



MySQL: principali comandi

Ing. Davide Vetrone



Sommario



- Introduzione a MySQL e alle sue licenze
- Connessione/Disconnessione da un database MySQL
- Creazione utenti
- Programmi client
- SQL in MySQL DDL e DML
- Tipi di dati
- Load data

- DBMS Relazionale
- Struttura Client/Server
- Motore MULTITHREAD
 - un thread fisso controlla le connessioni in ingresso
 - un thread attivo per ogni connessione
- Interfaccia ODBC
- Portabile su qualsiasi piattaforma

MySQL 5

Performance

- Stored Procedures
- Cluster query push down
- Query optimizations
- Archive Engine
- InnoDB storage improvements

Faster



Reliability

- SQL Mode
- Triggers
- View
- Precision Math
- Distributed Transactions
- Cluster object support

Better



Ease of Use

- Instance Manager
- Information Schema
- Cursors
- Enhanced GUI Tools
- Migration Toolkit

Increased
Manageability



- MySQL viene rilasciato con:
 - Licenza GNU GPL :
 - Utilizzo open source di MySQL; le applicazioni basate su MySQL sono distribuite con la stessa licenza.
 - Licenza commerciale (conosciuta come OEM License)
 - Utilizzo commerciale di MySQL; le applicazioni basate su MySQL possono essere distribuite senza sorgenti.



Free

Subscription

OEM License

Community Edition

- ✓ Software tested by Community
- ✓ Basic MySQL AB testing
- ✓ Available under GPL
- ✓ Release early & often cycle (3 to 4 weeks)
- ✓ Bleeding edge
- ✓ No Maintenance
- ✓ SLA Not supported
- ✓ No ISV certification

Great for open source developers & technology enthusiasts

MySQL Network

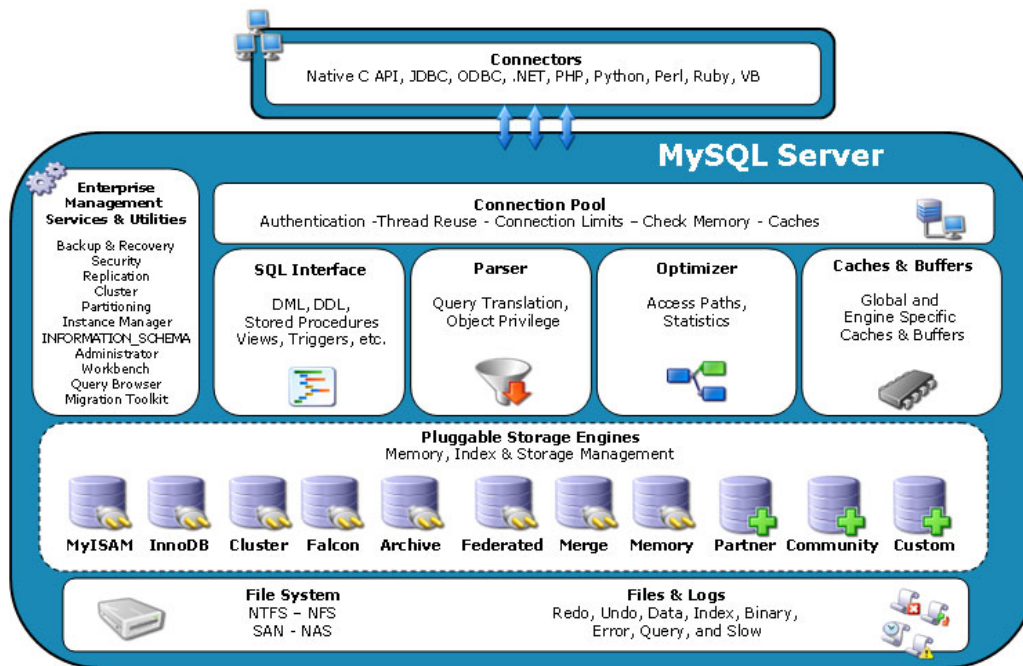
- ✓ Software fully tested, certified and optimized by MySQL AB
- ✓ PRO Edition (Transactions enabled)
- ✓ RTU is via the GPL Enterprise release cycles (6 to 9 months)
- ✓ Annual Subscription
- ✓ Update Advisor
- ✓ Technical Alert Advisor
- ✓ Knowledge Base
- ✓ Production Support
- ✓ Indemnification

Great for enterprise customers

Perpetual, non GPL License

- ✓ Software fully tested, certified and optimized by MySQL AB
- ✓ Sales to OEMs/ISVs and Enterprises
- ✓ One time fee, non-GPL non-viral license, no product updates
- ✓ Optional Support Agreement available (break/fix)
- ✓ Developer Support via Prof Svcs
- ✓ Updates via optional support agreement

Great for OEMs, ISVs & VARs



- Per connettersi al server è necessario fornire login e password

```
shell> mysql -h host -u user -p
Enter password: *****
```

- *host* and *user* rappresentano:
 - l'hostname dove risiede MySQL;
 - lo username di un utente che possiede un account sul server;
- *-p* specifica al server la richiesta della password all'utente.

```
C:\>mysql -u davide -p
Enter password: *****

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13 to server version: 5.0.27-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

- Per avere l'help:

```
C:\>mysql - help
```

- Per uscire:

```
mysql> QUIT
```

```
CREATE USER davide@localhost;
CREATE USER davide@localhost IDENTIFIED BY 'password';
DROP USER davide@localhost;
```

```
SET PASSWORD = PASSWORD('pw');
```

```
SET PASSWORD FOR davide@localhost = PASSWORD('pw');
```

} richiedono il
permesso
CREATE USER

è possibile per chiunque
sia collegato come
utente non anonimo

richiede il permesso
UPDATE sul database
mysql

1. Crea un utente senza password
2. Crea un utente con password
3. Elimina un utente
4. Imposta la password 'pw' per l'utente collegato
5. Imposta la password 'pw' per l'utente davide@localhost

- Tipicamente viene eseguita dall'utente *root*, mediante l'uso del comando GRANT.

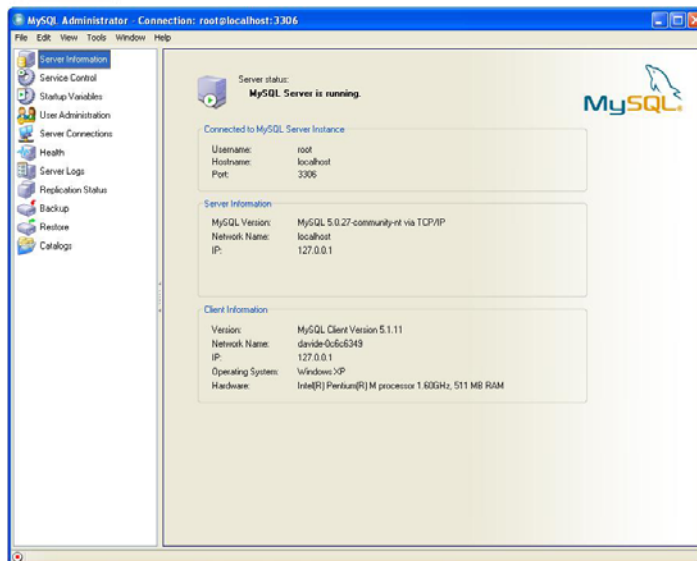
```
C:\>mysql -u root -p
Enter password: *****

mysql> GRANT ALL ON nomeDB.* to
-> 'user'@'localhost' IDENTIFIED BY
-> 'nome_password' ;
```

- Per consentire la connessione da un server specifico.

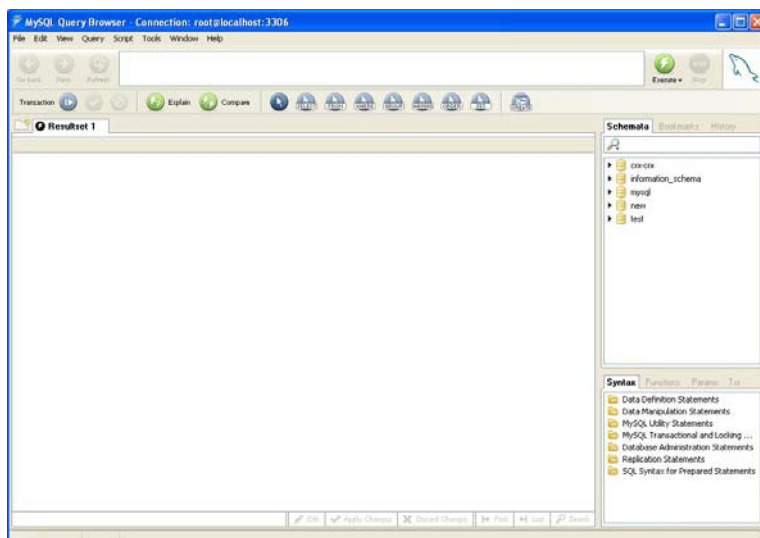
```
mysql> GRANT ALL ON nomeDB.* to
-> 'user'@'nome_server' IDENTIFIED BY
-> 'nome_password' ;
```

MySQL Administrator



- Amministrazione del database
- degli utenti
- operazioni pianificate
- carico del server

Query Browser



- Esecuzione di svariati tipi di query

SELECT	Estrazione Dati
INSERT UPDATE DELETE	Data manipulation language (DML): servono a manipolare i dati (interrogazione, modifica, rimozione)
CREATE ALTER DROP RENAME TRUNCATE	Data definition language (DDL): servono a definire le tabelle, e gli elementi accessori
COMMIT ROLLBACK SAVEPOINT	Controllo Transazioni
GRANT REVOKE	Data control language (DCL): servono a definire il controllo dell'accesso dei dati contenuti nel database.

Prompt	Significato
<code>mysql></code>	Pronto per ricevere un comando;
<code>-></code>	In attesa della successiva linea per un comando multiple-line.
<code>'></code>	In attesa della successiva linea per un comando multiple-line, apice di apertura nella riga precedente (').
<code>"></code>	In attesa della successiva linea per un comando multiple-line, doppio apice di apertura nella riga precedente (").
<code>`></code>	In attesa della successiva linea per un comando multiple-line, che fa riferimento ad un identificatore che inizia con backtick (`).

CREATE DATABASE



```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name  
    [create_specification [, create_specification] ...]
```

```
create_specification: [DEFAULT] CHARACTER SET  
    charset_name | [DEFAULT] COLLATE collation_name
```

- CREATE DATABASE crea un database con il nome dato
- Per poter eseguire il comando bisogna avere il privilegio CREATE per creazione database

DROP DATABASE



```
DROP {DATABASE | SCHEMA} [IF  
    EXISTS] db_name
```

- Cancella il database e tutte le sue tabelle

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS]  
tbl_name [(create_definition,...)]  
[table_options] [select_statement]
```

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS]  
tbl_name [( ) LIKE old_tbl_name ( )];
```

- CREATE TABLE crea una tabella nel db
- Per poter eseguire il comando bisogna avere il privilegio CREATE per le tabelle

- Nella CREATE TABLE può essere usata la clausola TEMPORARY. Una tabella TEMPORARY è visibile solo nella connessione corrente all'uscita la tabella è rimossa.
- Bisogna avere il privilegio CREATE TEMPORARY TABLES.
- Può essere specificata la parola chiave IF NOT EXISTS per evitare errori se la tabella già esiste.
 - Nota: non viene verificato che la tabella che si sta creando ha la stessa struttura di quella esistente.

- MySQL memorizza il formato di ogni tabella come un file *'frm'* nella directory del db. Possono essere creati anche altri file...
- Nel caso di tabelle di tipo MyISAM il sistema crea tre file per una tabella *tbl_name*:
 - *tbl_name.frm* File di definizione del formato
 - *tbl_name.MYD* Data file
 - *tbl_name.MYI* Index file

Storage Engine	Descrizione
ISAM	Primo storage engine di MySQL
HEAP (MEMORY)	I dati della seguente tabella sono solamente memorizzati nella memoria principale
MERGE	Una collezione di tabelle MyISAM usate come una singola tabella
MyISAM (MRG_MyISAM)	Storage engine binario portabile. Evoluzione di ISAM
BDB (BerkleyDB)	Tabelle transaction-safe con page locking
InnoDB	Tabella transaction-safe con lock di righe e chiavi esterne

Questi Engine non supportano le chiavi esterne

Questi si

- MyISAM è lo storage engine di default da MySQL 3.23. E' basato sullo storage ISAM.
- Ogni tabella MyISAM è memorizzata all'interno del disco con tre file.
- Per specificare in maniera esplicita la tipologia di tabella desiderata bisogna usare il comando ENGINE o TYPE.

```
CREATE TABLE t (i INT) ENGINE = MYISAM;  
CREATE TABLE t (i INT) TYPE = MYISAM;
```

- Alcuni storage consentono di specificare il tipo di indice (struttura dati) durante la creazione della tabella.
- Bisogna usare la parola chiave USING *type_name*.

MyISAM	BTREE
InnoDB	BTREE
MEMORY/HEAP	HASH, BTREE

Esempio:

```
CREATE TABLE lookup  
  (id INT,  
   INDEX USING BTREE (id))  
  ENGINE = MEMORY;
```

- Comandi **show - use**

```
mysql> show databases;
+-----+
| Database |
+-----+
| sampledb |
| mysql    |
| test     |
+-----+
3 rows in set (0.00 sec)

mysql> use sampledb;
Database changed
```

- **SHOW:** fornisce informazioni riguardo ai database, alle tabelle, colonne, indici, ecc. Fornisce anche informazioni riguardo il server.
- **USE:** seleziona il database corrente sul quale si intende operare.

- Si effettua l'accesso come DBA:

```
C:\mysql -u sampledba -p
```

- Si crea il proprio database;

```
mysql> CREATE DATABASE stdxxxxxdb;
mysql> use stdxxxxxdb;
```

- Si crea un account per il proprio db:

```
GRANT ALL ON stdxxxxxdb.* to
-> 'stdxxxxx'@'localhost' IDENTIFIED BY
-> 'password';
```

- Successivamente l'utente crea le tabelle che desidera:

```
CREATE TABLE studenti
    (Matricola integer primary key,
     NomeCognome char(50) not null,
     DataNascita date not null,
     LuogoNascita char(50),
     Sesso char,
     idCorsoLaurea int not null
     references CorsiLaurea(idCorsoLaurea)
    );
```

Tipo	Byte	Minimum Value (Signed/Unsigned)	Maximum Value (Signed/Unsigned)
TINYINT[(M)]	1	-128	127
		0	255
SMALLINT [(M)]	2	-32768	32767
		0	65535
MEDIUMINT [(M)]	3	-8388608	8388607
		0	16777215
INT [(M)]	4	-2147483648	2147483647
		0	4294967295
BIGINT [(M)]	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615
FLOAT[(M,D)]	4	+/-1.175494351E-38	+/- 3.402823466E+38
DOUBLE [(M,D)]	8	+/-2.2250738585072014E-308	+/-1.7976931348623157E+308

Tipo	Max size	Byte
CHAR[(M)]	M	M
VARCHAR(M)	M	L+1
TINYBLOB/ TINYTEXT	2^8-1	L+1
BLOB/TEXT	$2^{16}-1$	L+2
MEDIUMBLOB/MEDIUMTEXT	$2^{24}-1$	L+3
LOB/TEXT	$2^{32}-1$	L+4
ENUM('value1','value2',...)	65535 elementi	1 o 2 byte
SET ('value1','value2',...)	64 elementi	1,2,3,4 o 8 byte

- L rappresenta la lunghezza effettiva del testo

Tipo	Range	Byte
DATE	'1000-01-01' '9999-12-31'	3
DATETIME	'1000-01-01 00:00:00' '9999-12-31 23:59:59'	8
TIMESTAMP[(M)]	Da 19700101000000 ad una data nel 2037	4
TIME	'-838:59:59' - '838:59:59'	3
YEAR [(M)]	1901 al 2155 per YEAR(4)	1

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name.txt'
[REPLACE | IGNORE]
INTO TABLE tbl_name
[FIELDS
  [TERMINATED BY '\t']
  [[OPTIONALLY] ENCLOSED BY '' ]
  [ESCAPED BY '\\ ' ]
]
[LINES
  [STARTING BY '' ]
  [TERMINATED BY '\n']
]
[IGNORE number LINES]
[(col_name,...)]
```

- Il comando LOAD DATA INFILE consente il caricamento di una tabella ad alta velocità leggendo le righe da un file di testo.

- Caricamento dei dati:

```
mysql> LOAD DATA LOCAL INFILE 'studenti.txt'
INTO TABLE studente;
```

N.B. Se il file di testo viene generato su alcuni editor Windows potrebbe essere necessario aggiungere

LINES TERMINATED BY '\r\n';

```
mysql> LOAD DATA INFILE `studenti.txt`  
-> INTO TABLE studente LINES STARTING  
BY "yyy";
```

- Quindi un file contenente
yyy"Row" , 1
blablabla yyy"Row" , 2
- Può essere letto e verrà caricato come
("row",1), ("row",2)