ( `tbsupporti`.`SuppId` = `tbmedia`.`SuppId` AND `tbar-
gomenti`.`ArgId` = `tbmedia`.`ArgId` )

ORDER BY `Descrizione` ASC

```

La parola chiave **SELECT** indica i campi da selezionare e gli alias, **FROM** indica quali tabelle usare, **WHERE** stabilisce la relazione tra le tabelle, **ORDER BY** l'ordine di visualizzazione. Ci sarebbero alcune cose da dire sulla “traduzione” in SQL della ricerca, e di questo ci occuperemo nella seconda parte di questo documento. Per adesso accontentiamoci di capire come OOo estrae i dati da un motore di Db come MySQL: semplicemente “spedisce” un comando SQL ed attende la risposta. Ma è il momento di passare al Calc.

7.3 Il Report in Calc

In effetti le cose sono molto più semplici di quanto pensate. Creiamo un nuovo documento Calc, apriamo le sorgenti dati con F4, selezioniamo la nostra ricerca st_media e “trasciniamola” nella prima casella del foglio elettronico. Avremo questo risultato (figura 46):

	A	B	C	D	E	F
1	Descrizione	Supporto	Argomento	Ubicazione	Prezzo	
2	Bad Boys II	DVD Video	Gialli		€0,00	
3	Benni - La cò	Libro	Romanzo		€0,00	
4	Cornwell - L'U	Libro	Gialli		€0,00	
5	Guccini - Rit	CD Audio	Pop		€0,00	
6	Il Signore deg	DVD Video	Fantasy		€0,00	
7	Joss Stone -	CD Audio	Rock		€0,00	
8	KDE 3.2 - No	Articolo	Linux	Linux & C. -	€0,00	
9	Montalbano -	Libro	Gialli		€0,00	
10	Montalbano -	Libro	Gialli		€0,00	
11	Pearl Jam -	CD Audio	Rock		€0,00	
12						

Illustrazione 46 Trascinamento di una ricerca su di un foglio di calcolo

Cioè la Ricerca riportata per righe e colonne nel foglio elettronico. Bene, dal menù dati scegliamo “Definisci Area”. Noterete che è stata creata una nuova area di nome Importa1; selezioniamola e premiamo il pulsante EXTRA.

Come in Figura, sarà necessario settare tutte e quattro le *opzioni*. In particolare “*Contiene Intestazioni Colonne*” viene abilitato in automatico. “*Inserisci/Elimina Celle*” dice ad

OpenOffice.org di aggiornare l'Area in base alle modifiche dei Dati usati come sorgente (quindi aggiungere o togliere righe). “*Mantieni Formattazione*” significa che tutte le opzioni di formattazione eseguite sul Foglio Elettronico saranno mantenute nel tempo. “*Non salvare i dati*

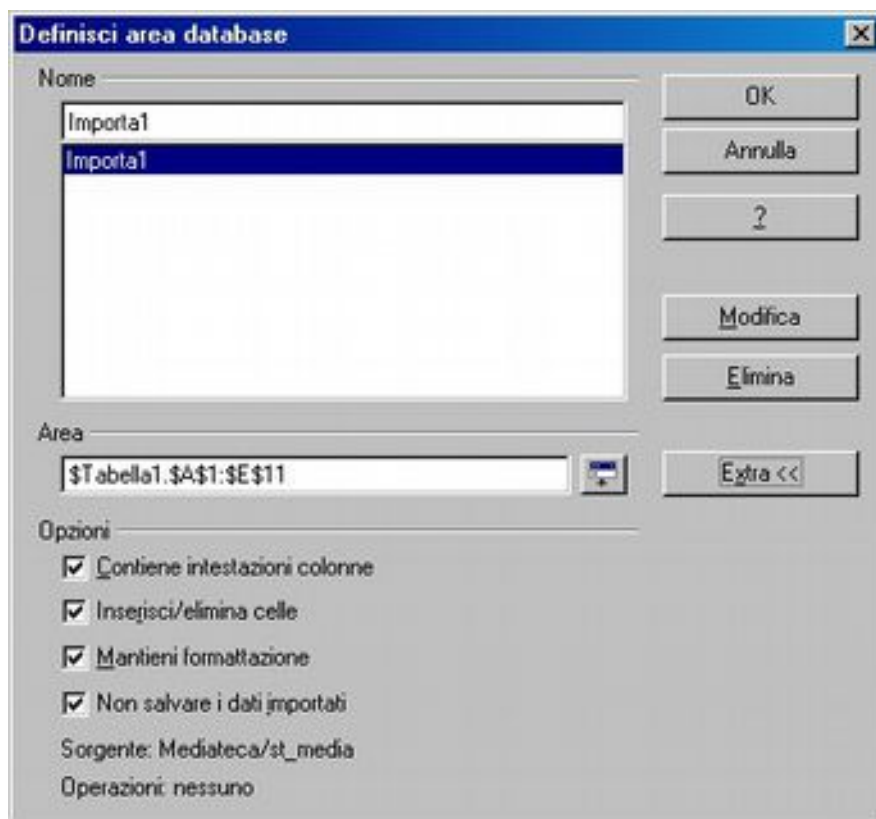


Illustrazione 47 Definizione area database in Calc

importati” serve ad evitare che ogni volta i dati siano salvati insieme al Foglio (in questo caso si salva solo il “collegamento” alla tabella). A seconda delle esigenze, la seconda e la quarta opzione possono essere disabilitate, se ad esempio desideriamo che il Foglio contenga un'immagine “statica” degli archivi, magari da spedire per posta elettronica.

	A	B	C	D	E
1	Descrizione	Supporto	Argomento	Ubicazione	Prezzo
2	Bad Boys II	DVD Video	Gialli		€ 0,00
3	Benni - La compagnia dei celestini	Libro	Romanzo		€ 0,00
4	Cornwell - L'Ultimo Distretto	Libro	Gialli		€ 0,00
5	Guccini - Ritratti	CD Audio	Pop		€ 0,00
6	Il Signore degli Anelli	DVD Video	Fantasy		€ 0,00
7	Joss Stone - The Soul Session	CD Audio	Rock		€ 0,00
8	KDE 3.2 - Novità Major Release	Articolo	Linux	Linux & C. - 38	€ 0,00
9	Montalbano - Il Ladro di merendine	Libro	Gialli		€ 0,00
10	Montalbano - La forma dell'acqua	Libro	Gialli		€ 0,00
11	Pearl Jam - Ten	CD Audio	Rock		€ 0,00

Illustrazione 48 Formattazione di un report in Calc

Ad ogni riapertura del Documento, OOo ci richiederà se vogliamo aggiornare le ricerche (bisogna ovviamente rispondere di sì). Se invece, a foglio aperto, desideriamo ricaricare “ma-

nualmente” i Dati in modo da avere sotto mano le ultime modifiche eseguite sull'archivio, basterà usare la voce del menù *Dati->Aggiorna Area*, dopo aver selezionato una cella qualsiasi del nostro Report.

A questo punto possiamo formattare l'area come meglio ci aggrada, ad esempio come in figura 48.

Un'ultima ritoccata all'area di stampa, quindi impostiamo la pagina con Intestazioni, Margini ed orientamento et voilà... il nostro Report è pronto per essere consultato e stampato quando vogliamo.

Tips



Ricordate che le modifiche sui dati che potreste eseguire nel foglio non vengono riportate nel Database, quindi sono inutili. Per modificare i dati è necessario, in OOo, un Formulario.

7.4 Report su Dati parziali

La domanda sorge spontanea.... Ma se invece vogliamo un Report che ci ritorni solo una parte dell'archivio, in base a parametri immessi dall'utente ?

La risposta è: esistono molti modi, semplici e complicati, per ottenere questo risultato. Quello più semplice è inserire manualmente un criterio nella nostra Ricerca. Ad esempio, se voglio solo i libri gialli, potrei scrivere “Gialli” nei criteri del campo “*Argomento*” della Ricerca st_media, come in figura (gli apici vengono aggiunti da OOo, se li scrivete manualmente devono essere singoli) :

Campo	MDes	SuppDes	ArgDes
Alias	Descrizione	Supporto	Argomento
Tabella	tbmedia	tbsupporti	tbargomenti
Ordine	crescente		
Visibile	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Funzione			
Criteri			'Gialli'

Illustrazione 49 Definizione di un report con criterio di ricerca

Selezionando *Dati->Aggiorna Area* dal menù, il nostro Report è pronto. Questa soluzione, per quanto semplice, manca di flessibilità perché ci costringe a modificare ogni volta la Ricerca. Potremmo impostare un “parametro” per la Ricerca, (OOo permette di definire uno o più parametri per le Query), oppure passare a qualcosa di più sofisticato. Nella seconda parte di questo documento ce ne occuperemo in dettaglio.

8 Fine del viaggio... per ora....

Questa introduzione all'uso dei Database con OpenOffice.org, finisce qui. Lo scopo era quello di illustrare in modo semplice le tecniche di base da utilizzare per accedere ai Db con OOo.

Molto, moltissimo altro ci sarebbe da dire, e questo fornisce materiale per una seconda parte. Non crediate, infatti, che OOo disponga, per la gestione dei Db, solo di quello che abbiamo visto.

Per ora potete usare questi spunti come punto di partenza per le vostre esplorazioni. L'accesso e la modifica delle basi di dati sono uno degli aspetti più interessanti dell'informatica, e su questo si basa una larga percentuale del “lavoro” dei nostri PC.

Ricordate che la fine di un viaggio è solo l'inizio di un altro, da soli oppure insieme ad altri amici; e che ogni viaggio serve a conoscere il mondo, a volte per il solo gusto di saperne di più.....

9 Appendice A - Trasferire le Sorgenti Dati

Uno dei problemi delle Sorgenti Dati è che non possono essere trasferite insieme al documento. Questo implica che su ogni PC su cui vogliamo che i nostri Report e Formulari siano funzionanti, dobbiamo ricreare manualmente la Sorgente Dati. Se questo non è eccessivamente complicato quando si collegano le Tabelle, sicuramente è molto fastidioso quando dobbiamo ricreare “a mano” tutte le Ricerche. Inoltre, se abbiamo eseguito una installazione di rete di OpenOffice.org, tutte le configurazioni sono “personali” dell'Utente, e quindi non è possibile nemmeno dividerle per tutti gli utilizzatori dello stesso PC.

I parametri delle sorgenti dati dovrebbero essere contenuti nel file:

C:\Documents and Settings\%Username%\Dati applicazioni

OpenOffice.org1.1.0\user\registry\data\org\openoffice\Office\DataAccess.xcu

per cui si potrebbe copiare questo file nel profilo di ogni utente, una volta creata la configurazione iniziale. Questo però rimuove tutte le impostazioni precedenti delle Sorgenti Dati, e quindi va bene solo se si parte da zero. Il file è un documento XLM, quindi con un po' di buona volontà si potrebbe “tagliare” la parte relativa alla Sorgente Dati che ci interessa ed “incollarla” nel file dell'utente di destinazione. Anche questo però è difficilmente realizzabile: pensate che, aprendo il file con OOo Writer, la configurazione per Mediateca, supponendo che io sia riuscito ad identificarla con esattezza, va dalla Pagina 18 alla Pagina 43.

Il mio sistema è più semplice: creo un documento di testo con la sintassi Sql di tutte le ricerche, copiando ed incollando le istruzioni dalla finestra di modifica, e col procedimento inverso mi ricreo le ricerche sul computer (o utente) di destinazione. Cioè mi costruisco un documento con contenuti di questo tipo:

rh_argomenti

```
SELECT `ArgDes`, `ArgId` FROM `mediateca`.`tbargomenti`
`tbargomenti` ORDER BY `ArgDes` ASC
```

In questo modo, ricreare almeno le ricerche è abbastanza immediato.

Sarebbe anche possibile usare l'utility **pkgchk**, presente nei binari di OOo, che serve a trasferire parti di configurazione tra due macchine. L'utilizzo del programma non è però immediato, e chi volesse approfondire può consultare la documentazione disponibile sul sito di OpenOffice.org.

10 Appendice B – Schema del Database di Esempio

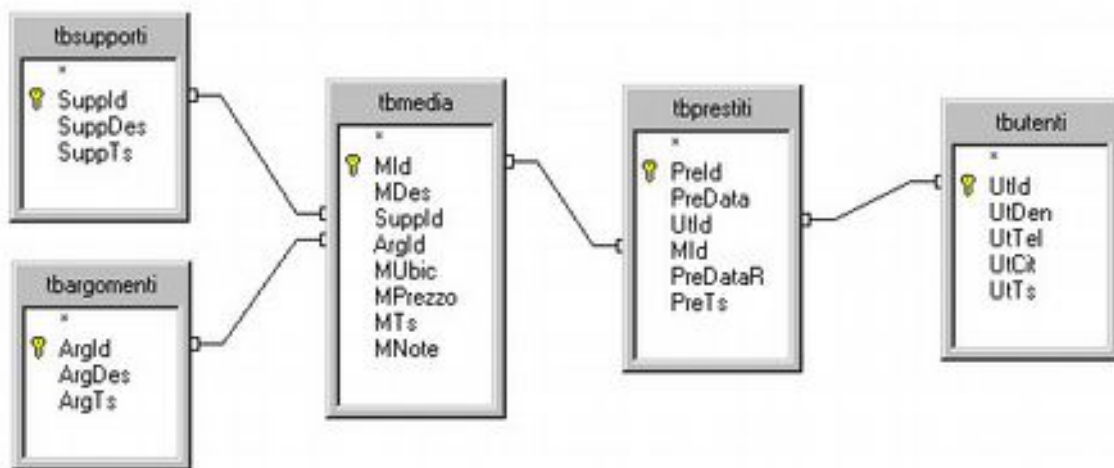


Illustrazione 50 Schema database di esempio con relazioni fra tabelle

11 Appendice C – Tipo di Dati di MySql

Riporto in questa appendice un riassunto dei tipi di dati utilizzabili con MySql. Vi ricordo che con **M** si indica il numero di caratteri (massimo 255) utilizzati per la visualizzazione. Con **D**, per i campi numerici Decimali, si indica appunto il numero di cifre dopo la virgola: il massimo è 30, ma non può essere ovviamente più grande di M-2. Con **UNSIGNED** si indica che il numero può essere solo positivo. Con **ZEROFILL** si chiede al motore di Db di completarlo con cifre 0 a sinistra fino alla lunghezza massima. Ricordate che se un numero viene definito **ZEROFILL**, diventa anche **UNSIGNED**.

11.1 Valori di Tipo Numerico

TINYINT [(M)] [UNSIGNED] [ZEROFILL]

Il numero Intero più piccolo, con un range da -128 a 127. Il valore unsigned va da 0 a 255.

BIT

BOOL

BOOLEAN

Queste definizioni sono sinonimi di **TINYINT(1)**. Il tipo **BOOLEAN** è stato aggiunto dalla versione 4.1.0.

SMALLINT [(M)] [UNSIGNED] [ZEROFILL]

Numero Intero breve. Il range va da -32.768 a 32.767. Il valore unsigned da 0 a 65.535.

MEDIUMINT [(M)] [UNSIGNED] [ZEROFILL]

Numero Intero Medio. Il range va da -8.388.608 a 8.388.607. Il valore unsigned da 0 a 16.777.215.

INT [(M)] [UNSIGNED] [ZEROFILL]

Valore Intero “classico”. Il range va da -2.147.483.648 to 2.147.483.647. Il valore unsigned da 0 a 4.294.967.295.

INTEGER[(M)] [UNSIGNED] [ZEROFILL] – Sinonimo per **INT**.

BIGINT [(M)] [UNSIGNED] [ZEROFILL]

Intero lungo. Il range va da -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807. Il valore unsigned da 0 a 18.446.744.073.709.551.615.

FLOAT(precisione) [UNSIGNED] [ZEROFILL]

Numero a virgola mobile. **precisione** può assumere un range da 0 a 24 per valori a precisione singola, e da 25 a 53 per valori a precisione doppia. Questi tipi di dati sono equivalenti a **FLOAT** e **DOUBLE** descritti subito dopo. **FLOAT(precisione)** possiede lo stesso range dei tipi corrispondenti **FLOAT** e **DOUBLE**, ma il numero di cifre da visualizzare ed il numero di decimali non sono definiti. Notate che l'uso di **FLOAT** potrebbe causare alcuni problemi,

perchè i calcoli sono eseguiti sempre in precisione doppia. In effetti questa tipologia è prevista solo per compatibilità con ODBC.

FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]

Numero a virgola mobile in precisione singola. Il range va da **-3.402823466E+38** a **-1.175494351E-38**, **0**, e **1.175494351E-38** fino a **3.402823466E+38**. Se viene specificato **UNSIGNED**, i valori negativi non sono permessi.

DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]

Numero a virgola mobile in precisione doppia. I valori consentiti sono da **-1.7976931348623157E+308** a **-2.2250738585072014E-308**, **0**, e da **2.2250738585072014E-308** a **1.7976931348623157E+308**. Se viene specificato **UNSIGNED**, i valori negativi non sono permessi.

DOUBLE PRECISION[(M,D)] [UNSIGNED] [ZEROFILL]

REAL[(M,D)] [UNSIGNED] [ZEROFILL]

Sinonimi per **DOUBLE**.

DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]

Un numero “esplicito”, con numero di cifre decimali fisse. Si mostra come una colonna di tipo **CHAR**: “esplicito” significa che il numero viene memorizzato come una stringa (quindi non trasformato), usando un carattere per ogni cifra del valore. Il punto decimale ed il carattere “-” per i numeri negativi non sono compresi nel valore **M**, ma il loro spazio è riservato automaticamente. Se **D** è zero, i valori non hanno né punto né cifre decimali. Il range massimo per **DECIMAL** è lo stesso del tipo **DOUBLE**, ma ovviamente può essere influenzato dalla scelta di **M** e **D**. Se viene specificato **UNSIGNED**, i valori negativi non sono permessi. Se **D** viene omesso, il default è 0. Se **M** è omesso, il default è 10.

DEC[(M[,D])] [UNSIGNED] [ZEROFILL]

NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL]

FIXED[(M[,D])] [UNSIGNED] [ZEROFILL]

Queste definizioni sono sinonimi di **DECIMAL**. Il tipo **FIXED** è stato aggiunto dalla versione 4.1.0 per compatibilità con altri servers.

11.2 Range dei valori ammessi per il tipo numerico intero

Tipo	Bytes	Valore Minimo	Valore Massimo
TINYINT	1	-128	127
SMALLINT	2	-32.768	32.767
MEDIUMINT	3	-8.388.608	8.388.607

Tipo	Bytes	Valore Minimo	Valore Massimo
INT	4	-2.147.483.648	2.147.483.647
BIGINT	8	-9.223.372.036.854.775.808	9.223.372.036.854.775.807

11.3 Valori di Tipo Date e Time

DATE

Valore Data. Il range (in anno,mese,giorno) va da '1000-01-01' a '9999-12-31'. MySQL mostra i valori **DATE** nel formato 'AAAA-MM-GG', ma permette di assegnare i valori alle colonne di tipo **DATE** usando sia stringhe che numeri.

DATETIME

Una combinazione di valori **DATE** (Data) e **TIME** (cioè orario). Il range va da '1000-01-01 00:00:00' a '9999-12-31 23:59:59'. MySQL mostra i valori **DATETIME** nel formato 'AAAA-MM-GG HH:MM:SS', ma permette di assegnare i valori alle colonne di tipo **DATETIME** usando sia stringhe che numeri.

TIMESTAMP [(M)]

Un valore di tipo timestamp. Il range è '1970-01-01 00:00:00' fino ad un momento indefinito dell'anno 2037. Una colonna **TIMESTAMP** torna utile per archiviare il momento di una operazione di **INSERT** o **UPDATE** sul record. La prima colonna **TIMESTAMP** di una Tabella è automaticamente riempita con la data e l'orario della modifica più recente eseguita sul record stesso, se non viene specificato un altro valore dall'utente. Potete inoltre riempire una qualsiasi colonna **TIMESTAMP** alla data e ora corrente semplicemente assegnando un valore **null**. L'argomento **M** indica solo come la colonna **TIMESTAMP** viene visualizzata, e non la lunghezza del valore archiviato, che è sempre 4 bytes.

TIME

Un valore orario. Il range va da '-838:59:59' to '838:59:59'. MySQL mostra i valori **TIME** nel formato 'HH:MM:SS', ma permette di assegnare i valori alle colonne usando sia stringhe che numeri.

YEAR [(2 | 4)]

Un valore che indica un Anno, nel formato a due o a quattro cifre. Il default è quattro cifre. I valori permessi vanno dal 1901 al 2155, nel formato a quattro cifre, e da 70 a 69, che rappresentano gli anni dal 1970 al 2069, nel formato a due cifre. MySQL mostra le colonne di tipo **YEAR** nel formato AAAA, ma permette di assegnare i valori alle colonne usando sia stringhe che numeri.

11.4 Valori di Tipo Stringa

[NATIONAL] CHAR(M) [BINARY | ASCII | UNICODE]

Una stringa di lunghezza fissa, che viene comunque riempita a destra di spazi, dal motore di Db, fino a raggiungere la lunghezza massima. **M** rappresenta il numero massimo di caratteri

che la colonna può contenere. Il range di **M** va da 0 a 255 caratteri. Gli spazi aggiunti vengono rimossi quando si accede al valore memorizzato. I valori **CHAR** vengono ordinati e comparati **NON** tenendo conto delle maiuscole e minuscole (**case-insensitive**), in accordo con il Set di caratteri di default, a meno che non venga specificata la parola chiave **BINARY**. Dalla versione 4.1.0, una colonna di tipo **CHAR** con una lunghezza specificata maggiore di 255 caratteri viene convertita nel più piccolo valore di tipo **TEXT** che può contenere stringhe di quella lunghezza. **CHAR** è una abbreviazione di **CHARACTER**. Il tipo **NATIONAL CHAR** (o la sua equivalente forma breve, **NCHAR**) è il modo standard per SQL di definire che una colonna di tipo **CHAR** deve usare il set di caratteri di default, e questa è l'opzione predefinita per MySQL. Dalla versione 4.1.0, l'attributo **ASCII** può essere specificato. Questa opzione assegna il set di caratteri **latin1** ad una colonna di tipo **CHAR**. Dalla versione 4.1.1 può essere specificato anche l'attributo **UNICODE**. MySQL permette la creazione di una colonna di tipo **CHAR(0)**. Ciò è utile soprattutto per motivi di compatibilità con vecchie applicazioni che prevedono l'esistenza di una colonna, ma non accedono mai ai valori contenuti nella colonna stessa.

CHAR

Sinonimo di **CHAR(1)**.

[NATIONAL] VARCHAR(M) [BINARY]

Una stringa di lunghezza variabile. **M** rappresenta il numero massimo di caratteri contenuti. Il range di **M** è compreso tra 0 e 255 caratteri. I valori **VARCHAR** vengono ordinati e comparati **NON** tenendo conto delle maiuscole e minuscole (**case-insensitive**), a meno che non venga specificata la parola chiave **BINARY**. Dalla versione 4.1.0, una colonna di tipo **VARCHAR** con una lunghezza specificata maggiore di 255 caratteri viene convertita nel più piccolo valore di tipo **TEXT** che può contenere stringhe di quella lunghezza. **VARCHAR** è una abbreviazione di **CHARACTER VARYING**.

TINYBLOB

TINYTEXT

Un valore di tipo **BLOB** o **TEXT** con una lunghezza massima di **255** ($2^8 - 1$) caratteri.

BLOB

TEXT

Un valore di tipo **BLOB** o **TEXT** con una lunghezza massima di **65.535** ($2^{16} - 1$) caratteri.

MEDIUMBLOB

MEDIUMTEXT

Un valore di tipo **BLOB** o **TEXT** con una lunghezza massima di **16.777.215** ($2^{24} - 1$) caratteri.

LOBLOB

LOBTEXT

Un valore di tipo **BLOB** o **TEXT** con una lunghezza massima di **4.294.967.295** o 4GB ($2^{32} - 1$) caratteri.

ENUM('value1', 'value2', ...)

Una lista di scelta. In sostanza un valore stringa che può avere un solo valore scelto tra una lista predefinita ('value1', 'value2', ...), oppure **NULL**, oppure il carattere speciale " che rappresenta un valore errato. Una lista **ENUM** può comprendere un massimo di 65,535 valori distinti.

SET('value1', 'value2', ...)

Un set di valori. Una stringa che può contenere zero o più valori, ognuno dei quali deve essere scelto dalla lista specificata ('value1', 'value2', ...). Un **SET** può avere al massimo 64 membri.

12 Licenza e crediti

12.1 Diritti d'autore e marchi registrati

© Copyright 2004 Filippo Cerulo – Soft.Com Sas

© Copyright 2004 Davide Dozza con riferimento alle modifiche elencate nella sezione *Modifiche ed Aggiunte*

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the appendix entitled "GNU Free Documentation License".

Una traduzione **non ufficiale** della licenza FDL è disponibile al link: <http://www.softwarelibero.it/gnudoc/fdl.it.html>.

Tutti i marchi citati all'interno di questa guida appartengono ai loro legittimi proprietari.

12.2 Sezioni Non Modificabili (no Invariant Sections)

Le sezioni Non Modificabili sono relative a:

- Il presente sotto capitolo
- le indicazioni di Copyright per le quali possono essere solo aggiunte nuove indicazioni relativamente alle parti aggiunte e modificate successivamente a questa versione
- la sezione *Modifiche ed Aggiunte* la quale possono essere solo aggiunte le descrizioni circa le parti aggiunte e modificate successivamente a questa versione
- la sezione relativa la licenza che accompagna il documento

12.3 Modifiche ed aggiunte

Maggio 2004, Davide Dozza: reimpaginazione grafica, correzione ortografica, dicitura sulle figure e relativi riferimenti incrociati.

12.4 Gnu Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete

Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this license does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate.

Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See

<http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of

"Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.