

MySQL per tutti

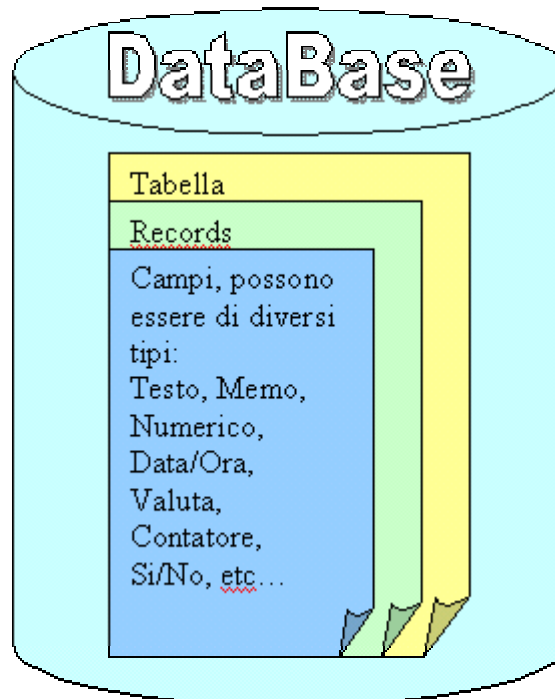
1 - Le basi

I database sono fondamentali nella programmazione di qualsiasi tipo di gestionali, questi permettono di organizzare in maniera ottimale i dati. Sui dati immagazzinati in un database si possono svolgere differenti operazioni, questi infatti possono essere modificati, aggiornati, aggiunti, eliminati o consultati, a seconda delle nostre necessità. Per svolgere queste operazioni si ricorre a un insieme di programmi detto DBMS (DataBase Management System) che in pratica è un software che fa da tramite tra l'utente e la base di dati, fornendo così una rappresentazione logica dei dati.

I DBMS forniscono diversi modi per rappresentare un database logicamente:

- Gerarchici
- Ad oggetti
- Relazionali

Qui li ho elencati in ordine di apparizione sul mercato. I gerarchici hanno caratterizzato i primi DBMS sviluppati intorno alla metà degli anni 70. Questo sistema è basato sugli alberi. E' tutt'ora utilizzato anche se in pochi casi. Il modello ad oggetti è dato da un insieme di classi, che definiscono le caratteristiche ed il comportamento degli oggetti del database. Questo permette di inserire nel database delle informazioni sulle operazioni da eseguire oltre ai dati. Ad oggi sfortunatamente manca un linguaggio standard per l'interrogazioni dei database basati sul modello ad oggetti. Il modello relazionale nato da Codd nel 1970 è stato però iniziato ad utilizzare intorno al 1980. Questo modello è basato sul principio di insieme ed è il modello che noi useremo in questo tutorial. I dati sono organizzati in delle tabelle bidimensionali composte da un numero fisso di colonne dette attributi ed in genere sono etichettati e da un numero fisso di righe dette enuple. Le righe identificano un record, cioè un insieme di campi che corrispondono al valore della colonna in quella riga. Qui di seguito segue una rappresentazione grafica sommaria di un database; quella più in alto nella seguente lista è formata da quella seguente e così via:



Approfondiamo ora l'argomento DBMS. Elenchiamo ora i suoi componenti software basilari :

- GESTORE DEGLI ACCESSI: trasforma le operazioni di accesso ai dati in chiamate a procedure del gestore della memoria fisica.
- PRECOMPILATORE DEL LINGUAGGIO OSPITE: svolge un'analisi delle istruzioni del Database Definition Language (DDL) e ne ricava le specifiche chiamate alle procedure del DBMS e passa il codice trasformato al processore di interrogazioni.
- PROCESSORE DI INTERROGAZIONI: traduce le istruzioni dal linguaggio usato a quello del DBMS.
- COMPILATORE DELLE DEFINIZIONI DEI DATI: riceve le istruzioni dal linguaggio usato per definire la tipologia dei dati del database e genera il Dizionario dei dati.
- GESTORE DELLA MEMORIA FISICA: gestisce l'allocazione dei dati nella memoria di massa.
- INTERFACCIA UTENTI: permette all'utente di interagire col database.
- Il DBMS deve svolgere i seguenti compiti:
 - Deve controllare in dati in modo da evitarne la ridondanza.
 - Gestisce l'accesso ai dati attraverso uno schema concettuale, cioè deve fornire una organizzazione visiva del database indipendentemente dall'organizzazione fisica dei dati.
 - Deve gestire l'accesso concorrente ai dati in modo da evitare blocchi e errori.

I database che prenderemo in considerazione in questa lezione sono quelli realizzati con MySQL è DBMS completamente gratuito ed estremamente veloce nato originariamente su UNIX, ma oggi presente anche su Linux e Windows. DBMS sta per DataBase Management System. Spieghiamo un po' cosa sono i DataBase:

Dopo questa piccola parte teorica elenchiamo gli strumenti che vi serviranno durante il tutorial:

- Il server MySQL (il nome del file in genere è: Mysql-shareware-3.xx.xx-win.zip (oppure la versione beta: Mysql-3.xx.xx-beta-win.zip. Per Linux sarà qualcosa tipo: mysql-x.xx.xx-pc-linux-gnu-xxxx.tar.gz).
- Un editor di testo (il notepad va benissimo).
- Tanta passione e un po' di tempo.

Per la prima voce troverete tutto su www.mysql.org per la seconda voce ormai tutti i sistemi

operativi sono muniti di questo strumento (alcuni potrebbero preferire editor creati appositamente per MySQL, facilmente reperibili su molti siti web), per l'ultima voce sfortunatamente non so quali risorse consigliare.

2 - L'installazione

Una volta effettuato il download passiamo finalmente all'installazione, qui di seguito presento l'installazione a seconda del diverso sistema operativo:

Windows 9x

Scompattiamo il server SQL in una cartella di nome mysql (io terrò come a riferimento il drive c, quindi il percorso completo sarà c:\mysql). Apriamo il prompt di MS-DOS e scriviamo:

```
C:\> cd mysql
C:\mysql\> cd bin
C:\mysql\bin\>mysqld-shareware -language=italian
(se usiamo la beta sostituiamo la riga di sopra con:
C:\mysql\bin\>mysqld -language=italian ).
```

Fatto questo siamo nella shell di MYSQL e possiamo iniziare a lavorare.

Windows 2000

Scompattiamo il server SQL in una cartella di nome mysql (io terrò come a riferimento il drive c, quindi il percorso completo sarà c:\mysql). Apriamo il prompt di MS-DOS e scriviamo:

```
C:\> cd mysql
C:\mysql\> cd bin
C:\mysql\bin\>nysqld-nt - install
```

Tramite questo comando abbiamo creato un nuovo servizio sul server, ora non ci resta che avviarlo svolgendo le seguenti semplici operazioni:

1. Apriamo il pannello di controllo.
2. Selezioniamo la cartella "Strumenti di Amministrazione".
3. Avviamo la voce "Servizi".
4. Ci apparirà una lista, scorriamola fino a quando non troveremo la voce "MySQL" se non dovesse essere presente o abbiamo errato il processo di installazione o non è ancora aggiornata la finestra, in questo caso premiamo F5 per aggiornare.
5. Una volta selezionato il servizio dalla lista il servizio clicchiamo col tasto destro del mouse e selezioniamo la voce "Proprietà".
6. Ci apparirà una finestra clicchiamo su "Avvia" per avviare il servizio. Se vogliamo evitare di avviarlo a mano ad ogni avvio di Windows possiamo scegliere da "Tipo di avvio" la voce "Automatico".

Ora MySQL è perfettamente installato.

1. Prima di tutto scaricate l'apposito pacchetto dal solito www.mysql.org
2. Scompattate il file appena scaricato in una cartella (sempre di nome mysql per comodità)
3. Assicuratevi di essere amministratori del sistema e fate partire lo script di configurazione automatica tramite il comando:

```
./configure
```

4. Dopo una breve attesa per la configurazione automatica potete finalmente passare all'installazione vera e propria. Digitate prima

Make

e subito dopo

```
make install
```

5. Una volta installato create le tabelle di sistema. Per fare ciò spostatevi nella directory in cui avete installato MySQL, accedete alla sottodirectory /bin ed eseguite:

```
./mysqk_install_db
```

6. Infine non vi resta che far partire il demone del mysql digitando:

```
./safe_mysqld &
```

Il carattere & serve fa eseguire il demone in background.

7. Se non ricevete errori l'installazione è avvenuta con successo, se no ripete l'installazione da capo, forse avete saltato alcuni passaggi.

Ora possiamo finalmente eseguire il nostro MySQL anche sotto Linux. **ATTENZIONE!!!** Questo tutorial è stato scritto con il sistema Windows come riferimento, non ci dovrebbero essere differenze, ma ci tengo comunque a precisarlo.

3 - La configurazione

Una volta installato presenterà un user di default di nome "root" ed una password vuota. Per inserire la password scrivete:

```
C:\mysql\bin\>mysqladmin -u root password nomepassword
```

Naturalmente sostituiremo nomepassword con la password che vogliamo inserire. Per cambiare la password in qualsiasi momento scriveremo:

```
C:\mysql\bin\>mysqladmin -u root -p password nuovapassword
```

Dopo la pressione del tasto "Invio" vi sarà richiesta la vecchia password. Per creare un nuovo utente scriveremo invece:

Ora possiamo finalmente accedere all'SQL vero e proprio per inviare i comandi presenti nella tabella presente tra poche righe. Per accedere scriveremo:

```
C:\mysql\bin\>mysql -uroot -ppassword
```

Naturalmente root e password dovranno essere sostituite rispettivamente dal nostro username e dalla nostra password. Se avete seguito bene le istruzioni MySQL si eseguirà. Verrà aperto una specie di prompt di MS-DOS dove al posto di "C:\>" troveremo scritto "mysql>".

Questa interfaccia è a riga di comando. Ogni comando deve essere terminato da ; e da un invio. Il solo tasto invio non basta!.

Per uscire basterà digitare:

```
mysql> quit;
```

Tramite il comando "use" selezioniamo il database sul quale vogliamo lavorare:

```
mysql> use nomepagina;
```

Ma come faccio a sapere quale database utilizzare se non sappiamo quali sono presenti sul mio computer? Per fare questo basta ricorrere semplicemente al comando "show". Tramite la sintassi:

```
mysql> show databases;
```

Visualizziamo tutti i database presenti sulla nostra macchina, mentre se vogliamo visualizzare le tabelle da cui è formato basterà scrivere:

```
mysql> show tables;
```

Mentre se vogliamo vedere i campi che formano una tabella scriveremo:

```
mysql> show columns from nomeTabella;
```

Una volta che ci siamo resi conto dei database presenti nel nostro database possiamo anche selezionare quello che preferiamo tramite il già visto comando "use" per poi procedere a interrogarlo, ma prima di procedere analizziamo le voci principali dell'help che ci apparirà digitando \h o help o \?:

clear o \c cancella il comando;

exit o quit o \q esce.

\g o ; invia il comando SQL;

\. o source esegue uno script (vedremo in seguito come crearli);

\s o status visualizza le informazioni di stato del server;

\u è sinonimo dell'use già visto in precedenza.

Libreria necessaria per il richiamo in un linguaggio

Spesso per utilizzare MySQL in un sistema operativo sono necessarie delle librerie dette MyODBC. Per installare la dll sopra citata basta eseguire un normale setup. Se si hanno i sorgenti è necessario prima compilare il tutto tramite queste tre righe da scrivere nel prompt di MS-DOS:

```
C:\> cd myodbc3-src
```

```
C:\> nmake -f Win_Makefile
```

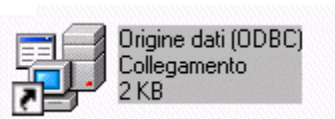
```
C:\> nmake -f Win_Makefile install
```

Perché questa installazione vada a buon fine bisogna assicurarsi di aver installato MDAC, Microsoft Data Access SDK scaricabile da: www.microsoft.com/data/ . Se invece ci troviamo sotto Linux la storia si allunga di non poco, per questo vi rimando alla guida di questa libreria reperibile su http://www.mysql.org/products/myodbc/manual_toc.html . Qui troverete tutte le informazioni che desiderate su questa libreria. Ho tralasciato Linux e non Windows perché questa guida è stata ideata pensando a Windows.

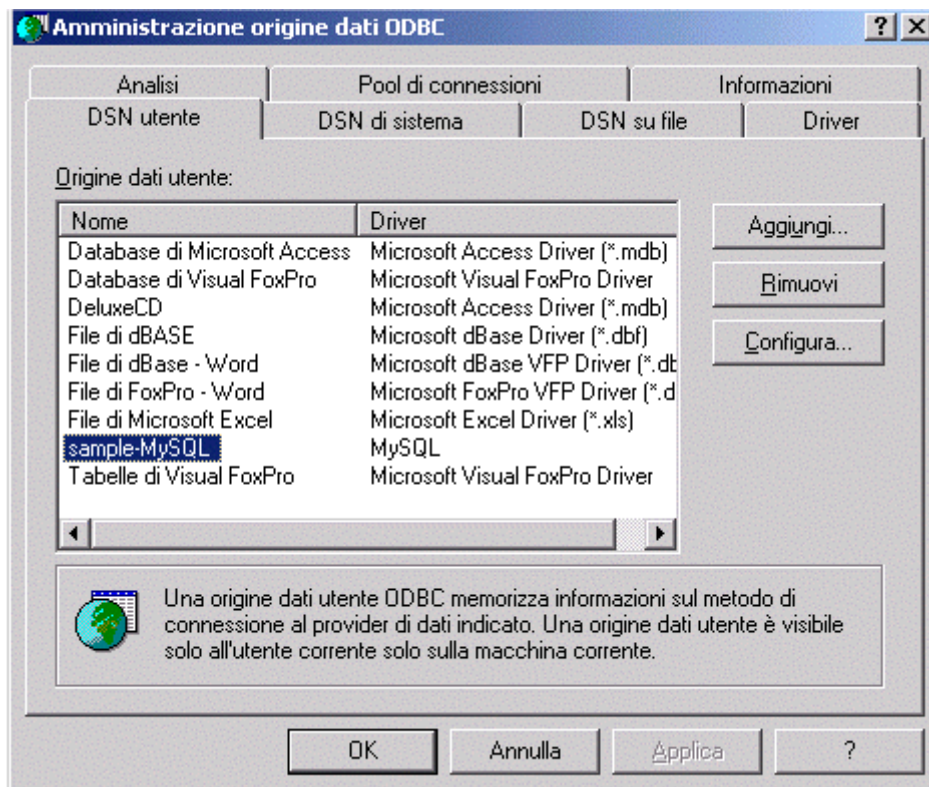
Se avete scaricato la versione già compilata basta scompattarla ed eseguire il setup.exe qui contenuto. Una volta terminata l'installazione aprite il Pannello di Controllo di Windows, cicate su Strumenti di Amministrazione



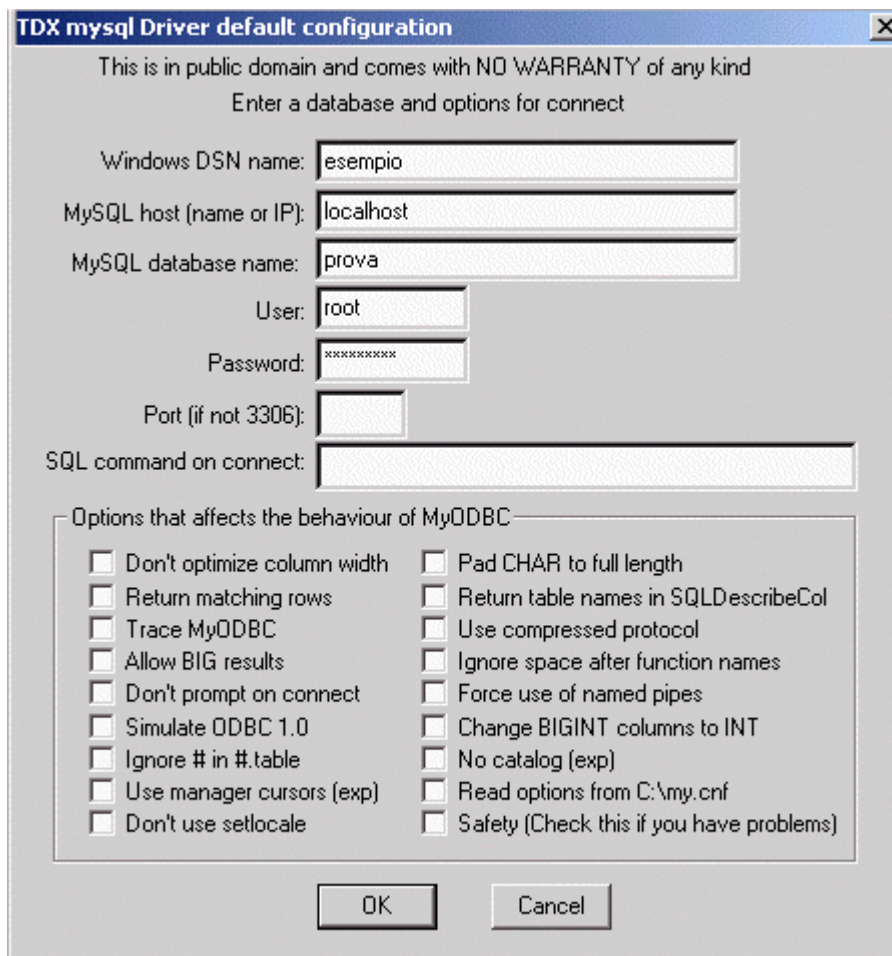
Poi cliccate su Origine Dati (ODBC)



Vi troverete davanti ad una finestra come la seguente:



Assicurateci dell'esistenza di questo DSN utente nel nostro sistema ne potremo creare uno nostro. Ci apparirà la seguente finestra:



Una volta completata potremo finalmente interrogare il nostro database MySQL tramite i linguaggi Microsoft™ compreso il nuovissimo ASP.NET. Le connessioni al database tramite i vari linguaggi le vedremo al punto sette di questo tutorial.

4 - Il nostro primo Database

Ora che abbiamo finalmente installato il nostro DBMS è giunto il momento di "giocarci" un po'. In questo paragrafo vedremo come creare un database, inserirci i dati e leggere i dati contenuti in esso. Per un riassunto dei principali comandi SQL vi rimando al paragrafo 4.1. Per prima cosa creiamo il nostro database, per fare questo dobbiamo usare il comando CREATE DATABASE. Scriviamo quindi:

```
mysql> CREATE DATABASE Clienti;
```

Ora il nostro database esiste. Scriviamo un show databases per verificare che sia stato veramente creato. Selezioniamo poi il database appena creato scrivendo:

```
mysql> use Clienti;
```

Ora dobbiamo inserirci le tabelle. Visto che questo è un esempio creeremo una sola tabella di nome anagrafe in cui inseriremo solo tre campi: Nome, Telefono, Codice Fiscale. Inseriamo il codice fiscale perché essendo un identificativo univoco ci sarà molto utile nel caso noi volessimo fare ricerche nel nostro database. L'istruzione per creare il database è:

```
mysql> CREATE TABLE anagrafe(nome char(35) not null, telefono int not null, cod_fis char(16)
```

not null);

Il not null indica che i campi non possono essere vuoti, omettendolo perdiamo questa clausola. Ora abbiamo creato la nostra tabella. Facciamo un show tables per vedere se tutto è andato per il giusto verso. Ora la tabella esiste ma è vuota. Inseriamoci dei dati. Per fare ciò ricorriamo all'istruzione INSERT INTO:

```
mysql> INSERT INTO anagrafe (nome, telefono, cod_fis) VALUES('Davide Cerbo', 0892419, 'CRBDVD...');  
mysql> INSERT INTO anagrafe (nome, telefono, cod_fis) VALUES ('Laura Rispoli', 0892241, 'RSPLUR...');  
mysql> INSERT INTO anagrafe (nome, telefono, cod_fis) VALUES ('Pinco Pallino', 0892324, 'PLLPNC...');
```

Notiamo alcune cose su questa istruzione. I valori dei campi dopo il VALUES vanno inseriti nell'ordine in cui sono citati i campi precedentemente. Le stringhe vanno inserite tra apici, i valori numerici no. Altra cosa che balza subito all'occhio è che queste istruzioni sono simili. Potrebbe risultare noioso scriverle tutte ogni volta. Per evitare questo basta usare i cursori della tastiera. Con le frecce su e giù si va alla riga precedente o successiva, mentre con le frecce destra e sinistra ci si sposta per la stringa, permettendoci quindi di modificare la stringa e rimandarla in esecuzione, senza doverla riscrivere da capo; più avanti vedremo un metodo ancora più veloce per ottimizzare i valori ripetitivi. Ora che i dati nel nostro database sono inseriti possiamo visualizzarli tramite l'istruzione SELECT. Scriviamo quindi...

```
mysql> SELECT * FROM anagrafe;
```

... e godiamoci lo spettacolo:).

Qui termina il nostro esempio guidato. Ora tocca alla vostra fantasia che potrete ampiamente sbizzarrire leggendo per bene il paragrafo 4.1 che è a dir poco fondamentale.

I comandi SQL

Scarica i comandi SQL

5 - Creare nuovi utenti in MySQL

Per creare utenti in MySQL si possono scegliere due strade. La prima consiste nel creare l'utente col GRANT, cioè mentre stiamo fornendo gli accessi ad un determinato utente. La sintassi del GRANT cambierà quindi in:

```
GRANT tipoprivilegio ON nomedatabase.nomeabella TO utente@host IDENTIFIED BY 'nuovapassword' WITH GRANT OPTION;
```

Le parole in blu da sostituire credo siano abbastanza chiare, fatta eccezione per host. Ad host dovete sostituire l'indirizzo IP o il nome dell'host dal quale è consentita la connessione. Se al posto di host scriviamo % la connessione sarà accettata da qualsiasi host. Se il server gira in locale, l'host della macchina è localhost. Altro metodo per creare un nuovo utente è quello di interagire direttamente sul database degli utenti. Per fare ciò dobbiamo andare a modificare la tabella user presente nel database mysql (creato di default durante l'installazione. Non cancellatelo per nessun motivo mi raccomando! Questo contiene informazioni basilari per il funzionamento di MySQL). Quindi apriamo il database in questione scrivendo:

```
mysql> use mysql;
```

Effettuiamo poi il seguente INSERT INTO:

```
INSERT INTO user VALUES('host','nomeutente',PASSWORD('password'), 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X');
```

oppure:

```
INSERT INTO user SET Host='host',User='nome utente', Password = PASSWORD(password),  
Select_priv= 'X', Insert_priv= 'X',Update_priv = 'X',Delete_priv= 'X',Create_priv  
='X',Drop_priv= 'X',Reload_priv= 'X',Shutdown_priv= 'X',Process_priv= 'X',File_priv=  
'X',Grant_priv= 'X', References_priv="X",Index_priv= 'X',Alter_priv = 'X';
```

Il numero degli argomenti può variare da versione a versione quindi è sempre meglio fare un semplice SELECT per visualizzare tutti i campi prima. Alla X va sostituito o Y o N, a seconda se vogliamo concedere oppure no i permessi a quell'utente. Se inseriamo tutte Y creiamo un SuperUser. Una volta fatto questo INSERT INTO dobbiamo fare un FLUSH (questo comando aggiorna le variabili di sistema, in questo caso quelle riguardanti i privilegi) scrivendo:

```
mysql> FLUSH PRIVILEGES;
```

6 - I tipi disponibili in MySQL

Quando creiamo il nostro database naturalmente dobbiamo definire i tipi dei campi. I tipi dei campi che usa MySQL sono i seguenti:

- TINYINT - Valore numerico compreso tra -128 e 127.
- TINYINT UNSIGNED - Valore numerico compreso tra 0 e 255.
- SMALLINT - Valore numerico compreso tra -32768 e 32767.
- SMALLINT UNSIGNED - Valore numerico compreso tra 0 e 65535.
- MEDIUM - Valore numerico compreso tra -8388608 e 8388607.
- MEDIUM UNSIGNED - Valore numerico compreso tra 0 e 16777215.
- INT o INTEGER - Valore numerico compreso tra -2147483648 e 2147483647.
- INT UNSIGNED - Valore numerico compreso tra 0 e 4294967295.
- BIGINT - Valore numerico compreso tra -9223372036854775808 e 9223372036854775807.
- BIGINT UNSIGNED - Valore numerico compreso tra 0 e 18446744073709551615.
- FLOAT (m,d) - Valore numerico con valori decimali. Compreso tra -3.402823466E+38 e -1.175494351E-38.
- DOUBLE(m,d) o REAL(m,d) - Valore numerico con valori decimali. Compreso tra -1.7976931348623157E+308 e -2.2250738585072014E-308.
- DECIMAL (m,d) o NUMERIC(m,d) - Valore numerico con decimali che sarà memorizzato come una serie di caratteri.
- DATE - Campo contenente una data in formato "aaaa-mm-gg".
- DATETIME - Campo contenente la data e ora, informato "aaaa-mm-gg hh:mm:ss".
- TIME - Campo contenente un'ora nel formato: "hh:mm:ss".
- CHAR - Campo contenente una stringa di caratteri che può avere la massima lunghezza di 255.
- TINYBLOB o TINYTEXT - Campo che può avere la massima lunghezza di 255 in formato binario nel primo caso o testo nel secondo.
- BLOB o TEXT - Campo che può avere la massima lunghezza di 65535 in formato binario nel primo caso o testo nel secondo.
- MEDIUMBLOB o MEDIUMTEXT - Campo che può avere la massima lunghezza di 16777215 in formato binario nel primo caso o testo nel secondo.

- LONGBLOB o LONGTEXT - Campo che può avere la massima lunghezza di 4294967295 in formato binario nel primo caso o testo nel secondo.
- ENUM ('value1','value2',...) - Campo che può contenere uno dei valori presente tra le parentesi.

7 - Gli script: rendiamo il lavoro sequenziale

Spesso ci capita di dover fare lunghe sequenze di comandi SQL, ed è un po' scomodo scriverli nel prompt che ci offre MySQL di base, per questo possiamo ricorrere a diversi strumenti facilmente trovabili sul web, ma basta anche un semplice blocco note. Nel file di testo che andremo a creare dovremo inserire tutti i comandi che desideriamo creando un vero e proprio file batch. Fatto questo torniamo al nostro prompt di MySQL e digitiamo:

```
mysql> \.
```

Questa opzione è molto comoda nel caso in cui ci trovassimo a dover fare diversi comandi ripetitivi (copia e incolla docet) e ci permette anche di costruirci nel linguaggio che preferiamo il nostro piccolo editor di script.

8 - MySQL utilizzato in diversi linguaggi di programmazione: esempi pratici.

MySQL nei linguaggi di programmazione

9 - Conclusione e ringraziamenti

Detto questo metto momentaneamente la parola fine al mio tutorial, sperando di aggiornarlo presto con altri esempi dedicati alla programmazione, una serie di domande comuni ed una raccolta di "trucchi del mestiere". Naturalmente ogni informazione aggiuntiva, segnalazione di errore, domanda è ben accetta. Distribuisco questo tutorial in un formato modificabile, proprio nella speranza di riceverne versioni ampliate e modificate nella mia casella di posta elettronica dagli stessi utenti che lo leggono.