MySQL for Oracle DBA's

Oded Raz
Brillix
Israel

Keywords:

Mysql, Oracle, DBA, MySQL samples, Oracle DBA

Introduction

MySQL community is rapidly growing, and a lot of non-start-up companies start using MySQL as an alternative to Oracle database. Those companies have their own DBA's usually Oracle / MS SQL that suddenly asked to manage MySQL databases as well. If you are experienced Oracle DBA, don’t worry, MySQL administration is just a walk in the park comparing Oracle administration, all you need to know are the differences in the architecture between Oracle and MySQL and some tips & tricks and are set to go as a MySQL DBA.

MySQL Architecture

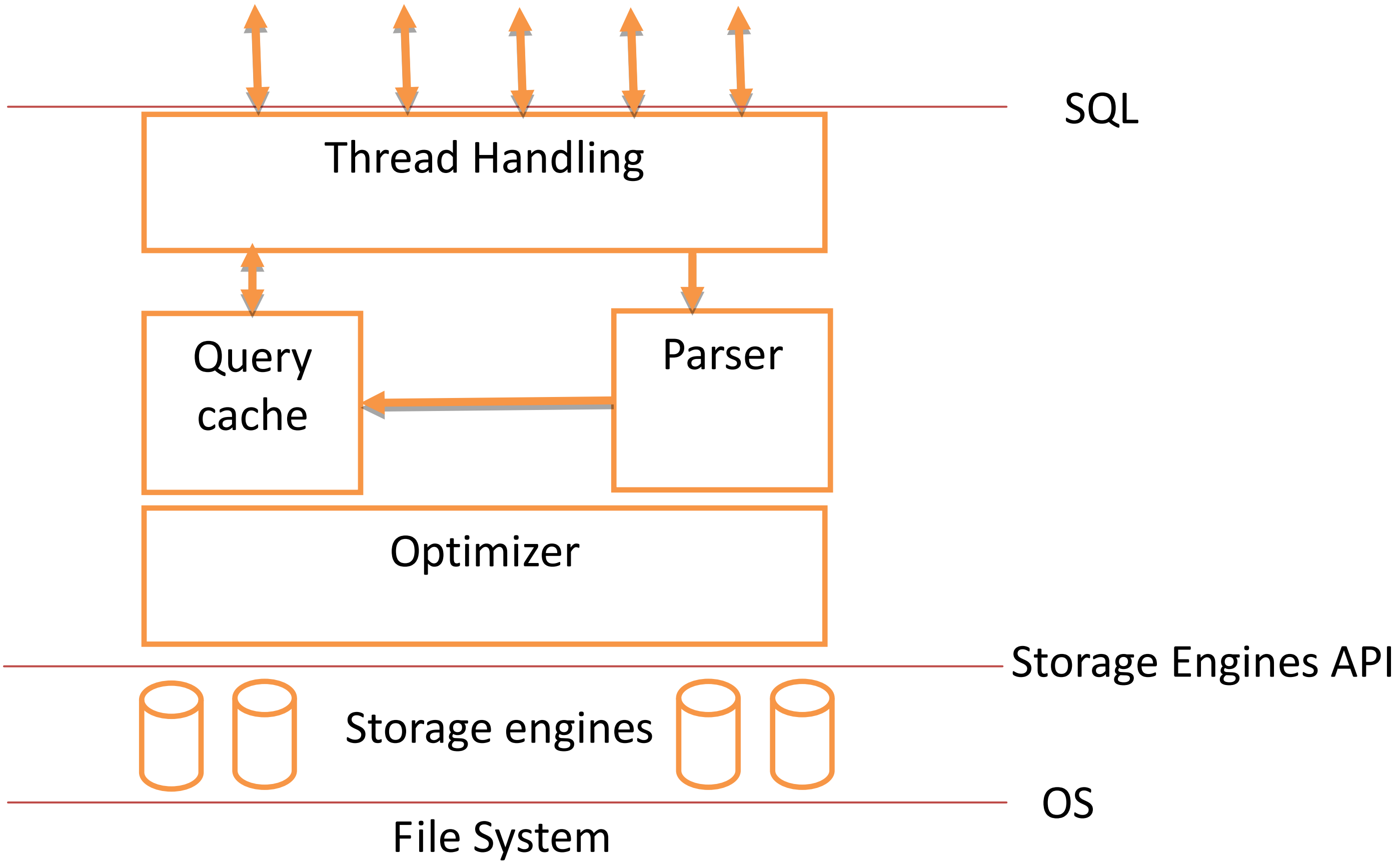

Illustration. 1: MySQL Architecture

MySQL Instance Architecture

Instance

MySQL installation supports many instances under one MySQL installation. In order to add a new instance, just add a new server-id definition in "my.cnf" configuration file:

```
SERVER-ID=2
```

MySQL instance doesn't have unique name instead the uniqueness of the server is the server-id and the port number.

Database

MySQL instance can hold up to 32,000 databases. All of the data components like, tables, indexes, triggers and cet… are held inside the database. Database can be compared to Oracle Schema BUT be aware that in oracle schema is also a user and in MySQL it's only a container of the data objects and doesn't represent any user object.

Users

User are instance level objects and not database level objects. MySQL user name is has two elements, the username and from where he allowed to connect:

```
mysql> CREATE USER 'demo'@'localhost' IDENTIFIED BY 'some_pass';
```

The user above name is demo and he is only allowed to connect to our instance from the localhost.

```
mysql> CREATE USER 'demo1'@'%' IDENTIFIED BY 'some_pass';
```

The user above name is demo1 and he can connect to our instance from anywhere in the world.

After we create the user we should give him the permission to access the databases in our instance:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'demo'@'localhost'
    ->     WITH GRANT OPTION;

mysql> GRANT ALL PRIVILEGES ON db1.* TO 'demo1'@'%'
    ->     WITH GRANT OPTION;
```

We gave demo@localhost a permission to handle all of the objects in all of the database in our instance, to demo1@% we have given the permissions to handle all of the object inside db1 database only.

Finally you can list all of the instance users:

```
mysql> SELECT * FROM mysql.user;
```

Tablespace

Currently there is no Oracle equivalent to tablespaces in MySQL. In MySQL 5.7 tablespaces will be introduced.

Storage Engine

Storage engines are MySQL components that handle the SQL operations for different table types. MySQL storage engines include both those that handle transaction-safe tables and those that handle nontransaction-safe tables. MySQL uses a pluggable storage engine architecture that enables storage engines to be loaded into and unloaded from a running MySQL server.

To determine which storage engines your server supports, use the SHOW ENGINES statement.

There are 2 main storage engines:

- **InnoDB:** The default storage engine as of MySQL 5.5.5. InnoDB is a transaction-safe storage engine for MySQL that has commit, rollback, and crash-recovery capabilities to protect user data. InnoDB row-level locking (without escalation to coarser granularity locks) and Oracle-style consistent nonlocking reads increase multi-user concurrency and performance. InnoDB stores user data in clustered indexes to reduce I/O for common queries based on primary keys. To maintain data integrity, InnoDB also supports FOREIGN KEYreferential-integrity constraints. For more information about InnoDB, see Chapter 14, The InnoDB Storage Engine.
- **MyISAM**: The MySQL storage engine that is used the most in Web, data warehousing, and other application environments. MyISAM is supported in all MySQL configurations, and is the default storage engine prior to MySQL 5.5.5.

See following URL for more MySQL storage engines:
http://dev.mysql.com/doc/refman/5.5/en/storage-engines.html

Connecting to the database

Before you can connect to MySQL database you have to login to the sever hosting MySQL, typically via ssh. Type the following to login:

```
mysql -u root –p
```

Now you are connected to the MySQL client and you will see a prompt like this:

```
mysql>
```

Next you can list the available databases inside the instance and connect to one of the databases:

```
mysql> show databases; -- list database inside the instance

mysql> use <databases_name>; -- connect to a specific database
```

Query MySQL objects structure

MySQL allow us to query it's metadata in several ways:

Tables

```
mysql> show tables; -- list all tables in database

mysql> show table status; -- list table data including storage engine

mysql> show create table <table_name>; -- list table DDL
```

Indexes

```
mysql> show index from <table name>; -- list all index defined on a table

mysql> show index from <db name>; -- list all index in a database
```

Columns

```
mysql> SHOW COLUMNS FROM City; -- show columns of a table
+------------+----------+------+-----+---------+----------------+
| Field      | Type     | Null | Key | Default | Extra          |
+------------+----------+------+-----+---------+----------------+
| Id         | int(11)  | NO   | PRI | NULL    | auto_increment |
| Name       | char(35) | NO   |     |         |                |
| Country    | char(3)  | NO   | UNI |         |                |
| District   | char(20) | YES  | MUL |         |                |
| Population | int(11)  | NO   |     | 0       |                |
+------------+----------+------+-----+---------+----------------+
```

Show command

Show command is MySQL DBA best friend. See some sample of useful show commands:

```
SHOW CHARACTER SET
SHOW COLLATION
SHOW COLUMNS
SHOW DATABASES
SHOW FUNCTION STATUS
SHOW INDEX
SHOW OPEN TABLES
SHOW PROCEDURE STATUS
SHOW STATUS
SHOW TABLE STATUS
SHOW TABLES
SHOW VARIABLES
```

INFORMATION_SCHEMA

Starting MySQL 5.0 the information_schema database has been available, containing data dictionary information. To see all the tables (actually views) contained in the data dictionary, within the mysql client:

```
mysql> SELECT table_name, table_type, engine
    -> FROM information_schema.tables
    -> WHERE table_schema = 'db5'
    -> ORDER BY table_name DESC;
+------------+------------+--------+
| table_name | table_type | engine |
+------------+------------+--------+
| v56        | VIEW       | NULL   |
| v          | VIEW       | NULL   |
| tables     | BASE TABLE | MyISAM |
| t2         | BASE TABLE | MyISAM |
| fk2        | BASE TABLE | InnoDB |
| fk         | BASE TABLE | InnoDB |
+------------+------------+--------+
```

You can read more about information schema here -
http://dev.mysql.com/doc/refman/5.0/en/information-schema.html

Summery

As an Oracle DBA I have been intimidated by MySQL, but after working with MySQL a bit more I was impressed with MySQL's features, reliability and ease-of-use. I hope I have given you a starter tool kit and the confidence to jump in there and have a look at those MySQL servers that are sitting in the corner of your server rooms. Know it all up to you to download and install MySQL and have a go with it. And now that MySQL is owned by Oracle you even have the right excuse for your boss.

Contact address:

Oded Raz
Brillix, DBAces
Arie Dolchine 6
69360, Tel-Aviv, Israel

Phone:	+972(54)54-4746926
Fax:	+972(54)3-7414034
Email	oded@dbaces.com
Internet:	www.dbaces.com