Configuring MySQL for full Unicode support

Revision 20200604

Denodo Inc.: 525 University Avenue, Suite 31, Palo Alto, CA 94301. USA www.denodo.com

Goal

This document explains the limitations of MySQL and other MySQL database implementations like Aurora in the support of UTF-8. Taking into account these limitations, the steps to enable full support of the Unicode charset in MySQL are listed. Within the context of Denodo this configuration is very important when using MySQL as a cache database as the data being cached and coming from different data sources can include any number of Unicode characters.

Content

The current version of the Unicode charset has 1,114,112 code points (last code point is U+10FFFF). In its version 1.0 Unicode was limited to 65,536 code points in the range from U+0000 to U+FFFF called BMP (Basic Multilingual Plane).

MySQL's utf8 charset only implements proper UTF-8 encoding partially. It has support for the BMP characters only as it can only store a maximum of three bytes per multibyte character. UTF-8-encoded symbols that take up four bytes are not supported.

Therefore, it is not possible to store any symbol whose code point ranges from U+010000 to U+10FFFF (needs four bytes in UTF-8) using MySQL's utf8 implementation. This affects a big number of symbols (only 5.88% of all possible Unicode symbols can be stored in MySQL utf8) including most of the emojis and the Mathematical Alphanumeric Symbols.

Starting from MySQL 5.5.3 a new encoding called utf8mb4 which fully supports Unicode is introduced. Therefore, the solution to fully support utf8 is to change the encoding from utf8 to utf8mb4 in the MySQL database and the tables involved.

To do this, follow these steps:

1.- Change the character set and collation properties of a database by running the following statement:

```
ALTER DATABASE cache_database_name
CHARACTER SET = utf8mb4
COLLATE = utf8mb4_unicode_ci;
```

2.- Modify the MySQL `my.ini` file (for Linux: `/etc/my.cnf`) to include/replace the following properties:

```
[client]
default-character-set = utf8mb4
[mysql]
default-character-set = utf8mb4
[mysqld]
init-connect='SET NAMES utf8mb4'
collation_server=utf8mb4_unicode_ci
```

```
character_set_server=utf8mb4
skip-character-set-client-handshake
```

Note: Starting from MySQL 8.0.x, most of the above properties are configured by default to the above values. We recommend checking your MySQL properties.

After following these steps, the MySQL database will be able to support every unicode character. Once the new character set is configured MySQL can be used as a cache database without risking the loss of information when the data cached in Denodo contains Unicode characters.

Finally, note that using utf8mb4 the maximum number of characters that can be stored in the columns and index keys is going to be affected. In this case, it will be possible to store fewer characters as MySQL is using one additional byte to represent each character, from 3 to 4 bytes.

These are the maximum lengths of the text types in a MySQL database:

Type	**Maximum length**
TINYTEXT	255 (2^8−1) bytes
TEXT	65,535 (2^{16}−1) bytes = 64 KiB
MEDIUMTEXT	16,777,215 (2^{24}−1) bytes = 16 MiB
MEDIUMTEXT	4,294,967,295 (2^{32}−1) bytes = 4 GiB

For instance, a TINYTEXT can store in up to 85 three-byte characters with utf8 whereas with utf8mb4 63 four-byte characters is the limit. This should be taken into in order to decide if the type of a column has to be changed.

References

The utf8 Character Set (3-Byte UTF-8 Unicode Encoding)
How to support full Unicode in MySQL databases
Unicode
New Defaults in MySQL 8