

Character set và collation trong mysql

Jul 15, 2013 • hungneox

Character set = characters + encoding method

Một character set là một tập hợp các ký tự và các phương thức chuyển mã ký tự (encoding). Còn một collation là một tập hợp các qui tắc để so sánh hai ký tự trong một tập hợp ký tự. Giả sử ta có 4 chữ cái A, a, B, b được encode là A = 65, B = 66, a = 97, c = 98. Thì trong máy tính, chữ 'A' là một ký hiệu và 65 là mã của được chuyển của 'A'. Thì sự kết hợp của một tập các ký tự và cách chuyển mã của chúng được gọi là `character set`.

Bạn có thể thấy mỗi ngôn ngữ khác nhau sẽ có những tập hợp các ký tự khác nhau, cùng là bản chữ cái latin nhưng tiếng Việt lại có thêm nhiều ký tự mà các ngôn ngữ dùng chữ latin khác không có chẳng hạn như chữ Æ, Å, Ê, Ô, ư, Đ. Và có rất nhiều kiểu chuyển mã (encode) tiếng Việt khác nhau ngoài Unicode ra như `TCVN3`, `VNI` hay `VISCII`. Do vậy không thể dùng font VNI mà gõ theo kiểu gõ Unicode được, vì bộ gõ sẽ ánh xạ sai mã và ký tự tương ứng.

Trong MySQL thì ta có thể lưu trữ dữ liệu ở nhiều dạng character set khác nhau ở các mức độ khác nhau như server, database, table và column. Mỗi character set có một collation mặc định của nó. Ví dụ trong MySQL, character set `latin1` (West European Character Sets) là character set mặc định và collation `latin1_swedish_ci` là collation mặc định của nó. `collation` là qui tắc sắp xếp của một character set, ví dụ trong tiến Phần Lan và Thụy Điển thì các chữ Å, Ä, Ö nằm cuối cùng trong bảng chữ cái sau cả Z, ta có thể thấy chữ gần giống chữ A chưa hẳn là nằm ngay sau A.

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Y, Z, Å, Ä, Ö

Theo qui tắc đặt tên collation của mysql, thì `_ci` có nghĩa là `case insensitive` do đó `latin1_swedish_ci` là một collation không phân biệt hoa thường tương đương với `CP1252` trong Windows. Và vì trong character set Latin1 chỉ chứa các ký tự của các ngôn ngữ Tây Âu, do vậy không thể sử dụng Latin1 để lưu trữ tiếng Việt được.

MySQL nguyên thủy được tạo ra bởi MySQL AB¹, một công ty Thụy Điển. Cho nên cũng không khó hiểu khi MySQL có default collation là `latin1_swedish_ci`.

Trong MySQL, có 2 dạng dữ liệu kiểu string đó là nonbinary string (`char` , `varchar` , `text`) và binary string (`binary` , `varbinary` , `blob`). Thì chỉ có kiểu non-binary string mới dùng collation khi sắp xếp và tìm kiếm. Còn kiểu binary string, sử dụng mã tương ứng để so sánh và tìm kiếm cho nên nó phân biệt hoa thường (`case sensitive`)

Nên dùng `utf8_general_ci` hay `utf8_unicode_ci`?

Theo tài liệu của MySQL thì các thao tác sử dụng `_general_ci` thì nhanh hơn các thao tác sử dụng collation `_unicode_ci`. Vì nó hỗ trợ ánh xạ mở rộng một số ký tự thành một tổ hợp các ký tự khác, ví dụ như chữ ß trong tiếng Đức có thể so sánh tương đương với sự kết hợp của 2 chữ `ss`.

Ví dụ: ta có 1 trong bảng `dictionary(id, word)` có 2 dòng như sau:

1. `hassen`
2. `heißen`

Khi ta thực hiện câu query sau, thì chỉ có 1 kết quả `heißen` trả về

```
SELECT * FROM `dictionary` WHERE word= 'heißen' COLLATE utf8_general_ci;
```

Tuy nhiên, nếu ta chỉ định `COLLATE` là `utf8_unicode_ci`

```
SELECT * FROM `dictionary` WHERE word= 'heißen' COLLATE utf8_unicode_ci;
```

thì có cả hai kết quả trên trả về, ta thấy ở `utf8_unicode_ci` thì chữ ß được xem tương đương với 2 chữ ss. Xem thêm [Examples of the Effect of Collation](#)

Ngoài ra nó còn hỗ trợ các cách viết rút gọn (contraction) và các ký tự có thể bỏ qua (ignorable character). Theo một số câu trả lời trên stackoverflow thì `utf8_general_ci` xem A, Ä, Â, a, ä, â đều tương đương với A khi sắp xếp và tìm kiếm, nhưng điều này cũng tương tự với `utf8_unicode_ci`.

Ignorable character là những ký tự không nhìn thấy được nhưng có tác dụng trong định dạng văn bản. Ví dụ như soft hyphen là một ký tự tương tự dấu "-" nhưng không nhìn thấy được và nó được sử dụng khi một từ quá dài phải xuống dòng.

Xem thêm [Soft Hyphen – A New URL Obfuscation Technique](#)

Unicode dựng sẵn hay tổ hợp²



Phần này không liên quan tới MySQL, nhưng cũng là một điểm nên lưu ý với người lập trình. Đối với unicode dựng sẵn, mỗi ký tự ta nhìn thấy chỉ bao gồm một mã duy nhất (ví dụ á là `U+00E1` hay 255 theo hệ thập phân), Tuy nhiên đối với unicode tổ hợp, thì một ký tự ta nhìn thấy có thể là một "tổ hợp" của một ký tự chính và các dấu ví dụ chữ á bao gồm chữ a và dấu sắc (`U+0061` theo sau bởi `U+0301`). Đối với người dùng thì tổ hợp hay dựng sẵn cũng không có ảnh hưởng lắm. Nó ảnh hưởng tới độ dài của chuỗi khi đếm để cắt chuỗi hoặc xử lý. Và unicode tổ hợp có thể gây những lỗi khó hiểu đối với chương trình của bạn. Do đó bạn cần chuyển chuỗi đầu vào qua unicode dựng sẵn hoặc xử lý chuẩn hóa chuỗi đó thêm, ví dụ lược bỏ các dấu.

Ví dụ Đối với unicode dựng sẵn Nguyễn có độ dài là 6 ký tự. Còn Unicode tổ hợp thì Nguyễn có độ dài là 7 do ẽ là 2 ký tự e + (dấu mũ và dấu ngã tính là một).

References:

1. "Why is MySQL's default collation latin1_swedish_ci? - Stack Overflow." 2011. 15 Jul. 2013 <http://stackoverflow.com/questions/6769901/why-is-mysqls-default-collation-latin1-swedish-ci>
2. "mysql - What's the difference between utf8_general_ci and ..." 2010. 15 Jul. 201 <http://stackoverflow.com/questions/766809/whats-the-difference-between-utf8-general-ci-and-utf8-unicode-ci>
3. "MySQL Character Set Support | Character Sets and ... - InformIT." 2007. 15 Jul. 2013 <http://www.informit.com/articles/article.aspx?p=328641>

Footnotes:

1. AB tương đương với Ltd trong tiếng Anh 
2. Precomposed vs composite unicode 

comments powered by Disqus



Hung Neox

Full stack developer.

[GitHub](#) [Twitter](#) [Email](#)



Newest Posts

- [WebAssembly - Phần 2: LLVM](#)

TOC

- WebAssembly - Phần 1: Giới thiệu
 - CORS - Trở về căn bản
 - Định lý Bayes
 - API Lifecycle
 - Character set = characters + encoding method
 - Nên dùng utf8_general_ci hay utf8_unicode_ci?
 - Unicode dựng sẵn hay tổ hợp 2
 - References:
 - Footnotes:
-