

Laurea in Ingegneria Informatica – SAPIENZA Università di Roma

Insegnamento di Basi di Dati

Esercitazione:

Il DBMS MySQL

Domenico Fabio Savo

www.dis.uniroma1.it/~savo/

Cosa vedremo

1. Presentazione del DBMS MySQL
2. Come scaricare ed installare MySQL
3. Il “client mysql”
4. Creazione e gestione di una base di dati
5. Creazione e gestione delle tabelle
6. Esercitazione sulle interrogazione di una base di dati

Il DBMS MySQL

- ▶ MySQL è un DBMS open-source disponibile gratuitamente su <http://dev.mysql.com/downloads/>
- ▶ In questa esercitazione si farà riferimento alla versione MySQL 5 per Windows.
(è possibile utilizzare MySQL anche su sistemi Linux e MacOS)

Installazione di MySQL

Per l'installazione procediamo come segue:

1. Scaricare il pacchetto **Windows Essentials** dal sito <http://dev.mysql.com/downloads/>
2. Eseguire il file `mysql-essential-5.0.67-win32.msi`, selezionare installazione "**typical**".
3. Dopo il termine dell'installazione è possibile lanciare la **Configuration Wizard** per configurare immediatamente il nostro server MySQL.
4. Selezionare la configurazione "**Standard**".
5. Selezionare le check box per:
 - a) eseguire MySQL come servizio;
 - b) lanciare automaticamente MySQL all'avvio;
 - c) includere la directory 'bin' nel path di Windows.
6. L'ultima schermata ci consente di impostare la password di **root**, tale password ci consentirà di amministrare il server.
7. Al termine l'installazione e la configurazione sono completate.

Il “client mysql”

- ▶ **Client mysql** è il programma client a riga di comando che consente di collegarsi al server MySQL per sfruttarne le funzionalità.
(viene installato insieme al server MySQL)
- ▶ Dalla pagina web <http://dev.mysql.com/downloads/gui-tools/5.0.html> è possibile scaricare ed installare dei client grafici, chiamati **MySQL GUI Tools**, che forniscono una interfaccia grafica intuitiva per la gestione e l'interrogazione delle basi di dati gestite dal DBMS MySQL 5

Il “client mysql”

- ▶ Per lanciare il client mysql è sufficiente richiamarlo dal prompt indicandogli utenza e password:

```
shell> mysql --user=root --password=xxx
```

oppure:

```
shell> mysql -uroot -p
```

In questo caso sarà il programma a chiedervi di introdurre la password senza visualizzarla.

- ▶ Una volta connessi alla base di dati appare il prompt di mysql:

```
mysql>
```

A questo punto si possono digitare i comandi SQL che andranno ad operare sul DBMS a cui si è connessi.

```
mysql> quit
```

- ▶ Per chiudere il client digitiamo:

I permessi in MySQL

- ▶ Una volta connessi al server, un utente deve possedere i permessi necessari per lavorare sui vari database.
- ▶ Per chiedere quali basi di dati gestite dal DBMS sono accessibili dall'utente **root** utilizziamo il comando:

```
mysql> show databases;

+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.05 sec)
```

Creazione di una base di dati

- ▶ Per poter creare un nuovo database su cui lavorare utilizziamo il comando:

```
CREATE DATABASE [IF NOT EXIST] nome_db
```

- ▶ Con l'opzione **IF NOT EXIST** possiamo evitare la segnalazione di errore nel caso esista già un database con lo stesso nome.
- ▶ Per eliminare un database si utilizza l'istruzione:

```
DROP DATABASE [IF EXIST] nome_db
```

- ▶ Con l'opzione **IF EXIST** possiamo evitare la segnalazione di errore nel caso non esista un database chiamato **nome_db**.

ES: Creazione di un database

Creiamo un database denominato “esempio” utilizzando il “client mysql”.

Le istruzioni da utilizzare sono:

```
mysql> CREATE DATABASE esempio;  
  
Query OK, 1 row effected (0.06 sec)
```

Ora i database gestiti dall'utente *root* sono:

```
mysql> show databases;  
  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| test |  
| esempio |  
+-----+  
4 rows in set (0.05 sec)
```

Importare i comandi

Anzichè eseguire comandi SQL digitandoli su terminale è spesso più conveniente scriverli in un file di testo e poi richiamarli dall'interprete dei comandi MySQL.

Supponiamo di aver scritto alcuni comandi SQL in un file **miaquery.sql** nella directory corrente. Possiamo eseguire il file da MySQL con il comando:

```
mysql> source miaquery.sql
```

Ovviamente è possibile anche specificare il path completo del file.

Creazione delle tabelle (1 / 4)

- ▶ Per selezionare il database su cui effettuare le modifiche usare il comando

`USE nome_database`

- ▶ L'istruzione per definire uno schema di relazione (specificando attributi e vincoli) in MySQL è

```
CREATE TABLE [IF NOT EXISTS] nome_tabella  
[(  
    [definizione attributi]  
    [opzioni di tabella]  
)]
```

- ▶ La tabella viene creata nel database in uso, è possibile indicare espressamente in quale database creare la tabella usando *nome_db.nome_tabella*.
- ▶ *IF NOT EXISTS* si usa per evitare messaggi di errore nel caso la tabella esista già.

Creazione delle tabelle (2/4)

```
CREATE TABLE [IF NOT EXISTS] nome_tabella
[(
  [definizione attributi]
  [opzioni di tabella]
)]
```

- ▶ Le **definizioni attributi** si riferiscono agli attributi della tabella, la loro sintassi è:

nome_colonna TIPO

[NOT NULL | NULL] (di default può contenere valori NULL)

[DEFAULT valore] (usato per impostare un valore di default)

[AUTO_INCREMENT] (per attributi di tipo intero per avere un valore sequenziale generato automaticamente)

[UNIQUE | [PRIMARY] KEY] (UNIQUE rappresenta un indice che non può contenere valori duplicati, PRIMARY KEY indica la chiave primaria, oltre a non ammettere duplicati non può contenere valori NULL)

[reference_definition] ()

Creazione delle tabelle (3/4)

```
CREATE TABLE [IF NOT EXISTS] nome_tabella  
[(  
    [definizione attributi]  
    [opzioni di tabella]  
)]
```

▶ **Reference_definition**

Tramite le **reference_definition** è possibile definire vincoli di integrità referenziale, ovvero l'attributo su cui è definito può assumere solo valori specificati nell'attributo di un'altra tabella.

```
REFERENCES nome_tabella [(colonna_indice,...)]
```

Creazione delle tabelle (4/4)

```
CREATE TABLE [IF NOT EXISTS] nome_tabella
[(
  [definizione attributi]
  [opzioni di tabella]
)]
```

- ▶ Le **opzioni tabella** si riferiscono all'intera tabella e permettono di definire diverse proprietà di questa.

- ▶ Le più importanti sono:

PRIMARY KEY (nome_attributo1, nome_attributo2,...)

Permette di definire come chiave primaria della tabella un insieme di attributi di questa.

INDEX (nome_attributo1, nome_attributo2,...)

Permette di definire degli indici su uno o più attributi della tabella

FOREIGN KEY (nome_att1, nome_att2,...)

REFERENCE nome_tab(nome_att1, nome_att2,...)

Permette di definire vincoli di integrità referenziale su più attributi

Visualizzare lo schema di una tabella

Per visualizzare lo schema di una tabella utilizzando il client “mysql” si procede come segue:

1. Ci si connette alla base di dati che contiene la tabella (se non è già stato fatto in precedenza) usando l’istruzione:

`USE nome_database`

2. Per visualizzare lo schema della tabella `nome_tabella` si utilizza poi l’istruzione

`DESCRIBE nome_tabella`

ES: Creazione di una tabella (1 / 2)

Vogliamo creare le seguenti tabelle:

- **persone(nome, reddito, eta, sesso)**
 - nome è una stringa di 20 caratteri (chiave primaria)
 - reddito è un intero di 10 cifre
 - eta è un intero di 3 cifre
 - sesso è un carattere
- **genitori(figlio,genitore)**
 - figlio (stringa di 20 caratteri, chiave esterna su PERSONE)
 - genitore (stringa di 20 caratteri, chiave esterna su PERSONE)
 - chiave primaria formata da “figlio” e “genitore”

ES: Creazione di una tabella (2/2)

- Creazione tabella Persone:

```
mysql> CREATE TABLE Persone(  
        Nome          CHARACTER(20) PRIMARY KEY,  
        Reddito       NUMERIC(10),  
        Eta           NUMERIC(3),  
        Sesso         CHARACTER,  
        );
```

- Creazione tabella Genitori:

```
mysql> CREATE TABLE Genitori(  
        Figlio        CHARACTER(20) REFERENCES PERSONE (Nome),  
        Genitore      CHARACTER(20) REFERENCES PERSONE (Nome),  
        PRIMARY KEY (Figlio, Genitore)  
        );
```

Modificare una tabella

È possibile modificare una tabella attraverso il comando **ALTER TABLE**:

```
ALTER TABLE nome_tabella
  ADD nome_attributo TIPO |
  ADD INDEX (nome_attributo,...) |
  ADD [CONSTRAINT [nome_vincolo]] PRIMARY KEY (nome_attributo,...) |
  ADD [CONSTRAINT [nome_vincolo]] UNIQUE (nome_attributo,...)
  ADD [CONSTRAINT [nome_vincolo]]
      FOREIGN KEY (colonna_indice,...) [reference_definition] |
  CHANGE vecchio_attributo nuovo_attributo TIPO |
  DROP nome_attributo
  DROP PRIMARY KEY
  DROP INDEX nome_attributo
  .....
  .....
```

ES: Modifica di una tabella

Date le tabelle:

- ▶ Aziende(Nome,Sede,Capitale)
- ▶ GruppoAziendale(Nome,Capogruppo)

Vogliamo aggiungere alla tabella GruppoAziendale un vincolo di chiave esterna sull'attributo Capogruppo su Aziende

```
mysql> ALTER TABLE GruppoAziendale
ADD CONSTRAINT fk_capogruppo
FOREIGN KEY (Capogruppo)
REFERENCES Aziende(Nome);
```

Eliminare una tabella

È possibile eliminare una o più tabelle utilizzando il comando:

```
DROP TABLE [IF EXISTS] nome_tabella [, nome_tabella]
```

Con l'opzione **IF EXISTS** possiamo evitare la segnalazione di errore nel caso non esista una tabella chiamata **nome_tabella**.

Esempio:

```
mysql> DROP TABLE Persone;
```