

SQLite

Вспомогательный класс для базы данных

```
package com.example.Data_table;

import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

class DBHelper extends SQLiteOpenHelper {
    public DBHelper(Context context) {
        super(context, "myDB", null, 1); //myDB – имя базы данных
    }

    public void onCreate(SQLiteDatabase db) {
        Log.d("Ivan", "--- onCreate database ---"); //создадим новую таблицу в базе
        db.execSQL("create table mytable ("
            + "id integer primary key autoincrement,"
            + "name text,"
            + "email text" + ");");
    }

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS mytable"); //перезапишем старую таблицу
        db.execSQL("create table mytable ("
            + "id integer primary key autoincrement,"
            + "name text,"
            + "email text" + ");");
    }
}
```

Основная Активность

```
package com.example.Data_table;

import android.database.sqlite.SQLiteDatabase;
import android.database.Cursor;
import android.util.Log;

public class MyActivity extends Activity {

    SQLiteDatabase db;
    DBHelper dbHelper;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

```
dbHelper=new DBHelper(MyActivity.this);  
db=dbHelper.getWritableDatabase(); // Открываем базу для записи
```

```
    //Добавлять данные в таблицу можно командой  
//INSERT INTO mytable ( 'id','name','email') VALUES ('1','Ivanov','spam1@mail.ru');  
//INSERT INTO mytable ( 'id','name','email') VALUES ('2','Petrov','spam2@mail.ru');  
//INSERT INTO mytable ( 'id','name','email') VALUES ('3','Sidorov','spam3@mail.ru');  
//-----
```

```
    // Теперь считаем из базы информацию  
    Cursor c; //Создаем курсор (это указатель на контекстную область памяти, с помощью  
    которого программа на языке SQL может управлять контекстной областью и ее  
    состоянием во время обработки оператора)
```

```
    c = db.rawQuery ("Select * from mytable where id<0"); //Выполняем запрос из базы
```

```
    if (c.moveToFirst()) { //переходим на первый элемент если он есть  
    do {
```

```
        int ColIndex = c.getColumnIndex("id"); // получаем номер колонки с имени id
```

```
        Log.v("Ivan",c.getString(ColIndex)); //выводим строковое значение по номеру  
        колонки
```

```
    } while(c.moveToNext()); //переходим к следующему элементу  
}
```