OpenOffice.org 2.0 e i Database

Introduzione all'uso dei Database con OpenOffice.org 2.0

Versione 0.99 – Dicembre 2005

Soft.Com

5. Database Server MySql

MySql (www.mysql.com) è probabilmente il più usato Database Server Open Source disponibile. La sua caratteristica principale è sempre stata la velocità di risposta, ed in nome dell'efficienza gli sviluppatori hanno probabilmente trascurato alcuni aspetti che dovrebbero essere presenti in un prodotto che aspira a fare concorrenza a nomi blasonati come *Microsoft Sql Server* o *Oracle*. La versione stabile attualmente disponibile è la 5.0.XX che introduce alcune novità largamente richieste dall'utenza, come le *stored procedure* ed i *trigger*. Mancheranno però ancora delle caratteristiche che molti ritengono essenziali, ma che gli sviluppatori di MySql valutano non compatibili con un prodotto snello ed efficiente.

Nei paragrafi che seguono ci occuperemo di alcuni aspetti interessanti del Db, utili anche in contesti non strettamente collegati ad OpenOffice.

5.1 Installazione Windows

5.1.1 Il Server

Avere in pochi minuti MySql attivo e funzionante in Windows è molto semplice. I file da scaricare dal sito www.mysql.com sono:

mysql-essential-5.0.XX-win32	il **server**, per circa 17 Mb – la versione *essential* è più che sufficiente ai nostri scopi
myODBC-3.51.XX-win	**Driver ODBC**, per circa 4 Mb (versione attuale 3.51.11)
mysql-connector-java-3.1.XX	**Driver JDBC** in formato Zip per circa 8 Mb, se si desidera connettersi al Db con Java
mysql-administrator-1.1.X-win	**MySql Administrator**, per circa 5 Mb, un tool che permette la creazione e la manutenzione dei Database e del Server MySql (versione attuale 1.1.4)
mysql-query-browser-1.1.XX-win	**MySql Query browser**, per circa 5 Mb, permette la modifica dei Dati di Database Sql ed il test delle Query

In realtà il *Query Browser* non è strettamente necessario, perché la modifica dei dati avviene di solito con programmi *client* (nel nostro caso OOo), ma torna utile in molti casi. Assolutamente indispensabile è invece *Administrator*, a meno che non vogliate impazzire con le utility a riga di comando. Tutte le operazioni di installazione vanno fatte da un utente con privilegi di amministratore. Il primo file (il **Server**) è un installer windows che è necessario lanciare nel modo tradizionale. Si avvia un Wizard e, tra le prime opzioni di installazione, la configurazione "*typical*" può andar bene nella maggior parte dei casi. La schermata successiva chiede di creare un account di registrazione su *MySql.com* ma si può evitare scegliendo "*skip*

sign-up". Alla fine il programma chiede se si desidera configurare il server appena installato, e questa è senz'altro una buona idea.

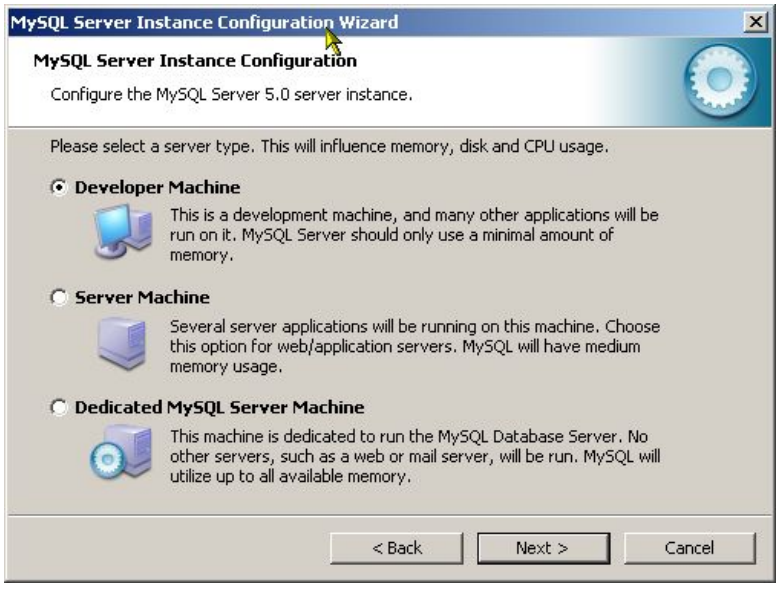


Figura 5.1.1: Configurazione dell'istanza di MySql Server

La prima scelta riguarda il tipo di configurazione (*standard* o *dettagliata*): si può selezionare ***dettagliata***, anche per farsi un'idea delle varie opzioni disponibili. Poi si passa al tipo di utilizzo che intendiamo fare di MySql (vedi figura). Di solito va bene anche "**Developer Machine**". Il passo successivo riguarda i "motori" interni da selezionare per l'archiviazione dei dati. In questo caso è opportuno selezionare "**Multifunctional Database**" in modo da disporre sia di *MyIsam* che di *InnoDb* (del cui utilizzo parleremo in seguito). Poi dobbiamo scegliere il tipo di server in funzione delle connessioni simultanee: se non ci sono esigenze particolari, va bene "**DSS**" che è la prima opzione (vedi figura).

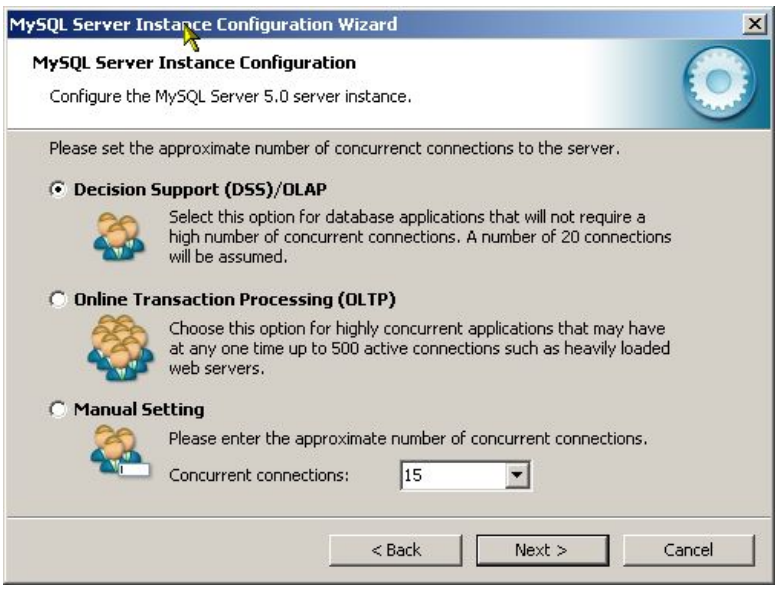


Figura 5.1.2: Numero di client previsti per MySql

Quindi si passa alla configurazione di Rete. Se intendiamo usare il Server sulla rete (come di solito accade) dobbiamo **abilitare il TCP/IP** e selezionare la **porta di ascolto** (3306 è quella standard). Dopo aver confermato il **Set di Caratteri** (quello proposto va bene), si sceglie di **avviare il Server come servizio** e (se volete usare le utility a linea di comando) di includere la **cartella Bin** nel path di Windows. Quindi è necessario impostare la **password** per l'Amministratore (che in MySql, come in Linux, si chiama **root**), scegliere se *root* può collegarsi anche dalle macchine in rete (dipende da voi, di solito si preferisce che *root* possa eseguire il logon solo sulla macchina che ospita il server), e se abilitare l'**accesso anonimo** (vivamente sconsigliato).

Figura 5.1.3: Impostazione della sicurezza in MySql

A questo punto la configurazione termina, e siete pronti ad usare il Server.

Tecnica

MySql può essere installato in Windows 2000 e XP come "***servizio***" ad esecuzione automatica. Se abbiamo scelto invece la partenza "*manuale*", per avviare il Server basta richiamare il pannello di controllo e nella sezione "*strumenti di amministrazione*", e scegliere la voce "*servizi*". Avremo una lista dei servizi installati sul PC, quindi click col tasto destro sul servizio *MySql* e dal Menu contestuale selezionare la voce "*Avvia*". Viceversa, se desideriamo che MySql non si avvii più in automatico, doppio click sul servizio *MySql* e selezionare nella casella a discesa "*Tipo di Avvio*" la voce "*Manuale*".

Il resto dei programmi citati può anche non essere installato sulla macchina che ospita il server in quanto si tratta essenzialmente di strumenti *client*. Questo significa che possono (ed a volte devono) essere caricati sui PC che si occuperanno della manutenzione del Server e delle Basi di Dati, e che eseguiranno l'accesso ai Dati stessi.

TIP

La configurazione del Server può essere modificata in qualsiasi momento con la voce *Start -> Programmi -> MySQL -> MySQL Server 5.0 -> MySQL Server Instance Config Wizard*

5.1.2 Il Driver MyODBC

Il file, l'abbiamo detto è ***myODBC-3.51.XX.msi*** (dove XX sta per il numero di versione corrente). Basta eseguire l'installazione ed avremo il nostro driver pronto all'uso. Per verificare che tutto sia andato bene, aprendo il *Pannello di Controllo -> Strumenti di Amministrazione -> Origine dati (ODBC) -> Driver* dovrebbe apparire :

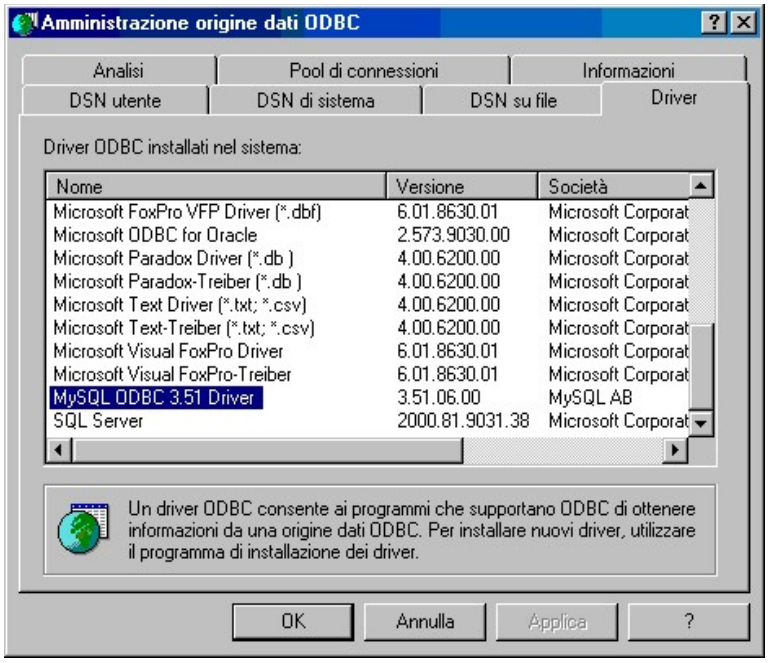


Figura 5.1.4 : Driver ODBC

5.1.3 Configurazione del DSN

Bisogna scegliere, innanzi tutto, se si desidera creare un DSN Utente (quindi valido solo per l'utente corrente) oppure di sistema (per tutti gli utenti). Quindi si preme il pulsante Aggiungi, si seleziona il driver MySql e quindi il pulsante Fine. A questo punto è possibile configurare la connessione, come in figura.

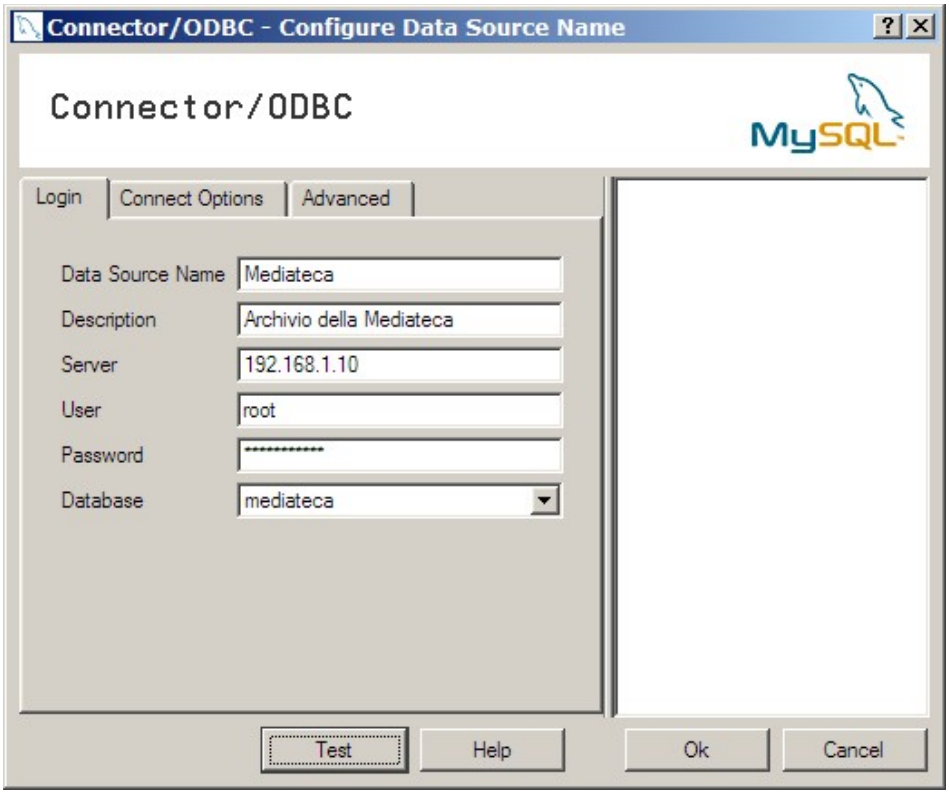


Figura 5.1.5: Configurazione del DSN per MySql

Si deve assegnare un *nome* al DSN, indicare *l'indirizzo IP* oppure il *nome* del Server, immettere le *credenziali* (nome utente e password) e, volendo, indicare un *Database* per la gestione. Se la connessione avviene sulla macchina che ospita il server, si può indicare *localhost* al posto dell'indirizzo IP. Il pulsante **Test** permette di provare la connessione. Nelle Tab *Options* ed *Advanced* è possibile impostare alcuni altri parametri, ma di solito non è necessario modificare quelli di default. Se volete approfondire, fate riferimento alla documentazione di MySQL.

5.1.4 Driver JDBC

E' necessario scaricare, dal sito di MySql, il file **mysql-connector-java-3.1.11.zip**. Una volta scompattato, copiare il file **mysql-connector-java-3.1.11-bin.jar** in una cartella a piacere (io uso *c:\programmi\JDBC*).

5.1.5 I programmi Client

Per la gestione del Server e delle Basi di Dati abbiamo già detto che sono disponibili due programmi: **Administrator** e **Query Browser**. L'installazione non comporta alcuna difficoltà e le rispettive voci compariranno nel Menu dei Programmi. Administrator installa anche una piccola utility che si chiama **System Tray Monitor** che serve anche ad avviare ed arrestare il Servizio, e che può tornare utile.

5.2 Installazione Linux

5.2.1 Il Server

Quasi tutte le moderne distribuzioni prevedono la possibilità di installare MySQL nelle sue varie versioni. Utilizzando i tool presenti nelle distribuzioni è dunque semplice avere immediatamente il server funzionante. Se proprio desiderate l'ultima versione, è disponibile anche in formato RPM sul sito di MySql.com (attenzione alle dipendenze!). In questo caso, la procedura di installazione e configurazione è ben spiegata nella documentazione ufficiale, quindi non mi dilungherò oltre.

TIP

Di solito le versioni Linux, appena installate, hanno una gestione della sicurezza particolarmente debole. Infatti la password dell'amministratore di MySql è vuota, anche se è possibile collegarsi al Database solo dalla macchina che lo ospita (*localhost*). La prima cosa da fare è dunque configurare la sicurezza.

5.2.2 Il Driver MyODBC

In molte distribuzioni il Driver è già incluso nell'elenco dei pacchetti, ma vi consiglio di scaricare l'ultima versione in formato RPM da MySql.com, nel nostro caso **MyODBC-3.51.11-2.i586**. Per funzionare il Driver ha bisogno del pacchetto ***UnixODBC***, disponibile su *Sourceforge*. Possiamo scaricare, ad esempio, **unixODBC-2.2.10-1.i386** (RPM delle librerie) e **unixODBC-gui-qt-2.2.10-1.i386** (interfaccia grafica di configurazione). La giusta sequenza di installazione prevede *unixODBC*, *unixODBC-qt* e quindi *MyOdbc*. Dopo l'installazione, il programma da lanciare è ***/usr/bin/ODBCConfig***. Sarebbe anche possibile installare il driver modificando con un editor il file *odbc.ini*, ma questo direi che esula dallo scopo di questa documentazione.

In ogni caso, il programma si presenta molto simile alla sua controparte Windows. Anche in questo caso è possibile aggiungere un *DSN* selezionando il tipo di Driver. La configurazione successiva della connessione è esattamente uguale a quella vista per i Sistema Operativo di Microsoft.

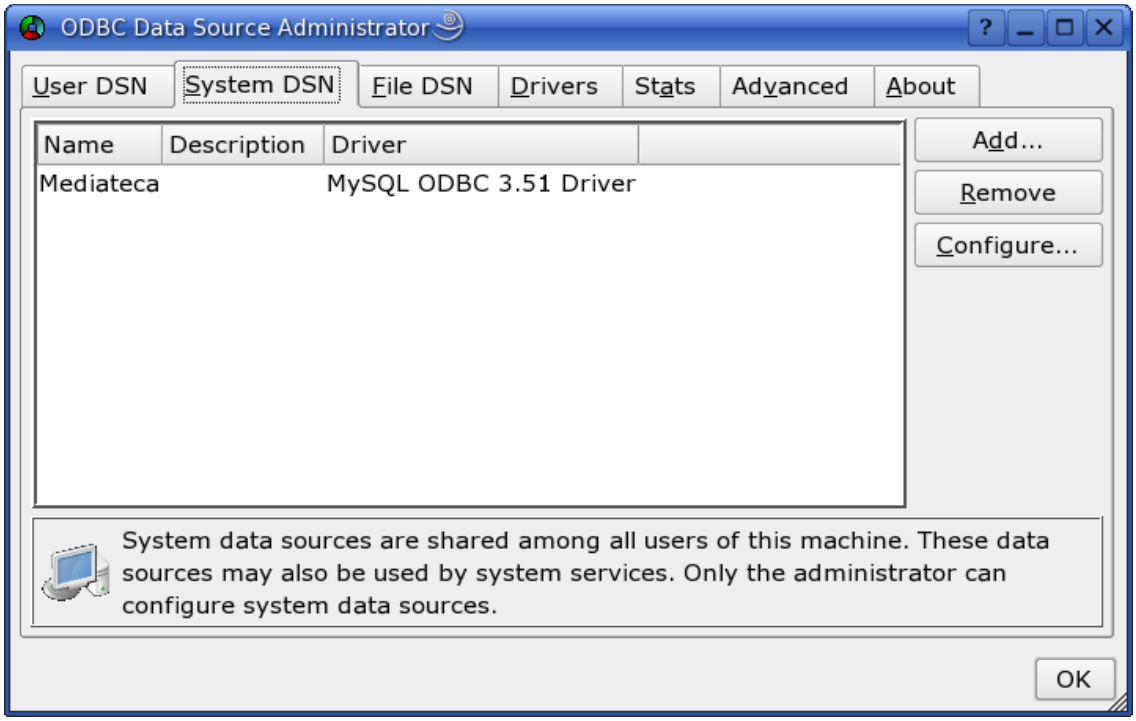


Figura 5.2.1: DSN in Linux (SUSE 9.3)

5.2.3 Il Driver JDBC

E' necessario scaricare, dal sito di MySql, il file **mysql-connector-java-3.1.11.zip**. Una volta scompattato, copiare il file **mysql-connector-java-3.1.11-bin.jar** in una cartella a piacere (io uso /opt/JBDC/).

5.2.4 I programmi Client

Dal sito MySql.com è possibile scaricare in formato RPM sia ***Administrator*** che ***Query Browser***. L'installazione, per le distribuzioni che prevedono un RPM manager è semplice, salvo problemi con le dipendenze. In alternativa, è anche possibile ricompilare dai sorgenti.

5.3 I Tipi di Dati

Ogni Server di Database possiede un lungo elenco di tipologie di Dati gestibili. In realtà i tipi più comuni (numeri, testo e date) sono molto simili in tutti i "motori" SQL. Credo però sia opportuno indicare con precisione, per ogni Server, la denominazione esatta della tipologia e le caratteristiche della informazione che può essere archiviata.

5.3.1 Campi di Tipo Stringa

MySql implementa i classici ***char*** e ***varchar***; una "*n*" davanti al tipo di campo ("***nchar***", "***nvarchar***") significa che il campo stesso usa il set di caratteri internazionali predefinito (opzione comunque attiva di default).

Tipo di Dato	*Lunghezza*	*Definizione*
char / **nchar**	Da 0 a 255	char(X) dove X è il numero di caratteri
varchar / **nvarchar**	Da 0 a 255 Da 0 a 65.535	varchar(X) dalla versione 5.0.3

5.3.2 Campi di Tipo Numerico

Qui troviamo qualche utile variante allo standard. I campi di tipo **numerico intero**, a seconda dell'intervallo di valori che possono contenere, si dividono in ***TinyInt***, ***SmallInt***, ***MediumInt***, ***Int***, ***BigInt***. Analogamente, i campi di tipo **numerico decimale** 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***. L'attributo **unsigned** elimina il segno e quindi permette la gestione di soli numeri positivi. **zerofill** invece riempie a sinistra il numero con zeri fino alla dimensione massima definita.

Tipo di Dato	*Intervallo di valori*	*Note*
bit	Da 0 a 255	Introdotto dalla versione 5.0.3 – sinonimo di tinyint[1]
tinyint	Da -128 a 127 signed Da 0 a 255 unsigned	Occupa 1 byte di spazio
smallint	Da -32.768 a 32.767 signed Da 0 65.535 unsigned	Occupa 2 byte di spazio
mediumint	Da -8.388.608 a 8.388.607 signed Da 0 a 16.777.215 unsigned	Occupa 3 byte di spazio
int	Da -2.147.483.648 a 2.147.483.648 Da 0 a 4.294.967.295 unsigned	Occupa 4 byte di spazio
bigint	Da -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 signed da 0 a 18.446.744.073.709.551.615 uns	Occupa 8 byte di spazio
float(M,D)	Virgola mobile singola precisione	M è il numero massimo di cifre da visualizzare, D il numero di cifre decimali
double(M,D)	Virgola mobile doppia precisione	
decimal(M,D)	Numero a virgola fissa	Vedi riquadro tecnica

Tecnica

La definizione (e la gestione) del tipo **Decimal** sono ***cambiate*** in *MySql 5.X*. Nella vecchia versione il valore era memorizzato fisicamente come una stringa, mentre ora si utilizza un formato binario. Questo potrebbe causare problemi con le applicazioni che accedono ai dati, come, ad esempio, OpenOffice.

5.3.3 Campi di Tipo Booleano

Definito come **boolean**, è l'equivalente di ***TinyInt[1]*** (quindi un valore numerico intero). Come tipo di dati è presente dalla versione 4.1.0 di MySql. Un valore uguale a zero viene considerato "*falso*", diverso da zero "*vero*".

5.3.4 Campi di Tipo Data/Ora

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

5.3.5 Campi di Tipo Binario / Text

In questo caso MySql usa una variante del tipo *text* chiamata ***blob*** (binay long object) con alcune tipologie sempre collegate alla dimensione massima archiviabile (***tinyblob***, ***mediumblob***, ***longblob***). Per l'archiviazione di dati di tipo *testo* abbiamo i corrispondenti ***tinytext***, ***text***, ***mediumtext***, ***longtext***. Le dimensioni massime archiviabili sono elencate al capitolo 11.5 della guida di MySQL.

5.3.6 Campi particolari: Intero ad incremento automatico

In MySql possiamo usare un campo ***Int*** con la caratteristica ***auto_increment*** settata.

TIP

Si noti che in MySql esiste anche un tipo ***SERIAL*** equivalente a ***BIGINT*** **UNSIGNED NOT NULL AUTO_INCREMENT**. Siccome in altri motori di Db il tipo *SERIAL* viene utilizzato per le chiavi primarie, se non avete particolari esigenze, per lo stesso scopo può essere usato il tipo ***INT*** **UNSIGNED NOT NULL AUTO_INCREMENT**, che risparmia 4 byte ed è più veloce nelle elaborazioni.

5.3.7 Campi particolari : Timestamp

Il ***Timestamp*** di MySql è di tipo "classico", cioè viene aggiornato automaticamente dal motore di Db. In una Tabella è possibile definire più campi Timestamp: quelli successivi al primo sono a disposizione dell'utente.

5.4 MySql Administrator

Ogni Database che si rispetti avrebbe bisogno di uno strumento di amministrazione semplice e potente, adatto anche a chi non vuole usare la pur potente riga di comando. Dall'*estate del 2004* è disponibile una nuova interfaccia di amministrazione per i server MySql, chiamata appunto **MySql Administrator**, scaricabile, per tutti i sistemi operativi supportati, da *www.mysql.com*. Con questo Tool è possibile gestire tutti gli aspetti della configurazione del nostro Db Server; è inoltre possibile modificare, aggiungere, cancellare Tabelle, Campi ed Indici, il tutto con una semplicità encomiabile. Dopo aver avviato il programma è necessario scegliere quale Server SQL vogliamo gestire, fornendo anche le eventuali credenziali di autenticazione, nella maschera in figura:

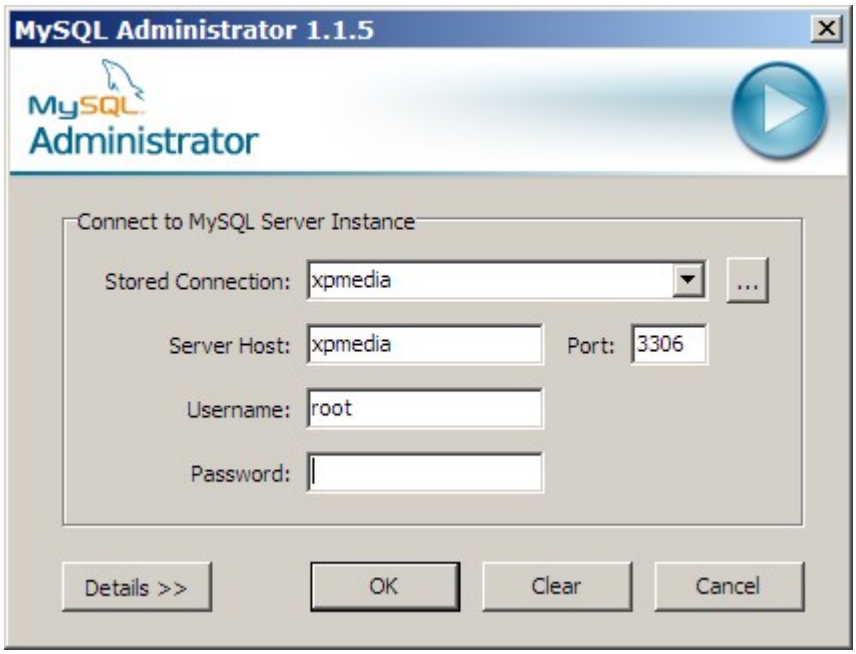


Figura 5.4.1: Connessione a un Db con Administrator

Vi ricordo che se MySql gira sulla stessa macchina che state usando, è sufficiente inserire come Hostname "*localhost*", in caso contrario è necessario indicare l'indirizzo Ip del server. A questo punto vi troverete con la finestra principale finestra attiva.

Non è ovviamente questa la sede per illustrare in dettaglio tutte le funzionalità di *MySql Administrator,* mi limiterò quindi a segnalarvi gli aspetti più utili. Selezionando la voce

desiderata nell'elenco a sinistra, il programma mostra sulla destra i pannelli di gestione della voce stessa.

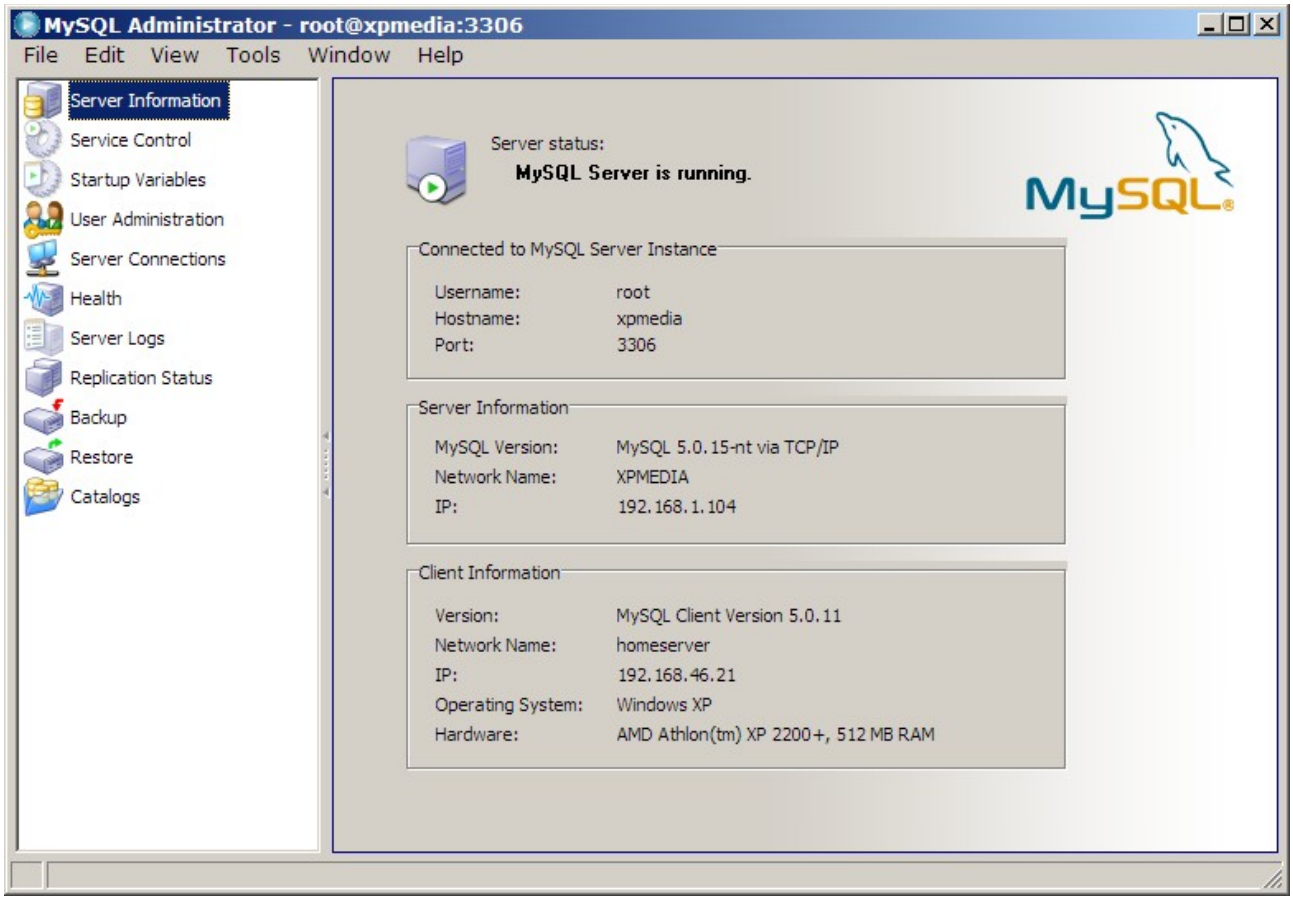


Figura 5.4.2: Finestra principale di MySql Administrator

La voce **Service Control** permette di stabilire le modalità di funzionamento del Server; può infatti essere utile decidere che il servizio sia eseguito in automatico dal Sistema Operativo all'avvio, ed in effetti questo è il caso più comune. Tra le altre opzioni, vi ricordo che *Support for InnoDb* (cioè la possibilità di usare le Tabelle di Tipo InnoDb) è attivo di default dalla Versione 4.0, quindi va eventualmente abilitato solo per le versioni precedenti; che *Support for BDB* serve solo se volete usare quel tipo di Tabelle (vi ricordo che viene ancora dichiarato in Beta Test); che *Named Pipes* permette l'accesso al Db sul Localhost (cioè in locale) attraverso questa modalità operativa al posto del Tcp/Ip (l'uso delle *Named Pipes* è comunque, a mio parere, una opzione inutile, anzi direi dannosa).

Startup Variables permette il *fine tuning* (configurazione dettagliata) dell'ambiente MySql, fornendo un'interfaccia grafica di accesso a tutti i parametri di configurazione specificati nel file *my.ini*. Per quanto una trattazione più approfondita sarebbe interessante, non è questa la sede adatta. Vi consiglio però di fare modifiche solo se ben consapevoli di quello che volete ottenere.

User Administration serve a gestire gli account Utente ed i privilegi di accesso al Server ed ai Database. Di questo ci occuperemo in dettaglio in uno dei prossimi paragrafi.

Server Connections si occupa di monitorare in dettaglio gli accessi di basso livello al server: è utile soprattutto nelle fasi di controllo del carico e delle connessioni attive.

Health propone un quadro molto dettagliato dello "stato di salute" del nostro Server attraverso grafici aggiornati in tempo reale su vari aspetti del funzionamento di MySql. Molto utile ed anche "scenografico".

Server Logs permette l'accesso a tutti i file di Log del Server.

Replication Status fornisce informazioni sulle eventuali "repliche" dei dati presenti nel Server; per gli scopi di questo documento è un aspetto che davvero non ci interessa.

Backup e **Restore** forniscono un strumento semplice per effettuare copie di sicurezza dei nostri Db, quindi ne parlerò in dettaglio in uno dei prossimi paragrafi.

Catalogs permette la gestione completa e semplice della struttura dei nostri Database; in pratica da questa opzione si può modificare un Db in ogni suo aspetto, ed è per questo che ne parleremo in modo esteso nel prossimo paragrafo.

5.4.1 Parametri di avvio del Server

Dopo l'installazione, è possibile stabilire una volta per tutte alcuni parametri di avvio del Server che vengono salvati in Windows nel File *My.ini* nella Dir di MySql.

TIP

I parametri di avvio sono modificabili solo se MySql Administrator viene eseguito sulla macchina Server, e solo se l'utente ha diritti di Amministratore sul Computer (quindi root per Linux). Tra le opzioni della Tab *General Parameters* di *Startup Variables* figura anche un *Disable Networking* che in pratica limita il funzionamento del Server alla macchina locale esclusivamente attraverso le *Named Pipes* (in Windows). A meno che non ci siano stringenti esigenze di sicurezza, non disabilitate ovviamente mai l'accesso tramite Tcp/Ip.

5.4.2 Gestione del Database

Se scegliamo **Catalogs**, ci appare in basso l'elenco degli ***Schemi*** cioè dei *Database* presenti nel Server. Selezionando con un click uno degli schemi (ad esempio *Mediateca*) ci ritroveremo nella parte destra, nella Tab **Schema Tables**, l'elenco delle Tabelle contenute nel Database, insieme ad una serie di utili informazioni sulle stesse (vedi figura). Da questa finestra è possibile inserire, modificare e cancellare i campi e le tabelle del Db. La seconda Tab, **Schema Indices**, contiene un elenco di tutti gli indici assegnati a tutte le tabelle. Le Tab **Views** e **Stored Procedures** sono specifiche della versione 5.X di MySql e saranno esaminate in seguito. Se si seleziona una Tabella nello *Schema Tables*, un click sul pulsante in basso ***Details*** ci fornisce informazioni più approfondite sulla Tabella stessa. Il pulsante ***Maintenance***

consente invece di ottimizzare le Tabelle e di recuperare i dati in caso di problemi. Tra le tre opzioni disponibili per la manutenzione, la più interessante (ma anche quella che richiede più tempo di elaborazione) è senza dubbio ***Optimize*** che in un colpo solo ripara (se necessario) la Tabella, ricostruisce gli indici ed aggiorna le statistiche di accesso. L'operazione si può eseguire anche su più tabelle, selezionando le tabelle stesse con le *SHIFT* (per quelle contigue) o con il *CONTROL* per quelle non contigue.

TIP

Nella Figura compare già il nostro Database di esempio "*mediateca*", ma nel seguito vedremo come costruirlo passo dopo passo. MySql indica col nome di "*Schema*" quello che in altri motori di Db viene indicato come "*Database*"; così la tradizionale "struttura del Database" cioè l'insieme delle definizioni delle Tabelle, dei Campi e degli Indici diventa "*Struttura dello Schema*"

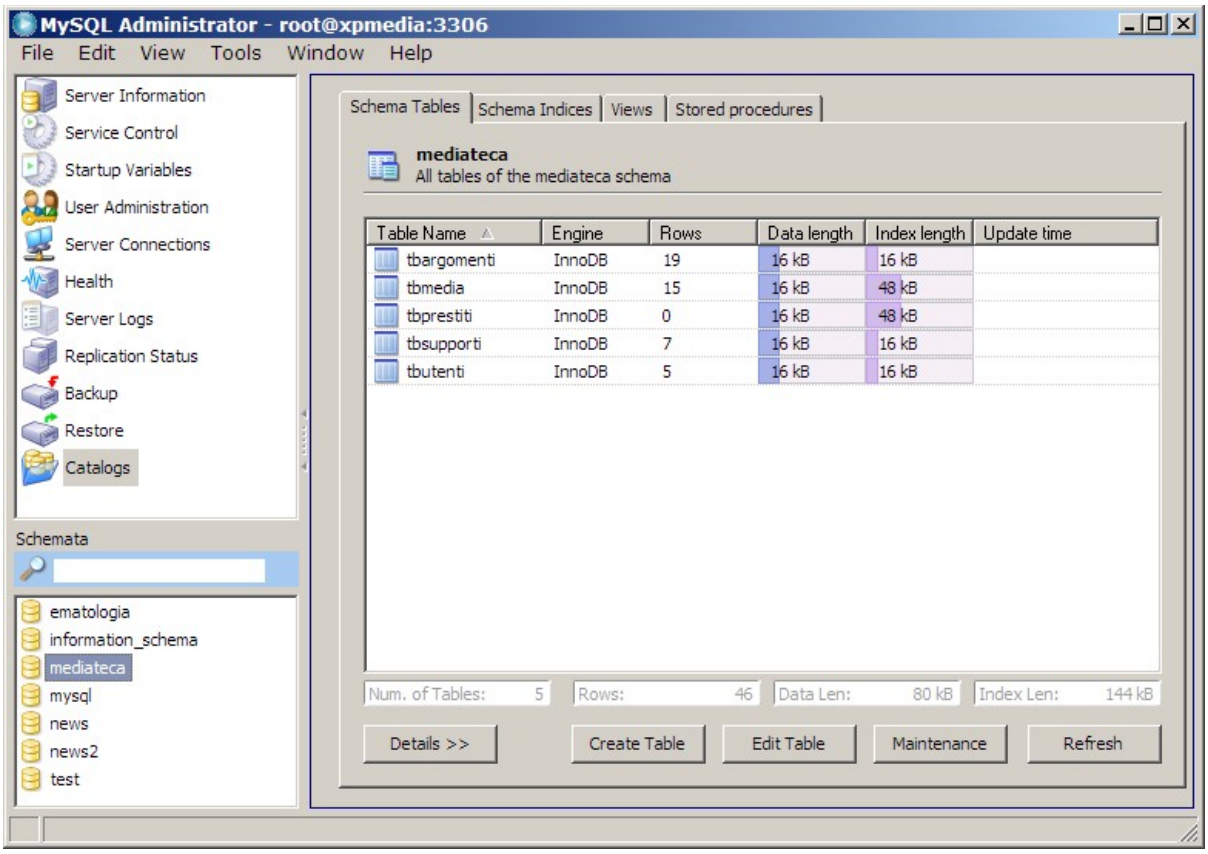


Figura 5.4.3: La gestione dei Catalog

5.5 Gestione delle Tabelle

MySql, a differenza di altri software simili, pur avendo un formato "nativo", può "appoggiarsi" ad altri prodotti Open Source per quanto riguarda il formato di archiviazione delle informazioni. Il formato "originario" del motore si chiama **MyIsam**, ed è in concreto una implementazione piuttosto efficiente del classico *B-Tree*. Nelle versioni più recenti MySql può usare anche Tabelle di tipo **InnoDb** e **BDB** (Berkley Db). In pratica però solo il supporto ad *InnoDb* è dichiarato "stabile", mentre *BDB* è in Beta. Parleremo perciò solo di *MyIsam* e *InnoDb*.

MyIsam, come abbiamo detto, è il formato "nativo" di MySql. Si tratta di Tabelle archiviate in singoli file con estensione ***.MYD*** per i dati e ***.MYI*** per gli indici. Si tratta di una soluzione affidabile e veloce, semplice da gestire anche per quanto riguarda i Backup. Purtroppo questo tipo di tabelle non supporta alcune caratteristiche che sono comuni nei moderni Database, e che possono rivelarsi utili in applicazioni di media complessità. In particolare *MyIsam non prevede la gestione dell'integrità referenziale* (come vedremo tra poco).

Il tipo **InnoDb** è, come abbiamo detto, una "aggiunta" al MySql, nel senso che è un motore di Db di terze parti che è stato inglobato in MySql. Il motore è di concezione più moderna rispetto a *MyIsam*, ed usa, invece dei singoli file, un unico "*tablespace*", più alcuni file di log. Le caratteristiche migliorative di *InnoDb* rispetto a *MyIsam* sono, in breve :

- supporto per le **transazioni**, cioè la possibilità di tornare indietro ad uno stato precedente se l'aggiornamento del Db non va a buon fine;
- **Row Level Locking**, cioè blocco del Record sulla singola riga, utile in ambienti multiutente;
- supporto per le **Foreign Keys**, (cioè le chiavi esterne) che è un primo passo verso l'integrità referenziale;
- **velocità di risposta**, soprattutto in ambiente Windows.

L'uso di un tipo piuttosto che l'altro dipende, quindi, dal tipo di applicazione e di dati che bisogna gestire. Per Db semplici, magari in Linux, va benissimo anche *MyIsam*. Per applicazioni complesse, ed in ambiente Windows, va meglio *InnoDb*. Comunque il passaggio da un tipo all'altro è sempre possibile, e nello stesso Db possono convivere Tabelle di tipi diversi.

Tecnica

Il supporto alle Tabelle di tipo *InnoDb* è incluso nella installazione standard di MySql dalla versione 4.0 in poi. Se non viene utilizzato, è comunque escludibile modificando il file di configurazione con l'opzione *skip-innodb*. Nelle versioni precedenti del motore di Db è necessario, invece, che sia abilitato seguendo le istruzioni dettagliate del manuale dell'applicazione.

5.5.1 Modifica della struttura delle Tabelle

Il primo passo nella gestione di un Db è la creazione dello *Schema*. Bisogna posizionarsi nel pannello dei *Catalogs* in basso a sinistra e quindi, dal menu contestuale del tasto destro del mouse selezionare ***Create New Schema*** e scegliere il nome da dare al "nascituro". Allo stesso modo, per creare una nuova *Tabella*, basta selezionare lo *Schema* e scegliere il pulsante ***Create Table*** in basso.

Allo stesso modo si può modificare la struttura di una Tabella esistente con ***Edit Table***. La finestra che si apre è divisa in tre Tab, nella parte alta. ***Columns and Indices*** si occupa dei campi e degli indici associati; ***Table Options*** permette di stabilire alcuni parametri riguardanti la Tabella; ***Advanced Option*** non è interessante per noi in questo momento.

La Tab ***Columns and Indices*** visualizza nella parte centrale l'elenco e le caratteristiche dei campi: un doppio click sulla parte che ci interessa permette la modifica del parametro.

TIP

Notate che nella colonna *Datatype* non c'è una casella a discesa per la selezione del tipo ma il parametro va specificato per esteso (scrivendo fisicamente ad es. INTEGER). In verità il programma usa una specie di completamento automatico, e chiude anche le parentesi: all'inizio ci si sente disorientati, ma dopo un po' sembra una buona idea.

Nella parte bassa troviamo una Tab (***Indices***) per la gestione degli indici associati alla Tabella. A *sinistra* sono elencati tutti gli indici presenti, ed è possibile aggiungere o cancellare nuovi indici con i pulsanti **"+"** e **"-"** che vedete in basso. Nella parte *centrale* è possibile modificare il nome ed il tipo dell'indice. Nella parte *destra* vengono elencati i campi che fanno parte dell'indice. Per aggiungere un campo è necessario selezionarlo nell'elenco in alto e premere il pulsante **"+"** sull'estrema destra. In alternativa è possibile usare il drag'n'drop, ma a me non sempre riesce.

La Tab ***Foreign Keys*** sarà esaminata in dettaglio tra un attimo.

La Tab ***Column Details*** contiene, in una forma diversa, le stesse informazioni dell'elenco dei campi in alto.

Con la Tab in alto ***Table Options*** è possibile stabilire il *Tipo* di Tabella che desideriamo gestire. Abbiamo parlato nel Paragrafo precedente di ***MyIsam*** e ***InnoDb***: bene qui scegliamo quale *motore* gestirà i dati. Le Opzioni sono molte (ben sette!), ma a noi interessano solo le prime due.

Una caratteristica interessante di *MySql Administrator* è che alla fine delle modifiche, premendo il pulsante ***Apply Changes*** viene mostrato l'elenco dei comandi SQL che dovranno essere eseguiti sul Db. Questo è estremamente "didattico", perché traduce in SQL quello che abbiamo richiesto, permettendoci di capire meglio quello che accade.

5.5.2 Aggiungere le Tabelle a Mediateca

Seguendo le indicazioni sulla struttura dei Dati fornite nel Capitolo "*Il nostro Database di esempio*" non dovrebbe essere difficile creare la struttura di Mediateca. In figura i campi di TbMedia.

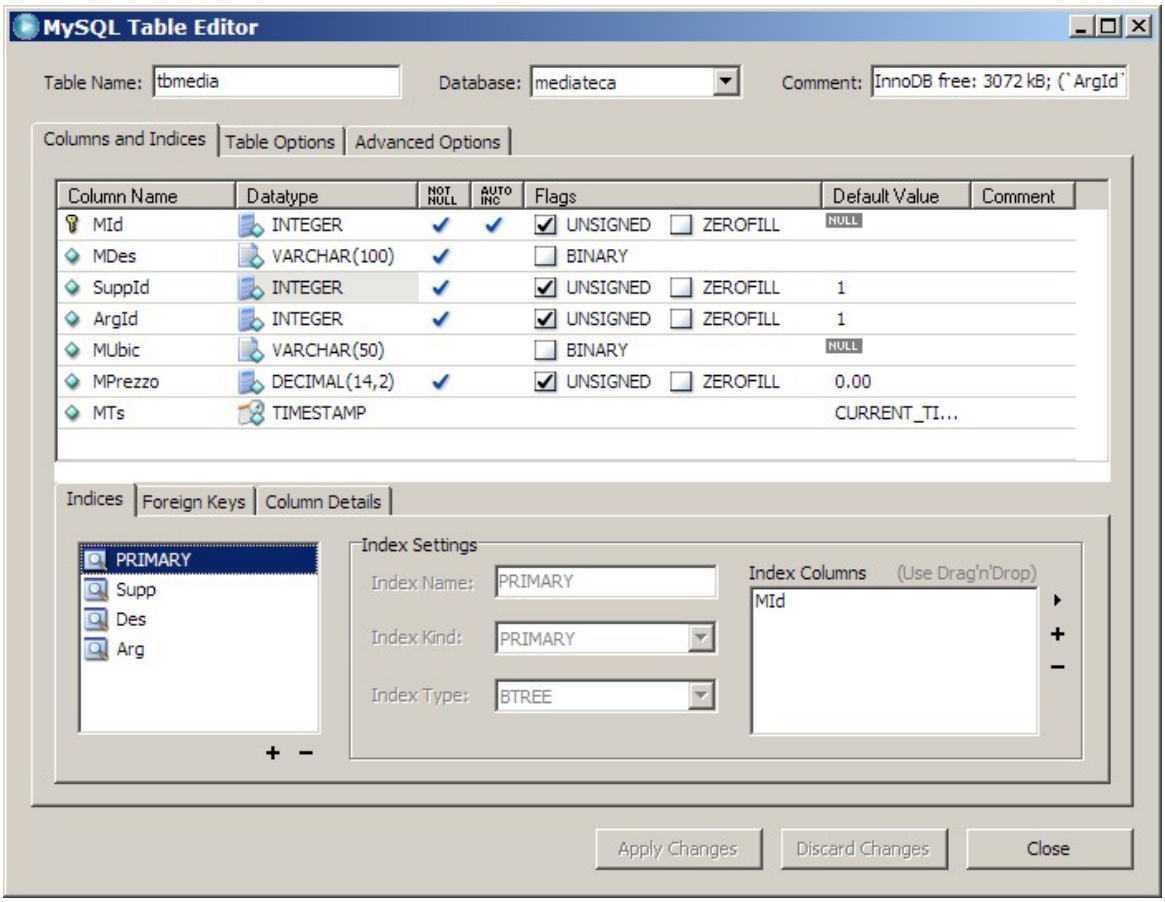


Figura 5.5.1: Modifica della struttura di una Tabella

TIP

La proprietà **valore predefinito** (*default value*) 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. Le ultime versioni di MySql Administrator sono un po' schizzinose sulla definizione dei valori predefiniti: se siete indecisi e non si tratta di un campo numerico assegnate pure *null*.

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".

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+*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.

Notate che Administrator è abbastanza "intelligente" da capire che se chiamo il primo campo di una Tabella *MId* (quindi il nome contiene la stringa "Id", cioè "identifier") ed assegno un valore *Integer*, allora quel campo potrebbe essere una chiave primaria; perciò l'indice viene creato in automatico. Gli altri indici invece vanno creati manualmente, come spiegato nei paragrafi precedenti.

Nella Tab ***Table Option*** scegliamo il tipo di motore che desideriamo usare (per comodità nell'esempio useremo *InnoDb*, visto che ci interessa anche l'integrità referenziale).

5.5.3 Integrità Referenziale

Molti motori di Db hanno a disposizione un meccanismo di controllo che impedisce la cancellazione o la modifica delle chiavi esterne, o comunque permette di stabilire regole precise per la gestione di queste eventualità. Questi motori si occupano perciò di *preservare l'integrità referenziale del DataBase*. Per molto tempo gli sviluppatori di MySql hanno sostenuto che *MyIsam*, per mantenere le caratteristiche di leggerezza e velocità, non doveva occuparsi dell'integrità referenziale, che quindi era lasciata completamente sotto la responsabilità dell'utente. Fortunatamente la cose con *InnoDb* sono un po' cambiate.

Le Tabelle InnoDb supportano la gestione delle chiavi esterne, non tanto nelle query di relazione (cioè quando si usano in una query più tabelle), quanto appunto nel controllo dell'integrità referenziale. Per applicare ad una tabella una chiave esterna, è necessario che :

- le tabelle coinvolte siano tutte di tipo *InnoDb*;
- i campi da mettere in relazione siano dello stesso tipo e della stessa lunghezza;
- il campo nella tabella che *contiene la chiave esterna* (nel nostro caso *TbArgomenti*) deve avere un indice univoco e deve preferibilmente essere la chiave primaria;

- il campo nella tabella che *fa riferimento alla chiave esterna* (*TbMedia*) deve essere indicizzato;

Se tutte le condizioni precedenti sono verificate, possiamo impostare la Foreign Key.

5.5.4 Impostazione dell'Integrità Referenziale

MySql Administrator è, a quanto mi risulta, il primo Tool grafico ufficiale (cioè direttamente supportato da MySql Ab) che permette la gestione delle chiavi esterne. Se i prerequisiti che abbiamo già elencato sono tutti verificati, nella finestra di modifica della struttura della Tabella (nel nostro caso TbMedia), in basso, troviamo una Tab chiamata ***Foreign Keys*:**

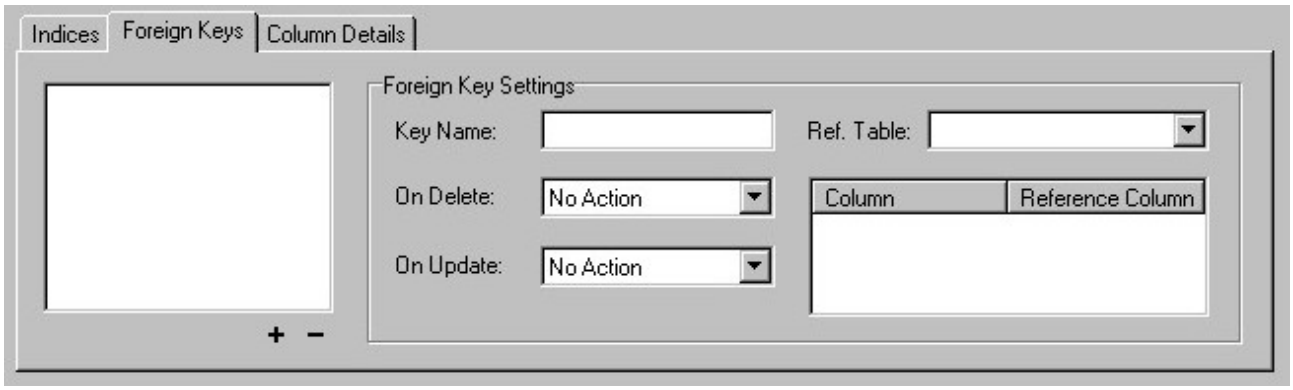


Figura 5.5.2 Finestra di gestione delle Chiavi Esterne

Vi ricordo che stiamo definendo una chiave esterna (*ArgId*) per la Tabella *TbMedia*, che faccia riferimento al Campo *ArgId* della Tabella *TbArgomenti*. Per aggiungere la chiave è necessario fare click sul pulsante **"+"** in basso a sinistra. Dopo aver assegnato un nome a piacere alla chiave, bisogna selezionare dalla casella a discesa ***Ref. Table*** qual'è la tabella che contiene il campo collegato (nel nostro caso *TbArgomenti*). Se esistono campi comuni tra le due tabelle (*ArgId*), il programma li aggiunge direttamente alla relazione (***Column***, ***Reference Column***) come in figura:

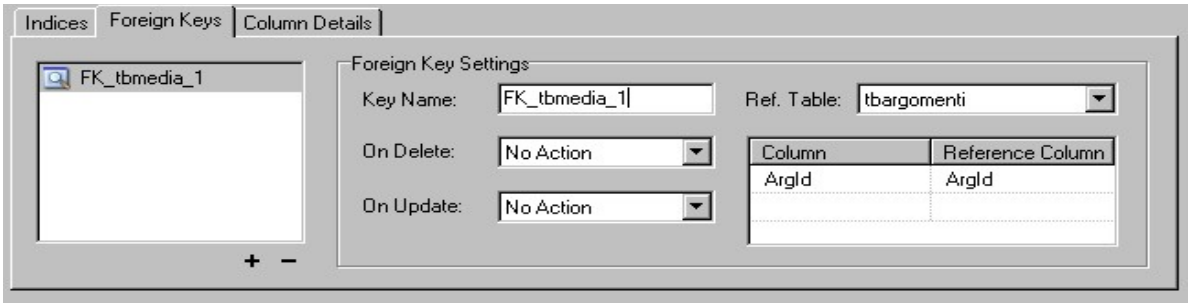


Figura 5.5.3: Chiave esterna degli Argomenti

Con il pulsante ***Apply Changes*** salviamo la chiave appena creata, e, se volete, date anche uno sguardo al comando SQL che *Administrator* si appresta ad eseguire:

```
ALTER TABLE `mediateca`.`tbmedia`
ADD FOREIGN KEY `FK_tbmedia_1` (`ArgId`)
REFERENCES `tbargomenti` (`ArgId`);
```

La sintassi è piuttosto comprensibile, quindi evito altri commenti. Quali risultati abbiamo ottenuto ? Per prima cosa se tentassimo di cancellare il Record con *Id* uguale a 5 (Gialli); otterremo questo messaggio di errore di questo tipo :

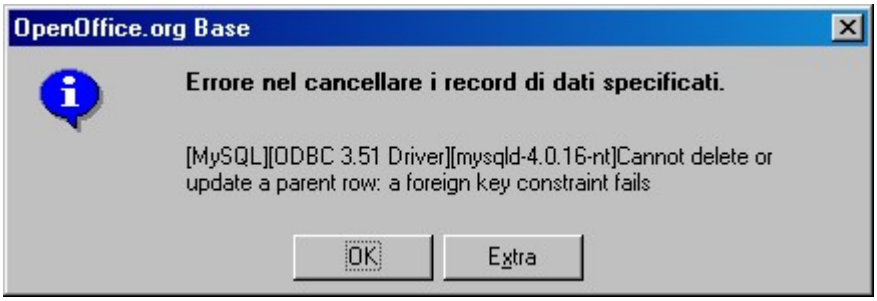


Figura 5.5.4 Messaggio di errore di integrità violata

Cioè il Motore di Db mi impedisce di cancellare una Chiave esterna referenziata in un'altra Tabella (ne nostro caso TbMedia). Significa, in parole povere, che finché in TbMedia esisterà un Record con ArgId uguale a 5, il Record con ArgId uguale a 5 di TbArgomenti NON POTRA' ESSERE CANCELLATO. In altre parole, non posso eliminare un genitore se esistono figli (e questo per chi crede che l'informatica manchi di etica...). Questo è già un buon risultato, ma andiamo avanti.

Se invece cercassimo di aggiungere a *TbMedia* un Record con un *ArgId* non presente nella Tabella *TbArgomenti* (ad esempio un valore 99, inesistente), il messaggio di errore sarebbe:

Figura 5.5.5 Errore di violazione dell'integrità referenziale

Cioè il Motore di Db mi impedisce di aggiungere un Record con un valore nel campo ArgId NON PRESENTE in TbArgomenti. Cioè non è possibile inserire figli senza genitore. Era quello che volevamo, no?

Torniamo però un passo indietro. Nella finestra di impostazione della chiave esterna, si possono notare due altre opzioni, precisamente ***ON UPDATE*** e ***ON DELETE***, entrambe settate su ***NO ACTION***. Bene, se quello che abbiamo appena visto è il comportamento standard del motore di Db (*NO ACTION*), possiamo comunque eventualmente scegliere delle opzioni alternative ne caso si modifichi (*ON UPDATE*) o si cancelli (*ON DELETE*) la chiave esterna (cioè il genitore di una serie di record presenti nella Tabella "figlia"). In particolare :

CASCADE estende la variazione della chiave esterna alla Tabella "figlia"; nel caso di ***ON UPDATE CASCADE***, se, ad esempio, in *TbArgomenti* modifichiamo *ArgId* da 5 a 55, tutti i record di *TbMedia* con *ArgId* uguale a 5 saranno aggiornati di conseguenza. Nel caso di ***ON DELETE CASCADE***, però, la cancellazione di una chiave esterna NON VIENE PIU' IMPEDITA, e vengono eliminati a cascata anche tutti i record della Tabella "figlia". *Cioè se cancello "5" in TbArgomenti, non avrò più nessun Giallo in TbMedia.*

SET NULL invece si limita ad impostare a Null tutte le chiavi figlie del genitore modificato.

RESTRICT infine, è equivalente a *NO ACTION*, cioè controlla l'applicazione dell'integrità.

Come vedete l'integrità referenziale è uno strumento molto potente, perché permette di stabilire regole a livello di motore di Db che non possono essere scavalcate da manovre errate dell'utente. Queste regole però vanno impostate con cognizione di causa, e, soprattutto, in fase di progettazione *iniziale* del Database (o, adottando la nomenclatura di MySql, della struttura del Catalogo). Applicare una nuova regola di integrità ad una Tabella già colma di dati infatti non sempre è possibile, perché alcuni dei record immessi potrebbero già non rispettare la regola stessa.

5.6 Backup degli Archivi MySql

Una sana politica di Backup, come ben sapete, ci mette al riparo da disastri che provocano giornate di lavoro perse e mal di testa cronici. Per quanto la natura umana sia pervasa dalla convinzione che "a me non succederà mai", la logica e la Legge di Murphy ci dicono che invece sicuramente prima o poi accadrà. Quindi, di fronte all'ineluttabile, meglio essere preparati (stiamo parlando di informatica, forse è meglio non generalizzare...).

La domanda da porsi, quindi, è : *come si fa un Backup degli Archivi di MySql ?*

Fortunatamente la risposta è abbastanza semplice: *basta trasferire periodicamente i file dei Dati su un altro supporto*. Già, ma *quali file ?*

Per le Tabelle ***MyIsam***, per ogni "catalogo" (cioè Database) esiste una cartella nella dir dei Dati definita per il Server. In Windows basta cercare in c:\programmi\mysql\MYSQL Server X.X\data\. Un Backup dell'intero percorso mette al sicuro tutti i cataloghi. In Linux, a seconda

dei parametri di installazione, esiste una struttura equivalente di solito in /usr/local/mysql. *Ovviamente questo metodo funziona quando il Server non è in esecuzione.*

Se utilizziamo anche Tabelle ***InnoDb***, a quanto detto prima vanno aggiunti i file di Dati e di Log di InnoDb, e sono tutti quelli che iniziano per "ib" (ib*).

Se non è possibile disattivare il Server, il manuale di MySql elenca con dovizia di particolari tutta una serie di metodi alternativi, alcuni basati su tool da linea di comando, per ottenere lo stesso risultato senza grossi sforzi. Non credo però che sia il caso di approfondire, anche perché abbiamo, volendo, il supporto di una ottima interfaccia grafica....

5.6.1 Backup dei Dati

Alla voce ***Backup*** di MySql Administrator, basta selezionare ***New Project*** col pulsante in basso e fornire i parametri del nostro Backup. In particolare, è possibile selezionare più di un Catalogo, oppure, all'interno di un Catalogo, solo alcune Tabelle. Il pulsante ***Save Project*** permette il salvataggio dei parametri di Backup, in modo da non doverli reinserire in seguito.

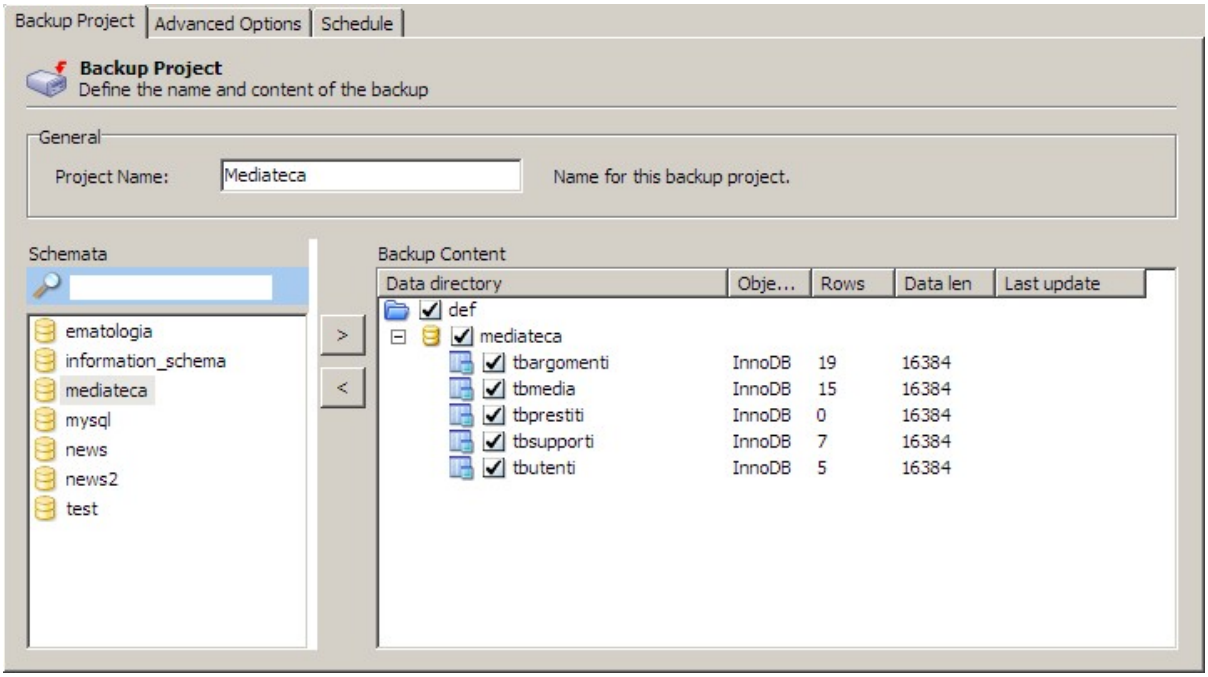


Figura 5.6.1: Backup con MySql Administrator

La Tab ***Advanced Options*** serve a definire alcune opzioni utili. Se le tabelle sono di tipo **MyIsam**, selezionate, nel ***Backup Execution Method*** la voce ***Lock All Tables***, che "blocca" le Tabelle per permettere un Backup sicuro dei dati. Se invece avete usato **InnoDb**, selezionate l'opzione ***InnoDb OnLine Backup***, più adatta allo scopo.

In ***Output File Options*** c'è poco da selezionare, e direi che è saggio lasciare le opzioni di default, anche se le scelte sono abbastanza intuitive. Il ***Backup Type*** permesso, almeno per questa versione, è unicamente ***Sql file***; questo significa che tutto il Db (o le tabelle che avete selezionato) sarà salvato come un file di testo con estensione *.sql* contenente i comandi SQL

necessari a ricreare il Db da zero. Non si tratta perciò di una copia "binaria" dei dati, ma della "trasposizione" SQL dell'intera struttura e del contenuto del Catalogo.

Non ci credete ? Allora date un'occhiata all'inizio del File creato dal Backup della Tabella tbArgomenti di Mediateca :

```
[.....]
--
-- Create schema mediateca
--

CREATE DATABASE /*!32312 IF NOT EXISTS*/ mediateca;
USE mediateca;

--
-- Table structure for table `mediateca`.`tbargomenti`
--

DROP TABLE IF EXISTS `tbargomenti`;
CREATE TABLE `tbargomenti` (
  `ArgId` int(10) unsigned NOT NULL auto_increment,
  `ArgDes` varchar(30) NOT NULL default '',
    `ArgTs` timestamp NOT NULL default CURRENT_TIMESTAMP on update
CURRENT_TIMESTAMP,
  PRIMARY KEY  (`ArgId`),
  KEY `Arg` (`ArgDes`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `mediateca`.`tbargomenti`
--

/*!40000 ALTER TABLE `tbargomenti` DISABLE KEYS */;
INSERT INTO `tbargomenti` (`ArgId`,`ArgDes`,`ArgTs`) VALUES
 (1,'Rock','2004-04-08 12:00:19'),
 (2,'Classica','2004-04-08 12:00:22'),
 (3,'Pop','2004-04-11 11:03:22'),
 (4,'Filosofia','2004-04-08 12:00:32'),
 (5,'Giallo','2005-04-26 21:33:31'),
 (6,'Saggio','2004-09-16 19:02:14'),
 (7,'Sistemi Operativi','2004-04-08 12:01:02'),
 (8,'Samba','2004-04-08 12:01:05'),
 (9,'Reti Locali','2005-05-05 20:16:53'),
 (10,'Giochi','2004-04-10 17:34:09'),
 (15,'Fantasy','2004-04-12 18:35:26'),
 (16,'Linux','2004-04-13 18:41:30'),
 (17,'Romanzo','2004-04-16 09:35:02'),
 (18,'Avventura','2004-04-16 09:38:18'),
 (19,'Windows','2005-04-29 11:42:59'),
 (20,'Ftp','2005-04-07 12:23:05');
/*!40000 ALTER TABLE `tbargomenti` ENABLE KEYS */;
[....]
```

Questo metodo ha dei vantaggi. Siccome abbiamo un file di testo con comandi SQL (quasi) standard, potrebbe essere usato, ad esempio, per il trasferimento dei dati ad altri motori di Db. Inoltre questo tipo di Backup è l'ideale per il passaggio di dati tra server MySql diversi, o tra Cataloghi diversi nello stesso Server. Infatti...

5.6.2 Restore dei Dati

La Tab ***Restore*** di MySql Administrator è un esempio di semplicità. Per prima cosa, col pulsante ***Open Backup File*** si seleziona il Backup da recuperare, quindi si può scegliere, in ***Target Schema*** se si desidera recuperare i dati nella posizione originale oppure in un altro catalogo. Se il Db non esiste, si può richiedere di crearlo da zero. Infine nella Tab ***Restore Content*** il pulsante ***Analyze Backup*** ci permette di selezionare solo alcune tabelle per il restore. Un lavoro davvero ottimo.

5.7 La sicurezza : Utenti e Diritti

MySql è un Db particolare e forse la cosa sta lentamente insinuandosi nelle vostre sveglie menti, dopo aver letto fin qui. "Particolare" significa che su alcuni aspetti questo software ha un approccio direi "nordico", vista l'origine. Il "nordico" va inteso nel senso che, per certi versi, gli sviluppatori hanno una visione delle cose assai personale e non vogliono cambiarla. Quelli di voi che hanno usato altri Db ed hanno a cuore la sicurezza dei propri dati, continuando nella lettura di questo paragrafo potrebbero storcere il naso (io direi meglio "sentire un sudore freddo su per la spina dorsale"), ma che volete, questi di MySql passano metà dell'anno al buio....

Allora, tanto per cominciare, ecco tre succose informazioni che servono a farsi un'idea :

1. ogni utente del Db è identificato dal **nome**, **dall'host** da cui si connette e dalla **password**; questo vuol dire che il nostro amico Alfred, che si collega sia dalla macchina che ospita il Server, sia dal Client in Rete deve avere DUE ACCOUNT;
2. il concetto di **Gruppi di Utenti** è ***sconosciuto*** a MySql, per cui non si possono assegnare diritti ad un Gruppo e quindi semplicemente aggiungere un Utente al Gruppo; sarebbe troppo comodo, e quindi dovrete assegnare i diritti singolarmente AD OGNI UTENTE;
3. appena dopo aver installato il Server, l'utente **root** (cioè l'amministratore) ha ***password vuota***; come se non bastasse, potrebbe esistere anche un account che permette l'accesso anonimo, dalla macchina che ospita il Server (localhost), con i poteri di Amministratore (in Windows) e di Ospite (in Unix / Linux), sempre con la password vuota (questo è un po' cambiato nelle ultime versioni: l'installer Windows permette di configurare alcuni importanti aspetti della sicurezza prima di avviare effettivamente il Server)

Bene, se finora pensavate che forse non era una cattiva idea migrare in MySql quel Db con una cinquantina di utenti che avete in Azienda, ora forse qualche dubbio in verità vi assale...

Ma non disperate: mai fermarsi alla prima impressione, e poi di tutto bisogna accettare sia i pregi che i difetti. In verità questa parte relativa alla sicurezza mi sembra un po' trascurata in MySql, e quindi se vogliamo mettere su un Server sicuro c'è un po' da lavorare. Cominciamo quindi con i concetti di base.

Il modello di sicurezza usato da MySql è, in realtà, assai semplice. I livelli di autenticazione (e quindi di assegnazione dei diritti) sono due; il primo permette l'accesso al Server e concede i **privilegi globali**, il secondo si occupa di gestire i diritti sui **singoli Database** (o Schema) presenti nel Server stesso. Inoltre si può scendere fino al dettaglio della singola colonna (o campo) di una Tabella (ad esempio stabilendo che per un utente quel campo deve essere in sola lettura). Allora, vediamo quali sono i diritti assegnabili :

Diritto	**Contesto di assegnazione**
ALTER	tabelle
DELETE	tabelle
INDEX	tabelle
INSERT	tabelle
SELECT	tabelle
UPDATE	tabelle
CREATE	cataloghi, tabelle, o indici
DROP	cataloghi o tabelle
GRANT	cataloghi o tabelle
REFERENCES	NON USATO
CREATE TEMPORARY TABLES	amministrazione del server
EXECUTE	NON USATO
FILE	accesso ai file
LOCK TABLES	amministrazione del server
PROCESS	amministrazione del server
RELOAD	amministrazione del server
REPLICATION CLIENT	amministrazione del server
REPLICATION SLAVE	amministrazione del server
SHOW DATABASES	amministrazione del server
SHUTDOWN	amministrazione del server
SUPER	amministrazione del server

Come vedete, in generale il nome è identico alla corrispondente istruzione SQL, quindi il significato è lampante. Il privilegio di GRANT permette di trasferire i propri diritti ad un altro Utente. *Un **privilegio globale** vale per tutti gli Schemi (Database) gestiti dal Server, e per il Server stesso.*

Abbiamo già detto che ogni **utente** viene identificato da tre parametri: il ***nome***, ***l'host*** da cui si connette e la ***password***. Per "host da cui si connette" intendiamo il **nome** o **l'indirizzo IP** del Computer che richiede la connessione. La macchina su cui gira il Server è identificata sempre dalla stringa "*localhost*". Quindi l'utente pippo@testws è diverso da pippo@multimedia. *Diverso* significa che può avere password diversa e privilegi differenti. Per fortuna possiamo

almeno usare delle wildcards (ma solo nel campo host), quindi ad esempio ***pippo@%*** significa "*l'utente Pippo, da qualsiasi workstation si connetta*". Se si indica solo il nome di host, invece, per MySql significa "*accesso anonimo dall'host*", meglio ancora "*tutti gli utenti dell'host*". Però questo implica che, ad esempio, ***@localhost*** significa "accesso anonimo dal localhost", ***@testws*** "accesso anonimo da testws", ***@%*** "accesso anonimo da qualunque PC collegato in rete" (con tanti ringraziamenti da parte di un eventuale intruso).

Ora, se vi sentite confusi, sappiate di essere in buona compagnia; ma non disperate, non è ancora finita. Supponiamo di voler affidare all'utente *pippo* la gestione del Database *Mediateca*; vogliamo inoltra che pippo possa collegarsi da tutti gli host della nostra rete. Quindi aggiungiamo l'utente *pippo@%* e concediamo tutti i diritti su *Mediateca*. Pippo si collega senza problemi e svolge il suo lavoro. Per caso abbiamo inoltre, sulla nostra rete, un PC che si chiama *testws*, e desideriamo che tutti gli utenti di questo host accedano al Database *news2*; quindi creiamo l'utente anonimo *@testws* ed assegniamo i diritti sul catalogo *news2*. La situazione attuale sarebbe, quindi :

Utente	Significato	database	diritti
pippo@%	pippo, collegato da qualsiasi host	mediateca	gestione
@testws	qualsiasi utente anonimo dell'host testws	news2	gestione

A questo punto *pippo* si siede al Pc *testws* e chiede l'accesso a *mediateca*: MySql rifiuta la connessione. Questo vuol dire che *i diritti di accesso anonimo dell'host invalidano quelli del singolo utente*. Perciò, **O** consentiamo l'accesso anonimo da *testws* anche a *mediateca*, ma questo apre il db anche agli altri utenti, **OPPURE** aggiungiamo un utente pippo@testws con i diritti appropriati.

A questo punto *pippo*, come utente di *testws*, vuole accede a *news2*, quindi fornisce le sue credenziali (nome e password) e MySql gli *rifiuta la connessione*. Cioè *per accedere a news2 da testws l'utente deve essere anonimo*. In caso contrario MySql controlla i diritti di pippo, e se non è specificato che può accedere a news2, nega la connessione.

Tutto questo ha una logica (anche se per me piuttosto oscura) ed è abbastanza ben spiegato ai capitoli 5.4 e 5.5 del manuale di MySql. Quindi coraggio e buona lettura. Vi risparmio anche i dettagli di come tutte queste informazioni siano archiviate nel catalogo (Database) di nome MySql (non sono un sadico, e voi interrompereste la lettura dopo due righe). Siccome però è probabile che (come me) avete capito poco, qui di seguito vi passo qualche semplice consiglio che viene fuori dalla mia esperienza di utente. Allora:

1. dopo l'installazione di MySql, per prima cosa assegnate una *password* all'utente **root**; ricordate che gli utenti root sono DUE, uno per la connessione da localhost, l'altro per la

connessione dalla rete; se desiderate che root possa collegarsi SOLO dal localhost, eliminate l'utente **root@%;**

2. eliminate o regolamentate l'***accesso anonimo*** al server; una buona soluzione è cancellare del tutto gli utenti **@%** e **@localhost;**
3. se non serve controllare l'host di collegamento, aggiungete solo utenti del tipo **utente@%**; questo garantisce comunque la verifica dell'autenticazione senza specificare da quale workstation la connessione debba avvenire;
4. se invece volete controllare l'host di collegamento, tenete presente che il nome host viene passato direttamente dal sistema (cioè voi non potete specificarlo);
5. il modello di sicurezza di MySql non si basa su quello del Sistema Operativo su cui è in esecuzione, quindi aggiungere troppi utenti complica la gestione; piuttosto, se sul server sono presenti più database, può essere utile seguire un modello "orientato al catalogo". Ad esempio, per l'accesso a mediateca, posso aggiungere un utente **mediateca@%,** con i soli privilegi su questo db;
6. ricordate che di solito l'accesso a MySql avviene attraverso la configurazione di un **DSN** (Data Source Name) di tipo ODBC sia in Linux che in Windows; i DSN possono essere specifici di un utente o di una macchina; potrei voler assegnare un DSN già configurato all'utente od alla macchina, senza rivelare all'utente la password di accesso a MySql; quindi posso aggiungere un DSN con nome utente "mediateca", uno con nome utente "news2" etc.; in questo modo all'utente non verrà MAI chiesta una password di connessione; questa soluzione però non funziona in Linux, perché nel DSN non si può specificare un nome utente ed una password;
7. limitate l'assegnazione di privilegi globali solo agli account che ne hanno effettivamente bisogno; limitate l'accesso di ogni utente ad un singolo Database, salvo casi eccezionali;
8. le password di MySql sono lunghe fino a sedici caratteri, ma non è possibile impostare alcuna policy (numero minimo di caratteri, scadenza etc.) su di esse; quindi è fondamentale la scelta di password non banali.

Bene, ora che abbiamo studiato la teoria, passiamo alla pratica.

TIP

Se, nei parametri del Server, avete disabilitato la gestione dei nomi allora tutti i permessi devono far riferimento all'indirizzo fisico della macchina. Non è perciò consentito (o almeno non funziona) un utente del tipo fc@homeserver. Questo, se ad esempio usiamo un server DHCP per l'assegnazione automatica degli indirizzi, è molto scomodo perché in teoria la stessa macchina potrebbe nel tempo assumere indirizzi IP diversi.

5.7.1 Gestione della sicurezza da linea di comando

Tutta la parte di gestione di un Server MySql (e quindi anche la sicurezza) può essere eseguita tramite tool a linea di comando presenti nella cartella **..\mysql\bin**, a cominciare dal già citato **mysql**. Credo non sia questa l'occasione per approfondire l'argomento. Sappiate almeno però che a livello SQL l'assegnazione e la revoca di diritti avviene tramite gli statements *GRANT* e *REVOKE*. A titolo di esempio ecco la sintassi di *GRANT* presa dal manuale di MySql:

```
GRANT priv_type [(column_list)] [, priv_type [(column_list)]] ...
    ON {tbl_name | * | *.* | db_name.*}
    TO user [IDENTIFIED BY [PASSWORD] 'password']
        [, user [IDENTIFIED BY [PASSWORD] 'password']] ...
    [REQUIRE
        NONE |
        [{SSL| X509}]
        [CIPHER cipher [AND]]
        [ISSUER issuer [AND]]
        [SUBJECT subject]]
    [WITH [GRANT OPTION | MAX_QUERIES_PER_HOUR count |
                          MAX_UPDATES_PER_HOUR count |
                          MAX_CONNECTIONS_PER_HOUR count]]
```

Inoltre, poiché tutta la struttura è memorizzata nelle tabelle user, db e host del catalogo mysql, è possibile usare anche semplici istruzioni INSERT, DELETE od UPDATE, purché si sappia quello che si sta facendo. Ma a noi va bene anche la GUI di MySQL Administrator...

5.7.2 MySql Administrator : la sicurezza

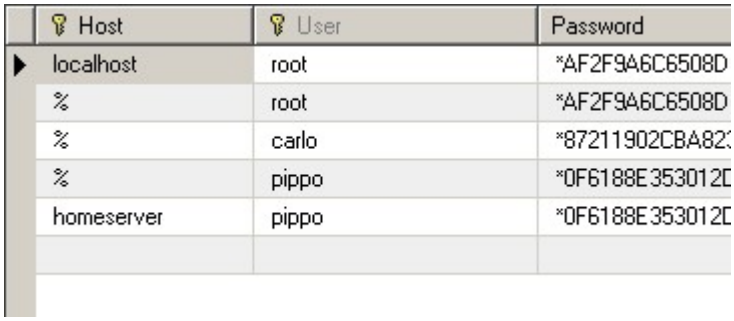
Selezionando la voce User Administration, nella finestra in basso a destra vengono elencati gli utenti autorizzati per il server, come in figura:

Figura 5.7.1: Gestione utenti

Ogni utente è identificato da un nome, e, se necessario, anche dagli hosts specificati per il suo accesso. *Se non viene indicato alcun host, si intende "ogni host"*. Nella sintassi di mysql, quindi il "carlo" della figura sarebbe "carlo@%". Se sono elencati altri host, ognuno rappresenta un utente diverso. Quindi nel nostro caso avremo ***pippo@%***, ***pippo@homeserver***.

Con un menu contestuale attivato dal tasto destro in questa finestra è possibile *aggiungere* un *nuovo utente*. Sempre con un menu contestuale, selezionato un utente si possono eventualmente *aggiungere nuovi host di connessione*. Questa rappresentazione può sembrare

lineare, ma secondo me non lo è affatto. Infatti un nuovo utente viene aggiunto sempre nella forma utente@%, cioè è possibile il collegamento da qualsiasi host. Anche se si aggiunge un host specifico (come per pippo@homeserver), non viene eliminato pippo@%. Questa situazione si vede bene se si apre la tabella ***user*** del Database di sistema ***mysql,*** come in figura:



Host	User	Password
localhost	root	*AF2F9A6C6508D
%	root	*AF2F9A6C6508D
%	carlo	*87211902CBA82
%	pippo	*0F6188E353012D
homeserver	pippo	*0F6188E353012D

Figura 5.7.2: Utenti di un Server MySql

qui si comprende assai meglio che in effetti ***pippo@%*** e ***pippo@homeserver*** sono proprio DUE UTENTI DIVERSI. Per fare in modo che *pippo* si colleghi SOLO da *homeserver* è necessario cancellare manualmente l'utente *pippo@%;* il problema è che, però, in questo caso l'utente viene visualizzato in Administrator solo come pippo (direi che questo modulo ha ancora bisogno di una limatura da parte dei programmatori...). In ogni caso, all'utente è possibile assegnare i privilegi con un comodo sistema mostrato in figura. Abbiamo le tab in alto riguardanti i privilegi globali, i Database (schema), le Tabelle e le colonne.

TIP

In realtà, nella installazione standard di MySql Administrator, nelle Tab compare solo la voce "*schema privileges*"; per abilitare le altre due, nella voce di menu Tools->*Options* è necessario abilitare le caselle di spunta alla voce "*User Administration*".

La gestione è abbastanza semplice; una volta selezionati l'utente e la sezione che ci interessa, tutto si risolve spostando i privilegi elencati a video dalla colonna ***available*** (*disponibile*) a quella ***assigned*** (*assegnato*), come in figura.

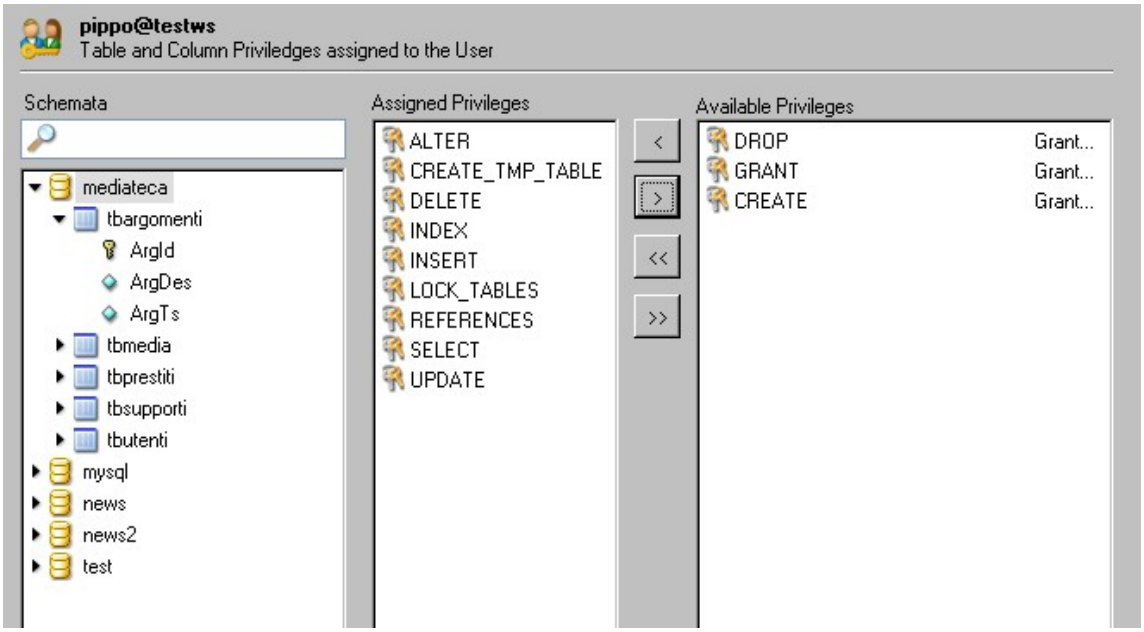


Figura 5.7.3 Assegnazione diritti ad un Utente

5.8 Importazione dei Dati da altri Db

Quando si decide di migrare da un Server di Database ad un altro è sempre abbastanza complicato spostare i dati da una parte all'altra. Nel caso di MySQL è possibile usare più di un sistema (compreso il "transito" attraverso il modulo Base di OOo), ma quello più semplice è l'uso di Migration Toolkit.

5.8.1 Migration Toolkit

Il **Migration Toolkit** è scaricabile da sito di MySql, ed è un Wizard che permette l'importazione di dati da ***Ms Access***, ***Ms Sql Server***, ***Oracle***, un driver ***JBDC*** generico oppure un'altra istanza di MySQL. Il funzionamento è piuttosto intuitivo, quindi vi risparmio i dettagli.

Con *Ms Access* funziona egregiamente: i tipi di dati vengono conservati e, quando non disponibili in MySQL, "tradotti" con intelligenza (ad es. *decimal(19,4)* per il tipo *valuta* oppure *tinyint(1)* per *Si/No*). Si può trasferire un intero Database oppure singole Tabelle; in definitiva si tratta di un ottimo strumento.

5.9 Novità della Versione 5

La Versione 5 di MySQL ha introdotto caratteristiche lungamente attese dalla comunità degli Utenti. In breve, riportando solo quelle essenziali, abbiamo:

- un nuovo tipo di dato **BIT** (equivale a *tinyint(1)*)

- la presenza dell'**Information Schema**, una sorta di catalogo generale di tutti gli oggetti del Database
- incremento della precisione matematica dei calcoli
- due nuovi motori di archiviazione: ARCHIVE e FEDERATED
- gestione di **Stored Procedure** e **Stored Function**
- maggiore conformità allo standard SQL
- supporto (ancora limitato) ai **trigger**
- aumento della capacità di archiviazione del tipo VARCHAR fino a 65.532 caratteri
- supporto per le Viste (**Views**)
- aumento generalizzato della velocità di elaborazione

I progressi sono notevoli, e permetteranno sicuramente a MySQL di acquisire nuovi utenti a scapito della concorrenza.

5.9.1 Creazione di Viste

La creazione di ***viste*** in MySQL 5 è abbastanza semplice: si tratta di utilizzare il comando SQL **CREATE VIEW**. Ad esempio, per la nostra mediateca, il comando potrebbe essere:

```
CREATE VIEW 'mediateca`.`media` AS
SELECT
    `tbmedia`.`MId` AS `Mid`,
    `tbmedia`.`MDes` AS `Mdes`,
    `tbsupporti`.`SuppDes` AS `SuppDes`,
    `tbargomenti`.`ArgDes` AS `ArgDes`
FROM
    ((`tbsupporti` join `tbmedia`) join `tbargomenti`)
WHERE
    ((`tbsupporti`.`SuppId` = `tbmedia`.`SuppId`)
    and (`tbargomenti`.`ArgId` = `tbmedia`.`ArgId`))
ORDER BY `tbmedia`.`MDes`;
```

La sintassi è molto simile alla **SELECT** normale, bisogna solo specificare il nome della vista nella forma ***database.nome***. Una volta creata, la vista può ovviamente essere modificata o cancellata con appositi comandi SQL. In teoria le viste dovrebbero poter essere gestite anche con l'interfaccia grafica di MySQL Administrator, ma la versione attuale (la 1.1.5) non è affidabile sotto questo aspetto.

5.10 Esecuzione di comandi SQL

Potremmo ora domandarci come fare ad "ordinare" al Server l'esecuzione di un comando SQL. I metodi sono sostanzialmente due:

1. usare l'utility a linea di comando **mysql** inclusa nel server;
2. trasferire al server il comando attraverso un Tool esterno (magari ad interfaccia grafica);

Se scegliamo la prima soluzione, dobbiamo lanciare il programma **mysql** contenuto nella dir **..\bin** dell'installazione del Server (in Windows di solito ***c:\mysql\bin***). Si tratta di una shell a linea di comando, quindi ci troveremo più o meno nella situazione in figura:

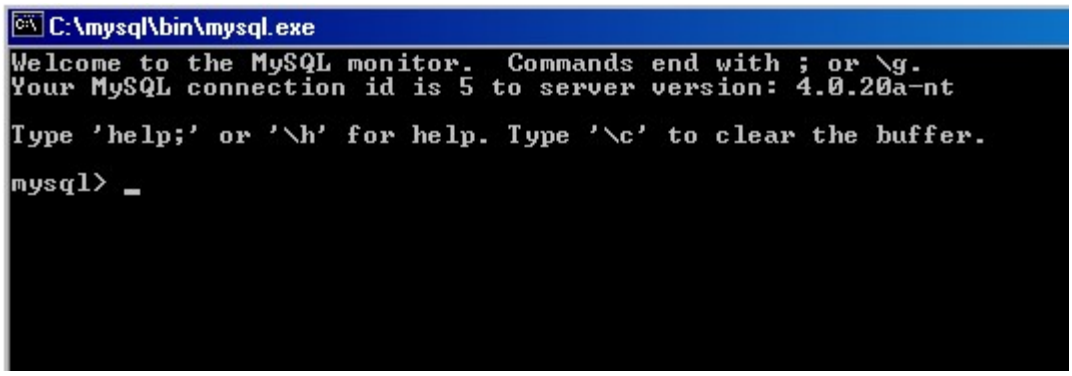


Figura 5.10.1 Il Client MySql

Per noi che abbiamo usato il DBase II la cosa è abbastanza familiare; per voi abituati alle interfacce grafiche potrebbe essere disorientante. Comunque non spaventatevi. La prima cosa da fare è selezionare il Db da usare quindi :

```
use mediateca
```

A questo punto possiamo scrivere qualunque comando SQL valido, con l'accortezza di terminare l'istruzione con un punto e virgola. Ad esempio :

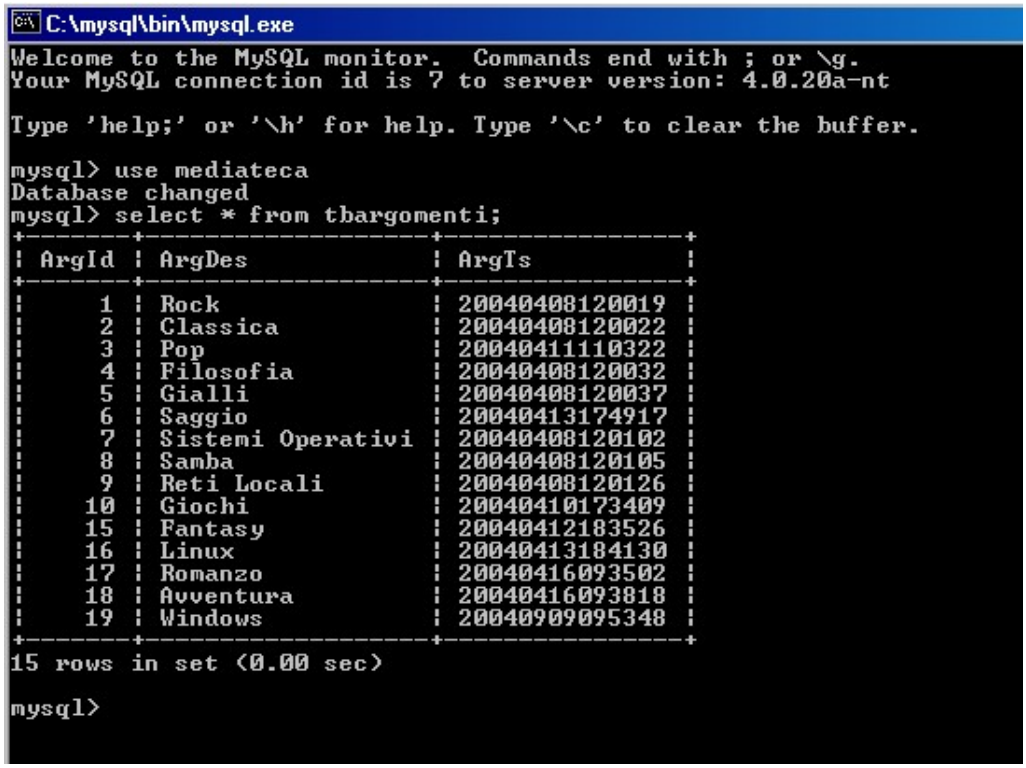


Figura 5.10.2 Il Comando Select

Se premiamo INVIO senza chiudere con il punto e virgola, è possibile continuare la scrittura sulla riga successiva, il che è comodo perché molti statements SQL sono lunghi e complessi. Perciò :

Figura 5.10.3 Ancora un Comando Select

Con questo semplice tool possiamo passare al Server *qualsiasi comando SQL*. Per uscire, un semplice **EXIT** chiude le operazioni. Con un po' di pratica ***MYSQL Monitor*** (nome ufficiale del tool) si rivela abbastanza immediato e semplice da usare. In più, può accettare in input anche un file di testo comprendente più istruzioni SQL, quindi si rivela utile in molti casi. Esistono però varie alternative, come adesso vedremo.

5.11 Altri Tools Grafici per la gestione di MySql

MySql Administrator, come abbiamo visto, è il nuovo strumento di amministrazione ufficiale per MySql. Sul sito www.mysql.com è comunque disponibile un altro Tool, e precisamente il **Query Browser**. Si tratta di un programma molto interessante orientato alla costruzione grafica di query SQL, non tanto nella maniera utilizzata in modalità Disegno di OOo, quanto nel supporto alla sintassi di TUTTI i comandi SQL (e quindi non solo delle SELECT). Inoltre dispone di uno strumento di scripting per la scrittura e l'esecuzione di Procedure SQL, anteprima di quello che sarà il supporto alle Stored procedure di MySQL 5.0.

Altro programma da considerare è **Db Manager Professional** di Db Tools, scaricabile gratuitamente dal sito www.dbtools.com.br. Questo software permette di fare tutte le cose essenziali necessarie all'amministrazione di un Db in modo semplice con una Gui efficace e senza fronzoli. Inoltre può essere usato anche con server Postgres, quindi due piccioni....

Se proprio volete affrontare il mondo dei Db server da professionisti, quello che fa per voi allora è l'ottimo **Fabforce Db Designer 4**, un progetto OpenSource scaricabile da www.fabforce.net. Si tratta di un "disegnatore" di Database di alto livello, che concentra l'attenzione sulla struttura logica delle basi di dati, svincolandosi dal singolo motore di Db. Si interfaccia con MySql e con Oracle, ma anche con ODBC, quindi con quasi tutto. Tools dedicato a chi di Db è già esperto, e quindi sconsigliato ai deboli di cuore.