

SQLite w Androidzie

Zadanie:

Stwórz prosty formularz z polami: imię, e-mail, komentarz. Za pomocą przycisku dodaj możliwość dodania nowego komentarza do bazy danych. Dodaj kolejne przyciski, które wykonają następujące zapytania (wynik może być przedstawiony za pomocą Toast):

- Zlicz liczbę dodanych komentarzy
- Policz liczbę komentarzy z adresem e-mail a@wp.pl
- Wyświetl komentarze dłuższe niż 100 znaków
- Kto dodał najwięcej komentarzy?
- Wyświetl komentarze osoby o imieniu Donald
- Usuń komentarze z adresem e-mail a@a.pl

SQLite jest zaimplementowany w każdym urządzeniu z systemem Android. Oznacza to, że nie potrzebujemy zajmować się administracją bazy danych. Programista musi zdefiniować instancję tworzenie i aktualizowania bazy danych. Pozostałe czynności są wykonywane automatycznie przez platformę Android.

Biblioteki:

```
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
```

1. Tworzenie bazy danych

Zmienne globalne:

```
private final String DB_NAME = "Baza.db"; //Nazwa dla nowej bazy
SQLiteDatabase baza = null;
```

Przy tworzeniu aplikacji możemy spróbować otworzyć lub utworzyć bazę danych w przypadku gdy one nie istnieje funkcję `openOrCreateDatabase`. Przyjmuje one trzy parametry: nazwę bazy, tryb dostępu oraz `CursorFactory`.

```
try{
    baza = this.openOrCreateDatabase(DB_NAME, MODE_PRIVATE, null);
    baza.execSQL("CREATE TABLE IF NOT EXISTS Persons (FirstName VARCHAR, Email VARCHAR, Comment VARCHAR)");
}
catch(SQLiteException e) {
    Log.e(getClass().getSimpleName(), "Could not create or Open the database"); }
```

2. Dodawanie danych

```
baza.execSQL("INSERT INTO Persons Values('Donald','q@q.pl','ala ma kota');");
```

Za pomocą funkcji `execSQL` możemy do bazy danych generować zapytania, dodawać nowe rekordy czy je usuwać.

3. Przeszukiwanie bazy danych

Do przeszukiwania bazy potrzebny jest kursor, który służy nam do dostępu do rezultatów zwróconych przez zapytanie SQL.

```
Cursor cursor = baza.rawQuery("SELECT Email FROM Persons WHERE FirstName like 'Lech'",null);
if(cursor.moveToFirst()){ //Metoda zwraca FALSE jeśli cursor jest pusty
    do{
        //getString Returns the value of the requested column as a String.
        //getColumnIndex Returns the zero-based index for the given column name,
        //      or -1 if the column doesn't exist.

        String FirstName = cursor.getString(cursor.getColumnIndex("Email"));
    }while(cursor.moveToNext()); //Metoda zwraca FALSE wówczas gdy cursor przejdzie
ostatni wpis
    cursor.close();
}
```

4. Zamykanie bazy danych

```
if (baza != null)
    baza.execSQL("DELETE FROM Persons ");
baza.close();
```