Introduzione a MySQL

Alessandro Bugatti (alessandro.bugatti@istruzione.it)

15 gennaio 2014

1 MySQL

MySQL è un server di database (un RDBMS) open-source che verrà usato quest'anno per comprendere il funzionamento dei DBMS. Questo strumento è stato scelto fra le altre cose perchè:

- è gratuito
- è veloce
- è piuttosto facile da amministrare
- supporta SQL (anche se non completamente a seconda della versione)
- ci si può connettere dalla rete ed è sicuro (per quanto lo possa essere un'applicazione in rete)
- è portabile (vari sistemi UNIX, Windows, Mac Os)

L'installazione in ambiente Windows è piuttosto semplice e nel nostro caso viene eseguita tramite il pacchetto xampp presente sul CD distribuito a inizio anno. Una volta installato MySQL deve essere avviato, essendo un processo server. Questo vuol dire che una volta avviato verrà eseguito in background e rimarrà in attesa di connessioni sulla porta di default 3306, permettendo sia a client sulla stessa macchina che ad altre macchine presenti in rete di usufruire dei servizi messi a disposizione (tutto ciò che si può fare con un DBMS, cioè creare tabelle, inserire dati, interrogare, ecc.). Per dialogare con MySQL i client utilizzeranno quella parte di SQL implementata all'interno di MySQL.

2 Connettersi a MySQL tramite client a riga di comando

Nella distribuzione standard di MySQL oltre al programma server vengono incluse una serie di utilità per la gestione di vari aspetti legati all'amministrazione di un database. L'utilità che utilizzeremo per accedere al processo server sarà **mysql.exe**, un programma che si trova all'interno della cartella *mysql|bin*, che a sua volta si trova in un percorso che dipende dall'installazione (tipicamente se si è utilizzato XAMPP potrebbe essere in *c:|xampp*).

Prima di poter accedere al database occorre però creare un account, cosa che si può fare solo se si ha già un altro account. Questo "problema" è risolto dal fatto

che alla prima installazione MySQL ha di default un utente *root* senza password con permessi illimitati (ovviamente questo pone ovvi problemi di sicurezza e quindi un buon amministratore dovrebbe quanto prima settare una password per root, cosa che noi non faremo sulle macchine del laboratorio per non lasciare "fuori dalla porta" tutti gli altri utilizzatori del server).

Quindi al primo utilizzo noi scriveremo:

c:\percorsocorretto\mysql -h *nomehost* **-p -u** *root*

dove *nomehost* sarà il nome (o l'indirizzo IP) del computer sul quale risiede il server (nel nostro caso può essere *localhost* se si sta lavorando in locale o l'indirizzo del server negli altri casi) e *root* è appunto l'account con permessi illimitati che ci permetterà di creare un nuovo account. L'opzione **-p** indica che il server richiederà la password[1] (in questo caso potremmo anche ometterla, ma negli esempi successivi no).

Ignorando la richiesta di password e schiacciando invio otterremo qualcosa di simile (dipende dalla versione installata) al seguente messaggio:

> Welcome to the MySQL monitor. Commands end with ; or \g.
>
> Your MySQL connection id is 51 Server version: 5.5.34-0ubuntu0.12.04.1 (Ubuntu)
>
> Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
>
> Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
>
> Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
>
> mysql>

L'ultima riga indica che siamo in una shell interattiva attraverso la quale possiamo inviare dei comandi al server.

L'istruzione SQL che dovremo lanciare per creare un nuovo utente (per il momento non indagheremo più di tanto sulla sua sintassi che verrà vista in seguito) sarà:

```
mysql> GRANT ALL ON nomedb.* TO 'utente'@'nomemacchina' IDENTIFIED
    BY 'password';
```

dove *nomedb* dovrà essere sostituito con il nome del database che si intende creare (oppure che esiste già), *utente* con il nome dell'utente che verrà successivamente utilizzato per connettersi al server, *nomemacchina* con il nome o l'indirizzo IP della macchina sulla quale si sta lavorando e *password* con la password che si intende utilizzare.

Per il resto della dispensa supporremo di avere creato un utente *user* sulla macchina *localhost* che può lavorare sul database *prova* con la password *prova* (ovviamente ognuno dovrà modificare gli esempi a seconda di ciò che ha creato).

[1] Se la versione server di MySQL è la 4.1 o successiva e il client è di una versione precedente questa opzione darà dei problemi, poichè la gestione delle password è stata modificata utilizzando un diverso meccanismo di hashing. Ulteriori informazioni possono essere trovate nel manuale di MySQL.

A questo punto è possibile disconnettersi dal server (sul quale ricordo eravamo connessi come *root*) e procedere a lavorare con il nuovo account che abbiamo creato. In alternativa è possibile saltare il passaggio della creazione del nuovo utente e utilizzare l'utente *root* senza password di default, stando comunque attenti alle implicazioni che ciò ha da un punto di vista della sicurezza.

3 Creare un database

L'istruzione vista in precedenza in realtà non serve a creare un database, ma dice solo che il nostro utente avrà tutti i permessi su un certo database quando questo esisterà. Per procedere quindi a creare il nostro database *prova* dovremo come prima cosa connetterci come *user* in questo modo:

c:\percorsocorretto\mysql -h *nomehost* **-p -u** *user*

Dopo aver inserito la password richiesta saremo nella shell interattiva e a questo punto basterà lanciare il comando:

```
mysql> CREATE DATABASE prova;
```

per creare il database *prova* (**attenzione**: nella shell il punto e virgola serve a definire la fine di un comando e quindi finchè non viene inserito la pressione del tasto invio causerà solo il ritorno a capo ma non l'effettiva esecuzione del comando). Come possiamo essere sicuri che sia stato veramente creato? Possiamo eseguire il seguente comando:

```
mysql> SHOW DATABASES;
```

che mostrerà tutti i database presenti nel server e dovremmo vedere quello appena creato (se non fosse possibile a causa dello scrolling della pagina sarà sufficiente tornare indietro con la barra laterale).

Aver creato un database non implica automaticamente che tutte le istruzioni che lanceremo da questo momento in poi lavoreranno all'interno di quel database, bisogna invece ricordarsi di indicare qual è il database sul quale vogliamo lavorare con l'istruzione:

```
mysql> USE prova;
```

4 Creare le tabelle

Supponiamo adesso di voler creare all'interno del database *prova* una tabella molto semplice chiamandola *studenti* che contenga nome, cognome, indirizzo e data di nascita di una serie di studenti.

L'istruzione che ci permetterà di fare questo sarà:

```
CREATE TABLE studenti
(
    Nome VARCHAR(20) NOT NULL,

    Cognome VARCHAR(20) NOT NULL,
```

```
    Indirizzo VARCHAR(40) NOT NULL,

    DataNascita DATE NOT NULL
);
```

dove VARCHAR(n) rappresenta una stringa di caratteri di lunghezza variabile con una dimensione massima di n e DATE una data (nel formato *SSAA-MM-GG* dove *SS* è il secolo, *AA* l'anno, *MM* il mese e *GG* il giorno). L'attributo NOT NULL indica che quei campi devono per forza contenere un valore (gli altri tipi di dati e di attributi utilizzabili in MySQL verranno mostrati nella prossima dispensa).

A questo punto è possibile utilizzare le istruzioni DML per inserire, modificare o cancellare dei record.