



MySQL™ Connect



Getting Started with MySQL

- Santo Leto
Principal Technical Support Engineer,
MySQL
- Jesper Wisborg Krogh
Principal Technical Support Engineer,
MySQL



Program Agenda

- MySQL Overview
- Users and Security
- Datatypes and Queries
- Data Import/Export and Backups
- Installation
- MySQL Workbench (Alfredo Kojima)

The MySQL in a Nutshell



The MySQL Homepages

- <http://www.mysql.com/> - Product information, events, etc.
- <http://dev.mysql.com/> - The developer zone.
- <http://bugs.mysql.com/> - The bugs database.

MySQL Features

- Supports a wide range of platforms
<http://dev.mysql.com/doc/refman/5.6/en/supported-os.html>
- Multithreaded
- Support for server side procedures, functions, events, and triggers
- Supports multiple storage engines
- Localization
 - Language for error messages
 - Character sets
 - Collations

MySQL Features (continued)

- Can be extended through plugins
 - Storage Engine plugins
 - Full-Text Parser plugins
 - Daemon plugins
 - INFORMATION_SCHEMA plugins
 - Semisynchronous Replication plugins
 - Audit plugins
 - Authentication plugins
 - Password-Validation plugins

Commercial Extensions

- MySQL Enterprise Monitor
- MySQL Enterprise Backup
- External authentication
- Thread pool plugin
- Audit plugin
- High-Availability (OVM template + Windows Clustering)
- For MySQL Cluster: Cluster Manager
- See also: <http://www.mysql.com/products/>

Installing the MySQL Server



Install MySQL on Linux as a Normal User

- `tar -zxf mysql-5.6.6-m9-linux2.6-x86_64.tar.gz`
- `ln -s mysql-5.6.6-m9-linux2.6-x86_64 mysql`
- `./mysql/scripts/mysql_install_db --no-defaults --basedir=$(pwd)/mysql`
- `./mysql/bin/mysqld --no-defaults --port=4406 --socket=$(pwd)/data/mysql.sock --console --basedir=$(pwd)/mysql --datadir=$(pwd)/data &`
- `./mysql/bin/mysqladmin --no-defaults --socket=$(pwd)/data/mysql.sock --user=root shutdown`

Demo: New MySQL Installer for Windows



Directory Layout

- basedir
- plugindir
- datadir
- InnoDB files
- Logs

MySQL Configuration

- The Reference Manual is your best friend:
<http://dev.mysql.com/doc/refman/5.6/en/server-system-variables.html>
- Naming convention for the configuration file:
 - Windows: config.ini
 - Linux/Unix: my.cnf

MySQL Configuration (continued)

- Create my.cnf in /home/ouser/mysqlconnect

```
[mysqld]
basedir          = /home/ouser/mysqlconnect/mysql
datadir          = /home/ouser/mysqlconnect/data
log_bin         = /home/ouser/mysqlconnect/logs/binary-logs
general_log_file = /home/ouser/mysqlconnect/logs/general.log
slow_query_log_file = /home/ouser/mysqlconnect/logs/slow.log
log_error       = /home/ouser/mysqlconnect/logs/error.log

socket          = /home/ouser/mysqlconnect/data/mysql.sock
port           = 4406

[mysqld_safe]
ledir          = /home/ouser/mysqlconnect/mysql/bin
```

- mkdir logs

mysqld_safe

- Recommended way to start MySQL on Linux/Unix.
- Restart MySQL automatically when an error occurs.
- Start MySQL with:
`./mysql/bin/mysqld_safe --defaults-file=my.cnf &`

MySQL Configuration (continued)

- Log into MySQL:
`./mysql/bin/mysql --socket=data/mysql.sock --user=root --password`
- Check the configuration from inside MySQL:

```
mysql> SHOW VARIABLES;  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| auto_increment_increment | 1 |  
| wait_timeout | 28800 |  
...  
| warning_count | 0 |  
+-----+-----+  
424 rows in set (0.07 sec)
```


Session and Global Variables

- Global variables are server wide.
- Session variables are for that one connection.
- Where a variable can both be a session and a global variable, the global variable is used as the default for the session variable.
- Changing a global variable only affects new connections unless it is for an explicit global feature.

Session and Global Variables (continued)

```
mysql> SELECT @@global.long_query_time, @@session.long_query_time;
+-----+-----+
| @@global.long_query_time | @@session.long_query_time |
+-----+-----+
|          10.000000 |          10.000000 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> SET GLOBAL long_query_time = 1;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SELECT @@global.long_query_time, @@session.long_query_time;
+-----+-----+
| @@global.long_query_time | @@session.long_query_time |
+-----+-----+
|          1.000000 |          10.000000 |
+-----+-----+
1 row in set (0.01 sec)
```

Session and Global Variables (continued)

```
mysql> connect
Connection id:      3
Current database:  *** NONE ***

mysql> SELECT @@global.long_query_time, @@session.long_query_time;
+-----+-----+
| @@global.long_query_time | @@session.long_query_time |
+-----+-----+
|                1.000000 |                1.000000 |
+-----+-----+
1 row in set (0.00 sec)
```

Session and Global Variables (continued)

```
mysql> SET GLOBAL general_log = ON;  
mysql> SELECT 1;  
  
shell$ cat logs/general.log
```

The World Database

```
mysql> CREATE DATABASE world;  
Query OK, 1 row affected (0.03 sec)
```

```
shell$ ./mysql/bin/mysql --socket=data/mysql.sock --user=root --password  
world < world_innodb.sql
```

```
-- While the import is running  
mysql> SHOW PROCESSLIST;
```

```
-- When the import is done  
mysql> use world;  
mysql> SHOW TABLES;  
mysql> SHOW CREATE TABLE city\G
```

Query Terminator

- A query is terminated by a delimiter, default is ; \g \G
- ; and \g are synonyms
- The difference between ; and \G is that the latter creates a vertical output

```
mysql> SELECT * FROM mysql.user LIMIT 1;  
mysql> SELECT * FROM mysql.user LIMIT 1\G
```

Limit

- The LIMIT modifier can be used to limit the number of rows in the result set.
- An optional offset can be specified.

```
mysql> SELECT * FROM City ORDER BY Population DESC LIMIT 10;  
mysql> SELECT * FROM City ORDER BY Population DESC LIMIT 10,5;
```

EXPLAIN

- Adding EXPLAIN in front of a query tells how the MySQL will execute the query.
- Add the FULL keyword and extra information will be available.
- In 5.5 and earlier only worked with SELECT.
- In 5.6 support for INSERT, UPDATE, DELETE as well.

EXPLAIN

```
mysql> EXPLAIN SELECT * FROM City WHERE CountryCode = 'USA';  
mysql> EXPLAIN EXTENDED SELECT * FROM City WHERE CountryCode =  
'USA';  
mysql> SHOW WARNINGS;  
Mysql> EXPLAIN UPDATE City SET Population = Population * 1.1  
WHERE CountryCode = 'USA';
```

Identifier Quoting

- Sometimes it is not obvious whether a word in a query refers to a database object or is a keyword.
- Will default to keyword.
- Quote the name of the database object with backticks (`) to tell MySQL it is not a keyword.

```
mysql> CREATE TABLE int (i INT);  
mysql> CREATE TABLE `int` (i INT);
```

- Just because it's possible to use reserved words as identifiers, doesn't mean it is a good thing!

TRUNCATE

- Truncate resets a table as if it had just been created.
- Effectively a DROP TABLE followed by a CREATE TABLE.
- Beware that it counts as a DDL statement.

```
mysql> CREATE TABLE t (i INT);  
mysql> INSERT INTO t VALUES (1);  
mysql> SELECT * FROM t;  
mysql> TRUNCATE t;  
mysql> SELECT * FROM t;
```

Temporal Data Types

- DATE, DATETIME, TIME, TIMESTAMP
- TIMESTAMP is timezone aware
- DATE, DATETIME, TIME are not timezone aware
- Can set to use current time automatically (before 5.6 only for TIMESTAMP).
- MySQL 5.6 adds microsecond support.

Temporal Data Types (continued)

```
mysql> CREATE TABLE v (  
->   a INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
->   b INT,  
->   created TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP(6),  
->   updated TIMESTAMP(6) ON UPDATE CURRENT_TIMESTAMP(6)  
-> );  
  
mysql> INSERT INTO v (b) VALUES (1);  
mysql> SELECT * FROM v;  
mysql> UPDATE v SET b = b+1;  
mysql> SELECT * FROM v;
```

Information Schema

- Provides access to “static” information about the database.
- All users have access to the Information Schema, but can only see what their privileges allow.

Information Schema

- See who has privileges to shutdown MySQL:

```
mysql> SELECT grantee FROM information_schema.USER_PRIVILEGES  
WHERE privilege_type = 'SHUTDOWN';
```

- Show me all the TIMESTAMP columns with their properties:

```
mysql> SELECT table_schema, table_name, column_name,  
column_default, extra FROM information_schema.COLUMNS where  
data_type = 'timestamp'\G
```

- Some information schema queries can be very slow and cause performance issues - use with caution in production.

The Query Cache

- Stores the raw query text with the result set.
- If a new query matches exactly, returns the result set from the cache.
- Completely transparent.
- Single threaded, so often a bottleneck for high concurrency cases.
- Keep small or turn off.

The Query Cache (continued)

```
mysql> SHOW VARIABLES LIKE 'query_cache%';

mysql> CREATE TABLE City1 LIKE City;
mysql> INSERT INTO City1 (Name, CountryCode, District,
Population) SELECT Name, CountryCode, District, Population FROM
City;

-- Run 6 times:
mysql> INSERT INTO City1 (Name, CountryCode, District,
Population) SELECT Name, CountryCode, District, Population FROM
City1;

mysql> SELECT * FROM City1 ORDER BY Population LIMIT 5;
mysql> SELECT * FROM City1 ORDER BY Population LIMIT 5;

mysql> SET GLOBAL query_cache_size = 16*1024*1024;
mysql> SHOW VARIABLES LIKE 'query_cache%';

mysql> SELECT * FROM City1 ORDER BY Population LIMIT 5;
mysql> SELECT * FROM City1 ORDER BY Population LIMIT 5;
```

Common Connect Problems

- Network access (firewalls, etc.). What does this look like? like the server isn't there:

```
shell$ mysql -uroot -P4407 --protocol=tcp
```

- How can you quickly test access? Use telnet:

```
shell$ telnet localhost 4407  
shell$ telnet localhost 4406
```

Common Connect Problems (continued)

- Slow DNS - used both client-side and server-side for reverse DNS lookups for authentication. If DNS is slow or flaky, consider using only IP addresses in account definitions and `--skip-name-resolve`. References: <http://dev.mysql.com/doc/refman/5.6/en/host-cache.html>
- Account verifies as `user@host` - Wildcards can give unexpected behaviour.

```
mysql> SELECT User, Host FROM mysql.user;  
mysql> SELECT USER(), CURRENT_USER();
```

SQL Modes

- The `sql_mode` option can be used to define how MySQL behaves.

<http://dev.mysql.com/doc/refman/5.6/en/server-sql-mode.html>

```
mysql> SET SESSION sql_mode = '';  
mysql> SELECT 0 || 1;  
mysql> SET SESSION sql_mode = 'PIPES_AS_CONCAT';  
mysql> SELECT 0 || 1;
```

```
mysql> SET SESSION sql_mode = '';  
mysql> SELECT "Name" FROM "City" LIMIT 1;  
mysql> SET SESSION sql_mode = 'ANSI';  
mysql> SELECT "Name" FROM "City" LIMIT 1;
```

SQL Modes (continued)

```
mysql> SET SQL_MODE = '';  
mysql> CREATE TABLE sm (a TINYINT) ENGINE = InnoDB;  
mysql> INSERT INTO sm VALUES ( 1234567890 );  
mysql> SHOW WARNINGS;  
mysql> SELECT * FROM sm;  
mysql> SET SQL_MODE = 'STRICT_ALL_TABLES';  
mysql> INSERT INTO sm VALUES ( 1234567890 );
```

```
mysql> SET SESSION sql_mode = '';  
mysql> SELECT "Name" FROM "City" LIMIT 1;  
mysql> SET SESSION sql_mode = 'ANSI';  
mysql> SELECT "Name" FROM "City" LIMIT 1;
```

Securing MySQL

- Default users not very secure:

```
mysql> SELECT User, Host, Password FROM mysql.user;
```

- Run the `mysql_secure_installation` script
- To set password manually:

```
mysql> SET PASSWORD FOR ''@localhost = PASSWORD('password');  
mysql> SELECT User, Host, Password FROM mysql.user;
```

Securing MySQL (continued)

- When creating a new user:

```
mysql> GRANT USAGE ON *.* TO newuser@localhost;
```

- Now user has no password!!!
- Safer way:

```
mysql> SET SESSION sql_mode = 'NO_AUTO_CREATE_USER';  
mysql> DROP USER newuser@localhost;  
mysql> GRANT USAGE ON *.* TO newuser@localhost;  
mysql> GRANT USAGE ON *.* TO newuser@localhost IDENTIFIED BY  
'password';
```

Securing MySQL (continued)

- MySQL 5.6 has password validation
<http://dev.mysql.com/doc/refman/5.6/en/validate-password-p>
- Update my.cnf:

```
[mysqld]  
plugin-load=validate_password.so
```

- Restart MySQL and:

```
mysql> SHOW VARIABLES LIKE '%password%';  
mysql> DROP USER newuser@localhost;  
mysql> GRANT USAGE ON *.* TO newuser@localhost IDENTIFIED BY  
'password';
```


Demo: MySQL Enterprise Monitor



Demo: MySQL Enterprise Backup





MySQL™ Connect