MySQL Idiosyncrasies that BITE

Ronald Bradford
42SQL
New York, NY

Keywords:

MySQL, compatibility, data integrity, transactions, ACID, SQL_MODE

Introduction

While MySQL is a popular and widely used RDBMS, some default features and settings are very foreign in comparison with other commercial RDBMS products. In this discussion we will discuss some of the MySQL defaults including a non-transactional state, silent data truncations, date management and transaction isolation options. These are all critical for data integrity and consistency. We cover in-depth topics including SQL_MODE that saves the day. We will also cover character sets and collations and the best practices to ensure your UTF8 is stored and retrieved correctly.

MySQL != Oracle

Oracle is a relational database, MySQL is a relational database. Why then is Oracle not equal to, or the same as MySQL and visa versa?

Oracle is an older product, starting in 1977 (33 years), compared with 1995 (15 years) for MySQL. Oracle is a commercial product with a greater research and development budget and physical resources while MySQL is an open source product. Oracle can spend more in one year in development, then the total cost of the sale of MySQL. The philosophy however is not like a true open source project with the ownership and commit capabilities always being controlled by the trademark owner which is now Oracle. The target audience is vastly different. While Oracle has a focus and firm hold on large commercial enterprises, MySQL is king of the website, internet startup or large web 2.0 site such as Facebook, You Tube, Twitter, Yahoo, Wikipedia, Craigslist to name a few.

The MySQL product also has a very different practice for development. Features, worklog and bugs are all open and public information, and the products generally move from an alpha,beta,release candidate and Generally Availability. For MySQL, most work is secret until released. While it's more likely that most MySQL installations have a version of MySQL this is newer then 2 years ago, for most Oracle installations, 11g which was released 2 years ago is not yet in use. This is a combination of factors including application complexity, long development project life cycles and risk of upgrading.

While both products are different, there are the characteristics you most expect in a relational database product irrespective of the brand of database?

MySQL includes the characteristics you expect for a relational database including data integrity, transactions, ACID compliance etc, however the problem is a number of these are not enabled by default.

This is not to say the MySQL is a product you can't use, millions of installations including most of the top websites worldwide successfully use MySQL even with various limitations.

A different state of SQL error

In MySQL there is the concept of a Warning. Often overlooked as you expect an error, this feature can be disturbing, especially if your not expecting this. This is no more clearly represented then for a difference in data in (INSERT) and data out (SELECT).

```
mysql> INSERT INTO orders (qty) VALUES (-1), (9000);
mysql> SELECT * FROM orders;
+-----+---------------------+
| qty | created             |
+-----+---------------------+
|   0 | 2010-10-25 12:23:06 |
| 255 | 2010-10-25 12:23:06 |
+-----+---------------------+
```

This is due to the datatype of TINYINT UNSIGNED which is a 1 byte column, and therefore can only hold 2^8 or 256 unique values 0-255. This is also representative for strings and for dates. In MySQL this is known as silent truncations, where MySQL is making a decision to take its best guess.

If you want MySQL to operate and perform data integrity closer to your interpretation of good integrity you need to set the SQL_MODE. The minimum best practice is:

```
SET GLOBAL SQL_MODE=‘STRICT_ALL_TABLES,NO_ZERO_ON_DATE,NO_ZERO_DATE’
```

Transactional and non transactional

The most foreign concept for Oracle resources is the understanding of MySQL storage engines. These engines change the way information can be stored and retrieved and can have impacts on performance, locking and most importantly transactional state. In MySQL it is possible to have non-transactional engines. This greatly affects the understanding of ACID compliance and the expectation of transactions.

Atomicity and consistency are attributes of transactional engines, not non-transactional engines and you may experience different results then you would expect.

Conclusion

These two examples are just an introduction to what is necessary in understanding and appreciating MySQL, In this presentation we will also discuss a number of other important characteristics you should be fully prepared for when using MySQL. You should never assume that features, syntax and functionality of one relational database translates in a different database.

Contact address:

Ronald Bradford
42SQL
One Penn Plz #6139
New York NY 10119 USA

Email me@ronaldbradford.com
Internet: http://ronaldbradford.com