Subject: [SQLITE] Search accents insensitive
Posted by jibe on Thu, 01 Jun 2017 13:25:37 GMT
View Forum Message <> Reply to Message

Hi,

I need to select records without sensivity to the case and accents : the SELECT query should return
Jeremy
Jérémy
JEREMY
JÉRÉMY
etc.

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.

Does somebody already created those functions ?

If not, I'm not very sure to be able to do that in U++ without (maybe a lot of) help... Any suggestion ?

A workaround would be to duplicate the column and fill the second with upper case text without accent, but it is not very elegant !

Subject: Re: [SQLITE] Search accents insensitive
Posted by koldo on Tue, 13 Jun 2017 08:16:49 GMT
View Forum Message <> Reply to Message

jibe wrote on Thu, 01 June 2017 15:25A workaround would be to duplicate the column and fill the second with upper case text without accent, but it is not very elegant !Hello Jibe

That is what I do. There is a function in Functions4U to remove accents and I duplicate the field with the data without accents. After that I use that field for searching.

Subject: Re: [SQLITE] Search accents insensitive
Posted by jibe on Tue, 13 Jun 2017 09:31:50 GMT
View Forum Message <> Reply to Message

Hi, Koldo,

Thanks for your reply

Ok, I'll do that if there is no better solution. Maybe something better will be possible when Mirek

will have finished with Unicodes ?

Subject: Re: [SQLITE] Search accents insensitive
Posted by dolik.rce on Tue, 13 Jun 2017 10:32:54 GMT
View Forum Message <> Reply to Message

Instead of additional column, you can use a user defined function:search * from some_table where normalize(name) like '%jeremy%';
You'd of course need to call the same normalization function on the input as well.

This would give you a workaround around the problems with LIKE, without adding more data to your database (for the price of some CPU cycles). But you'll still need to write the code to strip the accents. I think that Mireks work on unicode decomposition might help with that a lot.

Best regards,
Honza

Subject: Re: [SQLITE] Search accents insensitive
Posted by jibe on Fri, 16 Jun 2017 09:40:11 GMT
View Forum Message <> Reply to Message

Hi, Honza,

Thanks 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.

My 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...

As I think that this problem is frequent and has probably already been solved by somebody, my question was more especially about an already existing solution for SQLITE...

I forgot to say : trying your code, Honza, I got the error "no such function: normalize"

Subject: Re: [SQLITE] Search accents insensitive
Posted by deep on Sat, 17 Jun 2017 13:39:02 GMT
View Forum Message <> Reply to Message

I have not checked this.
But mostly should work.

There are 2 methods suggested.

https://stackoverflow.com/questions/16282083/how-to-ignore-a ccent-in-sqlite-query-android

https://stackoverflow.com/questions/24037982/how-used-regexp -in-sqlite

for using with REGEXP.

https://stackoverflow.com/questions/5071601/how-do-i-use-reg ex-in-a-sqlite-query

---

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

Subject: Re: [SQLITE] Search accents insensitive
Posted by jibe on Mon, 19 Jun 2017 20:36:52 GMT
View Forum Message <> Reply to Message

Hi Honza,

Thanks for your help

All my database is UTF8... So, effectively, it will not be so easy. I'll see if I can wait that Mirek finishes, or I'll do as Koldo : another column... But before, I'll also see if I can use Koldo's function to remove the accents in association with a sqlite3_create_function()...

@deep :

Thanks for your reply, but in my case it will not be very easy, because the code is to allow users to create their own requests. So, I should probably be obliged to automatically generate a regexp each time... And as sqlite doesn't support regexp, I'll also have to extend it with this support... And as says Honza, this depends also what encoding is used...

Subject: Re: [SQLITE] Search accents insensitive
Posted by jibe on Thu, 29 Jun 2017 13:11:45 GMT
View Forum Message <> Reply to Message

Hi,

I'm trying to do it with dolik.rce explanations and sample, but I get this error :
/home/joseph/uppMyApps/Owl/main.cpp (432): error: invalid conversion from 'void*' to 'void (*)(sqlite3_context*, int, sqlite3_value**) {aka void (*)(sqlite3_context*, int, Mem**)}' [-fpermissive]

/home/joseph/upp/uppsrc/plugin/sqlite3/lib/sqlite3.h (4487): note: initializing argument 6 of 'int sqlite3_create_function(sqlite3*, const char*, int, int, void*, void (*)(sqlite3_context*, int, sqlite3_value**), void (*)(sqlite3_context*, int, sqlite3_value**), void (*)(sqlite3_context*))'

I'm surely missing something, but I don't understand what...

Maybe the sSqlite pointer used in dolik.rce's sqlite_md5 function ? I don't understand where it comes from and how to declare/initialise it... So, I omited it in my own code, and the compiler does not complains...

Here are the most interesting parts of my code :

```
extern "C" {
 void Sqlite3NoAccents(sqlite3_context *context, int argc, sqlite3_value **argv) {
  String str = RemoveAccents((const char*)sqlite3_value_text(argv[0]));
  sqlite3_result_text(context, ~str, str.GetCount(), SQLITE_TRANSIENT);
  return;
 }
}

Sqlite3Session sqlite3;
::sqlite3* sqlite3db = (::sqlite3*)(Sqlite3Session::sqlite3*)sqlite3;
sqlite3_create_function(sqlite3db, "NoAccents", 1, SQLITE_ANY, 0, (void *)Sqlite3NoAccents, 0,
0);
```

(RemoveAccents() is from Koldo's Functions4U)

Any help ?

Subject: Re: [SQLITE] Search accents insensitive
Posted by jibe on Fri, 30 Jun 2017 09:12:05 GMT
View Forum Message <> Reply to Message

Hi,

Ooops !

```
extern "C" {
 void Sqlite3NoAccents(sqlite3_context *context, int argc, sqlite3_value **argv) {
  String str = RemoveAccents((const char*)sqlite3_value_text(argv[0]));
  sqlite3_result_text(context, ~str, str.GetCount(), SQLITE_TRANSIENT);
  return;
 }
}

void (*pSqlite3NoAccents)(sqlite3_context *, int, sqlite3_value **);
Sqlite3Session sqlite3;
::sqlite3* sqlite3db = (::sqlite3*)(Sqlite3Session::sqlite3*)sqlite3;
sqlite3_create_function(sqlite3db, "NoAccents", 1, SQLITE_ANY, 0, pSqlite3NoAccents, 0, 0);
```

is surely a lot better ! The compiler accepts it very well, but stays a
/home/joseph/uppMyApps/Owl/book.cpp (2891): error: 'NoAccents' was not declared in this scope

Seems normal that the compiler complains, but I don't know how to declare it...

The sqlite3_create_function is called in main(), when I use the "NoAccents" (Sqlite3NoAccents) in a dialog box...

```
w = (Like ((SqlId)NoAccents((SqlId)("TITLE")),"%" + AsString(~str) + "%"));
book.Query(w);
```
(book is a SqlArray in the dialog's layer)
