Emro Science and Education
Open-access scientific journal
Journal homepage: https://journal.emro.uz
OPEN ACCESS

CONFERENCE PAPER OPEN ACCESS

Vol. 1 No. 1 (2023) · “Axborot oqimlari va intellektual tizimlarning ijtimoiy, iqtisodiy va texnik-texnologik tarmoqlardagi o‘rni” mavzus...

materiallar to‘plami

WORKING WITH SQLITE DATABASES

Ikromov Khusan Kholmakhamatovich, Anijan machine-building intitute.
Akhmadaliyeva Dilorom, Markhamat district school №39

Abstract. This article details the principles and methods of working with SQLite databases. It also shows how to connect to SQLite database for new users.

Keywords. Data, databases, SQLite, Android, data type.

Android has built-in support for one of the most common database management systems - SQLite. For this, the package android. database. sqlite a set of classes is defined that allow you to work with SQLite databases. And each application can create its own database.

To use SQLite on Android, you need to create a database using an SQL statement. After that, the database will be stored in the application directory along the path:

DATA / data /[Application_Name]/ databases /[Database_File_Name]

Android OS by default already contains a number of built-in bad SQLite, which are used by standard programs - for the list of contacts, for storing photos from the camera, music albums, etc.

The basic functionality for working with databases is provided by the package android. database. Functionality directly for working with SQLite is in the package android. database. sqlite.

A database in SQLite is represented by a class android. database. sqlite. SQLiteDatabase. It allows you to query the database, perform various manipulations with it.

Class android. database. sqlite. SQLiteCursor provides a query and allows you to return a set of rows that match that query.

Class android. database. sqlite. SQLiteQueryBuilder allows you to create SQL queries.

The sql expressions themselves are represented by the class android. database. sqlite. SQLiteStatement, which allow you to insert dynamic data into expressions using placeholders.

Class android. database. sqlite. SQLiteOpenHelper allows you to create a database with all tables if they do not already exist.

SQLite uses the following data type system:

- INTEGER: represents an integer equivalent to a type int in java;
- REAL: represents a floating point number, similar to float and double in java;
- TEXT: represents a set of characters, analogous to String and char in java;
- BLOB: represents an array of binary data, such as an image, analogous to type int in java.

The data being stored must represent the appropriate types in java.

To create or open a new database from the Android Activity code, we can call the method openOrCreateDatabase (). This method maybe accept three parameter:

- title for bases data;

Published: 10.11.2023

Cite this article: Ikromov Khusan Kholmakhamatovich, Akhmadaliyeva Dilorom (2023). Working with SQLite databases. Emro Science and Education, 1(1). https://journal.emro.uz/public/maqola/401

© 2023 The Author(s). Published by Emro Science and Education. This is an open access article.

- a numeric value that specifies the mode of operation (usually as a constant MODE_PRIVATE);
- optional parameter as an object SQLiteDatabase. CursorFactory, which represents a factory for creating a cursor for working with a database.

For example, creating a database app. db:

```
SQLiteDatabase db = getBaseContext (). openOrCreateDatabase (" app.db ", MODE_PRIVATE, null);
```

To execute a database query, you can use the method execSQL SQLiteDatabase class. An SQL expression is passed to this method. For example, creating a users table in the database:

```
SQLiteDatabase db = getBaseContext (). openOrCreateDatabase (" app.db ", MODE_PRIVATE, null);
db.execSQL ("CREATE TABLE IF NOT EXISTS users (name TEXT, age INTEGER)");
```

If we need not only to execute an expression, but also to get some data from the database, then the method is used rawQuery (). This method takes an SQL expression as a parameter, as well as a set of values for the sql expression. For example, getting all objects from bases data:

```
SQLiteDatabase db = getBaseContext (). openOrCreateDatabase ("app.db", MODE_PRIVATE, null);
db.execSQL ("CREATE TABLE IF NOT EXISTS users (name TEXT, age INTEGER)");
Cursor query = db.rawQuery ("SELECT * FROM users;", null);
if( query.moveToFirst ()){
= query.getString ( 0 );
int age = query.getInt ( 1 );}
```

Method db . rawQuery () returns a Cursor object with which we can retrieve the received data.

It is possible that there will be no objects in the database, and for this the method query. moveToFirst () trying to navigate to the first object retrieved from the database. If this method returns false, then the query has not received any data from the database.

App database.db new table users, and then two objects are added to the database using an SQL INSERT statement.

Next, using the SELECT statement, we get all the added users from the database in the form of a Cursor.

Calling query.moveToNext() moving in a cycle while consecutively for all objects.

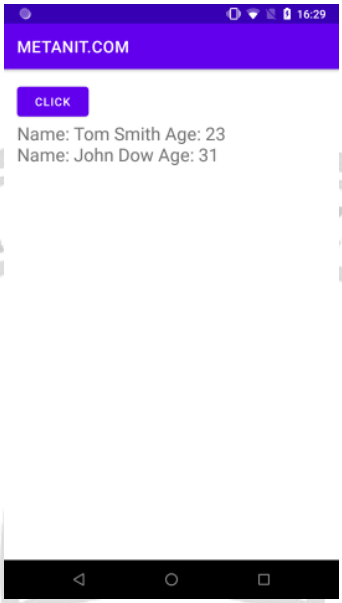
Methods are used to get data from the cursor. query.getString (0) and query. getInt (1). In parentheses, the number of the column from which we get the data is passed to the methods. For example, above we added the username first as a string and then the age as a number. This means that the zero column will be the string value, which is obtained using the method getString(), and the next - the first column is the numeric value for which the method is applied getInt ().

Once we're done with the cursor and the database, we close all related objects:

```
1  query.close ();
2  db.close ();
```

If we don't close the cursor, we might run into a memory leak problem.

And if we turn to the application, then after clicking on the button, the added data will be displayed in the text field:



References:

1. Butayev, E. H. (2021). Algoritmlar Tizimli Tahlili Va Ularning Takomillashtirish. ФАРҒОНА ПОЛИТЕХНИКА ИНСТИТУТИ И Л М И Й–Т Е Х Н И К А ЖУРНАЛИ, 25, 36-41.
2. Khomitjonovich, B. E. (2022). DEVELOPING ALGORITHM FOR CONVERTING INTEGER TYPE VARIABLES INTO STRING VIEW. Academicia Globe: Inderscience Research, 3(04), 105-108.
3. Xoldarboyev, R. A., & Abduvaxobova, R. A. Q. (2022). Kiberxavfsizlik. Science and Education, 3(7), 29-34.
4. Xolmaxamatovich, I. X. (2023). AXBOROT VA KOMMUNIKATSION TEXNOLOGIYALAR BILAN BIRGALIKDA TA’LIM SIFATINI YAXSHILASH VA INNOVATSION G‘OYALARNI AMALDA QO‘LLASH. JOURNAL OF INNOVATIONS IN SCIENTIFIC AND EDUCATIONAL RESEARCH, 6(2), 270-272.
5. Икромов, Х. Х. (2021). СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ. Universum: технические науки, (2-1), 21-22.