

ИСПОЛЬЗОВАНИЕ СТОРОННЕЙ БАЗЫ ДАННЫХ В ПРИЛОЖЕНИЯХ ПОД ANDROID

В данной статье приводится пример конвертации базы данных из формата FoxPro в SQLite.

This article provides an example of converting a database from the FoxPro format to SQLite.

Ключевые слова: база данных; FoxPro; SQLite; Android; конвертация.

Key words: database; FoxPro; SQLite; Android; conversion.

Android-приложения работают в условиях жестких ограничений на объем используемой памяти. Поэтому существует два основных подхода к использованию баз данных: либо хранить базу данных на сервере и обращаться к ней через соответствующий API, либо использовать внутреннюю базу данных SQLite.

Первый подход позволяет хранить большие объемы данных и использовать практически любую СУБД. Недостатком этого варианта с точки зрения клиента является отсутствие автономного режима работы (без доступа к Интернету), а с точки зрения разработчика – необходимость создания, администрирования и поддержки сервера для размещения этой базы данных.

Использование внутренней базы данных SQLite – хорошее решение для небольших по объему баз данных. SQLite называется встроенной, поскольку не использует концепцию «клиент-сервер», а ее движок после компоновки программы становится составной частью кода. Сама база данных представляет собой обычный файл. Когда она не используется, то не расходует процессорное время. Это особенно важно на мобильных устройствах, чтобы избежать разрядки аккумулятора.

Базы данных SQLite невероятно надежны. Они поддерживают транзакции баз данных (другими словами, если при обновлении нескольких блоков данных что-то пойдет не так, SQLite сможет вернуться к исходному состоянию). Кроме того, операции чтения и записи данных реализуются на оптимизированном коде C. Этот код не только быстро работает, но и сокращает объем необходимых вычислительных ресурсов.

Таким образом, в качестве внутренней базы данных в приложении под Android практически нет альтернативы использованию SQLite. Но как быть, если база данных, которую требуется использовать, уже существует в другом формате? Как преобразовать эти данные в формат SQLite, подключить готовую базу к проекту и использовать в программе?

Такая задача возникла в процессе работы над приложением «КОPERTIME», которое предоставляет пользователю расписание занятий (для студента или для преподавателя) и должно использовать базу данных диспетчерской, существующую в формате FoxPro. Данная статья посвящена описанию процесса решения данной проблемы.

Конвертация базы данных FoxPro в SQLite. Идея конвертации базы данных FoxPro в SQLite основана на использовании приложения DBBrowser (SQLite). Это бесплатное приложение для визуального проектирования и редактирования баз данных SQLite. Скачать его можно с официального сайта (<https://sqlitebrowser.org/>). В числе прочих возможностей этого приложения имеется вариант импорта таблицы из файла в формате CSV. Поэтому, если имеется возможность выгрузить таблицу в этом формате из исходной базы данных, то дальнейшее преобразование не представляет проблемы.

Приложение VisualFoxPro имеет вариант экспорта данных в формат Excel, который легко преобразуется затем в CSV. Однако, к сожалению, это приложение уже устарело и не может быть установлено на компьютер с 64-битной операционной системой. Поэтому было решено использовать бесплатное приложение OpenOffice, скачать которое можно с сайта (<https://openoffice.ru>). Приложение умеет открывать базы данных различного формата, в том числе таблицы FoxPro с расширением .DBF. А среди вариантов сохранения данных имеется и CSV-формат. Некоторые, вполне решаемые, проблемы могут возникнуть только с представлением кириллицы.

Ниже подробно описан процесс конвертации таблиц FoxPro и добавления их в базу данных SQLite.

1. Откроем OpenOffice и выберем приложение для работы с базами данных (Database). Создадим новую базу данных и сохраним ее произвольным именем. Затем откроем файл с таблицей данных FoxPro (File/Open), например таблицу со списком групп sgroup.dbf. В процессе открытия файла потребуются указать кодировку. Нужно подобрать подходящую кодировку, чтобы русские символы верно отражались. Это может быть Windows 1251 либо DOS/OS2-866/Russian (рисунок 1).

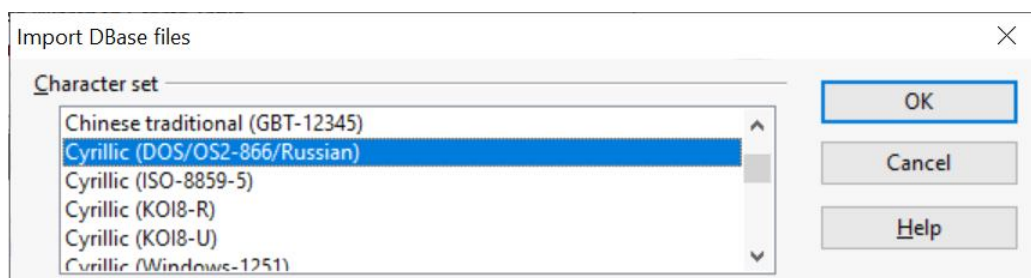


Рисунок 1 – Выбор кодировки при загрузке .dbf - файла

Далее следует поправить заголовки полей, поскольку в них кроме собственно имени поля попадает его тип и размер (просто удаляем лишнее). Отредактированную таблицу нужно сохранить в формате CSV (File/SaveAs и тип файла выбрать TextCSV). Если появляется предупреждение о несоответствии форматов, нужно продолжить преобразование, выбрав *Keep Current Format*. При запросе параметров csv-файла нужно установить разделителем полей точку с запятой (Fielddelimiter) – рисунок 2.

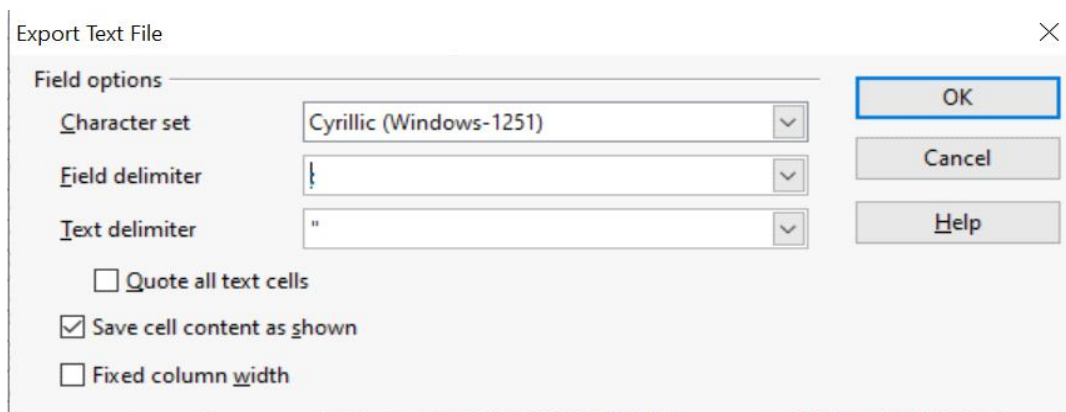


Рисунок 2 – Параметры экспорта файла в CSV-формат

2. Далее откроем DBBrower (SQLite) и создадим новую базу данных и дадим ей имя timetable.db, если это первая таблица. Либо откроем эту уже существующую базу и продолжим добавлять в нее другие таблицы. Если при создании пустой базы появляется окно *Редактирование определения таблицы*, то нужно закрыть его, нажав кнопку *Отмена*. Мы будем импортировать таблицу, поэтому ее структура должна определиться автоматически.

Импорт таблицы в формате .csv выполняется командой *Файл/Импорт/Таблицы из файла CSV*. В окне параметров csv-формата следует верно указать разделитель (;) и кодировку (Windows 1251) (рисунок 3).

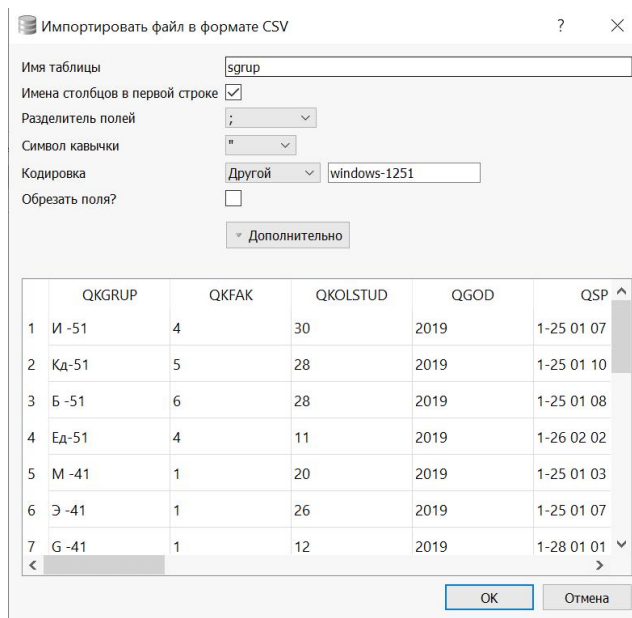


Рисунок 3 – Импорт файла в формате CSV в DBBrowser (SQLite)

Для удобства использования таблицы SQLite, как правило, имеют ключевое поле `_id`, для которого устанавливается авто инкремент. Добавим это поле в таблицу. Для этого выделим добавленную таблицу и зададим команду *Редактирование/Модифицировать таблицу*. В окне Редактирование определения таблицы нажимаем кнопку *Добавить поле*, даем полю имя `_id`, и проставляем флажки ПК (первичный ключ) и АИ (авто инкремент). Поле в нижней части окна показывает, как изменяется команда `CREATETABLE` (рисунок 4).

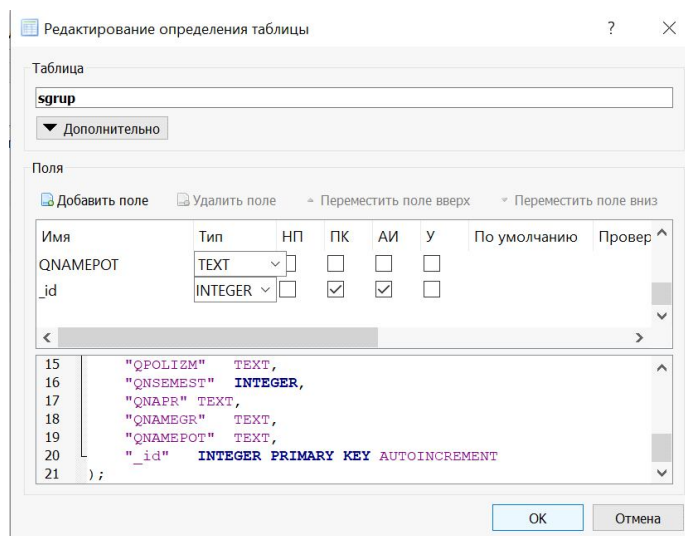


Рисунок 4 – Добавление ключевого поля `_id` в таблицу

Аналогичным образом добавляем другие необходимые таблицы в созданную базу данных `timetable.db`.

Таким образом, задача конвертации базы данных из одного формата в другой была решена с использованием промежуточного формата CSV и двух бесплатных приложений: OpenOffice и DBBrowser (SQLite).

Для использования полученной базы данных SQLite в приложении *KOPERTIME*, следует создать в проекте папку `assets` (*File/New/Folder/AssetsFolder*) и скопировать в нее файл `timetable.db`.

Дальнейшая работа с базой данных выполняется стандартными средствами классов `SQLiteOpenHelper`, `SQLiteDatabase` и `Cursor` [1]. Создается класс-наследник `SQLiteOpenHelper`, который предоставляет средства создания и управления базой данных. В конструкторе данного

класса выполняется копирование базы данных из папки assets в предназначенную для нее папку databases [2]. Те же действия выполняются и в методе onUpgrade(), который вызывается при изменении базы данных. Для подключения новой версии базы к проекту необходимо скопировать обновленный файл в папку assets, а в тексте программы увеличить значение константы DB_VERSION. Несоответствие номера версии вызовет метод onUpgrade(), и база данных обновится.

Список использованной литературы

1. **Гриффитс, Д.** HeadFirst. Программирование для Android / Дон Гриффитс, Дэвид Гриффитс. – СПб. : Питер, 2016. – 704 с.
2. **Подключение** существующей БД SQLite в AndroidStudio [Электронный ресурс]. – Режим доступа : <http://blog.harrix.org/article/6610>. – Загл. с экрана. – Дата доступа : 05.02. 2020.