Сидоренко В.А., старший преподаватель
ФГБОУ ВО «КалмГУ»,
г. Элиста, Российская Федерация

Проблемы СУБД SQLite в использовании русского текста

Аннотация

В статье показана проблема СУБД SQLite в использовании текста в русской кодировке (вполне возможно, что и для других языков, кроме английского), применяемого в базах данных. Рассмотрены практические примеры обхода этих проблем в некоторых операциях в среде программирования Lazarus. Данная статья будет полезна для разработчиков баз данных, использующих среду программирования Lazarus для работы с SQLite.

Ключевые слова: Lazarus и SQLite, русский текст в SQLite, проблемы русского текста в SQLite.

В учебном процессе при изучении технологий баз данных мы используем интегрированную среду программирования Lazarus, в которой есть возможность работать с некоторыми СУБД, включая SQLite. Для работы с SQLite удобно также использовать специальный менеджер БД **SQLiteStudio**, с помощью которого можно управлять базой в визуальном режиме. Там же есть возможность работать в интерактивном режиме в SQL: Обычно, разрабатывая приложения, программист проверяет какие-то запросы к базе данных именно таким способом, с помощью подобного менеджера. То есть, различные эксперименты, опыты, идеи удобно проверять напрямую на базе данных, используя язык SQL-запросов – если это возможно. Когда эти идеи проверены и подтверждены опытом, можно переносить текст SQL-запросов в рабочее приложение, т.е. в данном случае в программу на Lazarus. Таким методом работа по разработке приложения идёт быстрее и надёжнее.

Иногда в SQLite встречаются проблемы, которые обнаруживаются как раз при выполнении тестовых SQL-запросов. Одну из них рассмотрим в этой статье. В языке SQL есть функция UPPER, которая переводит буквы в верхний регистр. Посмотрим результат запроса

```
SELECT NAME, UPPER(SNAME) FROM TAB3
```

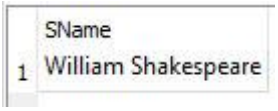
	SName	UPPER(SName)
1	Фёдоров Я.Я.	Фёдоров Я.Я.
2	William Shakespeare	WILLIAM SHAKESPEARE
3	jack London	JACK LONDON
4	пушкин А.С.	пушкин А.С.

Рис.1. Работа функции UPPER

Здесь мы видим, что UPPER правильно работает с английским алфавитом, но не работает с русским.

Есть ещё интересные моменты в работе SQL с текстом в базах данных SQLite. Например:

```
SELECT SName FROM TAB3 WHERE SName LIKE '%shake%'
```



	SName
1	William Shakespeare

Рис.2. Работа оператора LIKE

Здесь видим, что при поиске строки по фрагменту текста с помощью оператора LIKE в английском тексте даже не учитывается регистр, что обычно удобно для пользователя. То есть, даже не надо для такого случая пользоваться функцией UPPER, например, запрос

```
SELECT SName FROM TAB3 WHERE UPPER(SName) LIKE '%SHAKE%'
```

здесь упрощается до приведённого здесь ранее.

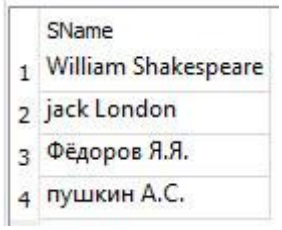
Естественно в русском тексте такого нет. То есть, запрос

```
SELECT SName FROM TAB3 WHERE UPPER(SName) LIKE '%УШКИН%'
```

здесь ничего не даст, хотя по идее ожидался Пушкин.

Рассмотрим попутно ещё работу команды ORDER BY (упорядочивание).

```
SELECT SName FROM TAB3 ORDER BY SName
```

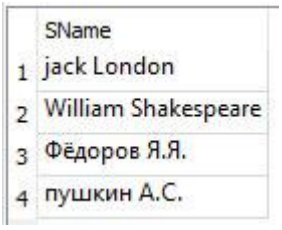


	SName
1	William Shakespeare
2	jack London
3	Фёдоров Я.Я.
4	пушкин А.С.

Рис.3. Работа оператора ORDER BY

Здесь видим, что при упорядочивании слов первыми идут заглавные буквы по алфавиту, затем после всех заглавных – строчные. То есть, чувствительность к регистру есть и у английских букв и у русских. В обычной жизни это неудобно. Поэтому напрашивается такой запрос:

```
SELECT SName FROM TAB3 ORDER BY UPPER(SName)
```



	SName
1	jack London
2	William Shakespeare
3	Фёдоров Я.Я.
4	пушкин А.С.

Рис.4. Работа оператора ORDER BY вместе с UPPER

Здесь для английского текста получен положительный результат – все строки строго по алфавиту, независимо от регистра. С русским текстом всё осталось, как и прежде.

Обычно перечисленные выше действия нам нужны в приложениях. В данном случае для разработки приложений имеем среду программирования Lazarus. Поищем здесь пути решения указанных проблем.

Для поиска по фрагменту без учёта регистра предлагается следующий метод. Команду SELECT применяем без LIKE, а для фильтра используем метод OnFilterRecord объекта SQLQuery:

```
procedure TFrmSTrack.SQLQuery1FilterRecord(DataSet: TDataSet;
  var Accept: Boolean);
begin
  Accept:=
    pos(SName, AnsiUpperCase(DataSet.Fields[0].AsString)) > 0;
end;
```

Здесь SName строка поиска, которую получаем в момент обработки события OnClick компонента Button (кнопка «Поиск»):

```
procedure TFrmSTrack.Button1Click(Sender: TObject);
begin
  SName:= AnsiUpperCase(Trim(LabeledEdit1.Text));
  SQLQuery1.Close;  // надо закрыть !!
  SQLQuery1.Filtered:= True;
  SQLQuery1.Open;
end;
```

Здесь перезагружается набор данных `SQLQuery1`, при этом перебираются все записи, удовлетворяющие запросу `SELECT` (подготовленному заранее), но в результирующий набор данных попадают только записи, отобранные в процедуре `SQLQuery1FilterRecord`.

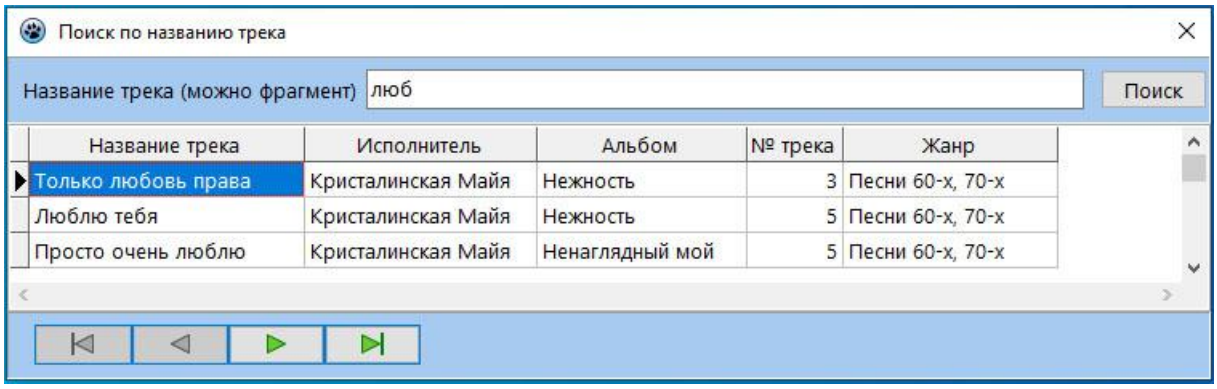


Рис.5. Результат работы фильтра

Список использованной литературы

1. Википедия: SQLite https://ru.wikipedia.org/wiki/SQLite
2. Википедия: Lazarus https://ru.wikipedia.org/wiki/Lazarus
3. Алексеев Е. Р., Чеснокова О. В., Кучер Т. В. Free Pascal и Lazarus: Учебник по программированию. – М.: Альт Линукс, ДМК Пресс, 2010. 440 с.
4. Осипов Д.Л. Базы данных и Delphi. Теория и практика. СПб: БХВ-Петербург, 2011. 752 с
5. Шейкер Т.Д. Разработка приложений баз данных в системе Delphi: учеб. пособие. Владивосток: Изд-во ДВГТУ, 2009. 138 с.

Сидоренко В.А., 2023