Subject: Re: [SQLITE] Search accents insensitive
Posted by dolik.rce on Mon, 19 Jun 2017 19:02:33 GMT
View Forum Message <> Reply to Message

jibe wrote on Fri, 16 June 2017 11:40Thanks for your reply, but did you noticed well that I am working with SQLITE and that I mentioned
Quote:But as far as I know, with SQLITE, we have to create a special collation function, which does not works with LIKE, so we have also to create a special LIKE function.

Is it what you mean when you say
Quote:But you'll still need to write the code to strip the accents.

Yes, that is exactly what I meant.

jibe wrote on Fri, 16 June 2017 11:40My problem is that I don't know exactly what to do, how to write this special collation function inside the SQLITE code and how to modify the LIKE funtion to use it...
It depends on what your strings look like :) If they're in some simple single-byte encoding, like (ISO-8859-2 or Latin-1), you could do just some really simple mapping of chars with values 128 and higher to ASCII. This is easy and straight forward, but works only in given encoding. If your text is unicode, it is much harder. Mirek is just working on that, or you can use one of ICU or iconv libraries, which both have functions to "normalize" unicode to ASCII. See their respective manuals and stackoverflow for examples on how to do that.

jibe wrote on Fri, 16 June 2017 11:40I forgot to say : trying your code, Honza, I got the error "no such function: normalize"Yes, "normalize" would be the UDF written by yourself. Any C function can be added to sqlite using sqlite3_create_function(). The code looks something like this (cut out from one of my older projects):

```
extern "C" {
   void sqlite_md5(sqlite3_context *context, int argc, sqlite3_value **argv) {
      String md5 = MD5StringS((const char*)sSqlite->sqlite3_value_text(argv[0]));
      sSqlite->sqlite3_result_text(context, md5.Begin(), md5.GetCount(), SQLITE_TRANSIENT);
      return;
   }
}

void RegisterSqliteFunctions(Sqlite3Session& session){
   ::sqlite3* db = (::sqlite3*)(Sqlite3Session::sqlite3*)session;
   sqlite3_create_function(db, "md5", 1, SQLITE_ANY, 0, (void*)sqlite_md5, 0, 0);
}
```

Of course, instead of MD5StringS() you'd need to map your input to ASCII or some other normalized form.

Best regards,
Honza