

Size of MySQL database

Vogliamo sapere lo spazio occupato da ogni singolo database usando la command line, una semplice query restituisce a video l'informazione richiesta.

Prestate attenzione perché questa query potrebbe richiedere molto tempo per DB di grandi dimensioni.

```
mysql> SELECT table_schema "DB Name",  
Round(Sum(data_length + index_length) / 1024 / 1024, 1) "DB Size in MB"  
FROM information_schema.tables  
GROUP BY table_schema;
```

Ecco un esempio dell'output:

```
+-----+-----+  
| DB Name | DB Size in MB |  
+-----+-----+  
| monitoraggio | 1505.0 |  
| mysql | 0.7 |  
| pcparts | 0.4 |  
| performance_schema | 0.0 |  
| photogulp | 193.3 |  
| photogulp_webalbum | 0.5 |  
| phplistdb | 33.4 |  
| pixellone_artistika | 7.9 |  
| pixellone_enter | 0.4 |  
| rc-bazar_oc | 4.8 |  
| wordpress_9 | 1.1 |  
+-----+-----+
```

Visualizzare la struttura di un database mysql da php

Se avete bisogno di "stampare" la struttura di un database mysql esistente e non volete ricorrere a soluzioni come mysql workbench o similari, potete utilizzare un comodissimo script php di David Walsh.

Si tratta di un file .php, lo copiate sotto la DocumentRoot del vostro webserver, lo editate con db_name, user e password e lo aprite da web. Il risultato sarà del tipo:

admin_roles

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI		auto_increment
nome	varchar(255)	NO			

admin_user

Field	Type	Null	Key	Default	Extra
iduser	int(11)	NO	PRI		auto_increment
username	varchar(255)	NO			
password	varchar(255)	NO			
salt	varchar(255)	NO			
nome	varchar(50)	NO			
cognome	varchar(50)	NO			

Lo script lo potete consultare in [questo post](#) del blog di David Walsh oppure scaricarlo da [questa pagina](#) su Mr.Webmaster

MySQL UDF Perl Regular Expression

Nel realizzare nuovi scraper per [g4play.it](#) Emanuele si è reso conto che la nostra istanza MySQL non supporta le espressioni regolari, a lui non servono solo query di ricerca ma manipolazioni di dati complesse. Con estrema semplicità mi chiede di installare la libreria `lib_mysqludf_preg`, non è complicato, ma neanche così banale. Iniziamo subito con l'installazione dei pacchetti che ci serviranno:

```
[root@mysqlbit lib_mysqludf_preg]# yum install pcre pcre-devel
[root@mysqlbit lib_mysqludf_preg]# yum install make gcc gcc-c++
[root@mysqlbit lib_mysqludf_preg]# yum install mysql-devel
```

Questo per evitare tutti gli errori relativi al compilatore, a `pcre` e `mysql`. Scarichiamo il pacchetto da [GitHub](#):

```
[root@mysqlbit lib_mysqludf_preg]# wget
https://github.com/mysqludf/lib_mysqludf_preg/archive/testing.zip
```

```
[root@mysqlbit lib_mysqludf_preg]# unzip testing.zip
```

ora lanciamo il configuratore

```
[root@mysqlbit lib_mysqludf_preg]# ./configure
```

ci siamo risparmiati gli errori avendo installato preventivamente i pacchetti, l'unico messaggio a video con la parola ERROR è

```
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
```

Possiamo rilassarci, avendo settato la password di root è normale che non riesca ad accedere. Ora installiamo:

```
[root@mysqlbit lib_mysqludf_preg]# make
[root@mysqlbit lib_mysqludf_preg]# make install
[root@mysqlbit lib_mysqludf_preg]# make installdb
```

```
ERROR 1548 (HY000) at line 5: Cannot load from mysql.proc. The table is probably corrupted
make: *** [uninstalldb] Error 1
```

Sull'ultimo passaggio ho ricevuto errore di tabella corrotta, per questo ho dovuto prima "sistemare" le tabelle MySQL e poi rilanciare il make installdb

```
[root@mysqlbit lib_mysqludf_preg]# make installdb
/usr/bin/mysql -p <./uninstalldb.sql
Enter password:
cat installdb.sql | sed 's/\.so/.dll/g' >installdb_win.sql
if test -f .libs/lib_mysqludf_preg.dll; then \
    /usr/bin/mysql -p <./installdb_win.sql; \
else \
    /usr/bin/mysql -p <./installdb.sql;\
fi
Enter password:
[root@mysqlbit lib_mysqludf_preg]# make test
cd test; make test
make[1]: Entering directory `/usr/local/lib/lib_mysqludf_preg/test'
/usr/bin/mysqltest -p --include=create_testdb.sql --result-f...
Enter password:
ok
/usr/bin/mysqltest -p --include=create_testdb.sql --result-f...
Enter password:
```

```
ok
/usr/bin/mysqltest -p --include=create_testdb.sql --result-f...
Enter password:
ok
/usr/bin/mysqltest -p --include=create_testdb.sql --result-fi...
Enter password:
ok
/usr/bin/mysqltest -p --include=create_testdb.sql --result-f...
Enter password:
ok
/usr/bin/mysqltest -p --include=create_testdb.sql --result-f...
Enter password:
ok
make[1]: Leaving directory `/usr/local/lib/lib_mysqludf_preg/test'
```

Finito, ora Emanuele potrà usare le espressioni regolari per manipolare i dati di g4play.it.

PRIMO POST

Dopo aver maturato esperienza di lunga data spaziando nei vari settori informatici, il team LBiT ha deciso di aprire un blog per condividere questo bagaglio con tutti voi.

Ci sembra la cosa più giusta da fare trovandoci in team dislocati, poiché la materia informatica continua ad essere la nostra passione e in questo modo potremo documentare le attività che svolgiamo su quello che ci piace chiamare “un diario di appunti”.

**Open
Source Code**



Come tutti gli informatici che seguono la filosofia dell'Opensource, crediamo che la condivisione sia alla base della crescita di ogni individuo. Non siamo gelosi del nostro lavoro, non abbiamo paura di mettere a disposizione il nostro sapere. Altri prima di noi hanno condiviso l'esperienza maturata sul campo offrendoci guide, sviluppando software e sistemi operativi dandoci

l'opportunità di accedere alla materia fino a diventare dei professionisti, e grazie a loro siamo qui a fare altrettanto oggi.

L'impresa LBit si propone di affiancare e supportare il cliente organizzando al meglio la sua attività con l'utilizzo e l'ottimizzazione degli strumenti informatici
