Лабораторная работа 3.

**БД SQLite в Android приложении.**

Лабораторный практикум

БД SQLite в Android-приложении

Упражнение 1

База данных **linux\_icons.db** содержит одну таблицу *myimg*, в которой созданы две колонки:   
1) *id* - номер изображения,   
2) *imgname* - название файла с изображением.   
Файлы в формате JPEG хранятся на Android-устройстве в папке **/data**. Возможно добавление записей в таблицу.   
Задание для самостоятельной работы: добавьте в приложение возможность создавать третью колонку в таблице: imganim, Анимация.   
**Внимание**! Откройте файл   
gen/org/ggpi/MyImgSQLite01/R.java   
и проверяйте **после** выполнения каждого из ниже приведённых соответствующих примеров, что в нём зарегистрированы объекты:

Самостоятельно определите: какие объекты здесь должны быть?

Указания к выполнению

1. После компиляции проверьте, создана ли база данных на Android-устройстве с помощью adb в командной строке, см. [Краткую теорию](https://sites.google.com/site/ifizmat/java/lab-02).

*Пример 1.1*. Файл **res/layout/main.xml**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<LinearLayout

android:id="@+id/linearLayout1"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:text="ID" >

</TextView>

<EditText

android:id="@+id/etID"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1" >

<requestFocus>

</requestFocus>

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout3"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:text="Name" >

</TextView>

<EditText

android:id="@+id/etName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1" >

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout2"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<Button

android:id="@+id/btnAdd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Add" >

</Button>

<Button

android:id="@+id/btnRead"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Read" >

</Button>

<Button

android:id="@+id/btnClear"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Clear" >

</Button>

</LinearLayout>

</LinearLayout>

*Пример 1.2*. Файл **MainActivity.java**

package org.ggpi.MyImgSQLite01;

import android.app.Activity;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener {

private static final String DATABASE\_NAME = "linux\_icons.db";

private static final String DATABASE\_TABLE = "myimages";

final String LOG\_TAG = "myLogs";

Button btnAdd, btnRead, btnClear;

EditText etID, etName;

DBHelper dbHelper;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

btnAdd = (Button) findViewById(R.id.btnAdd);

btnAdd.setOnClickListener(this);

btnRead = (Button) findViewById(R.id.btnRead);

btnRead.setOnClickListener(this);

btnClear = (Button) findViewById(R.id.btnClear);

btnClear.setOnClickListener(this);

etID = (EditText) findViewById(R.id.etID);

etName = (EditText) findViewById(R.id.etName);

// создаем объект для создания и управления версиями БД

dbHelper = new DBHelper(this);

}

@Override

public void onClick(View v) {

// создаем объект для данных

ContentValues cv = new ContentValues();

// получаем данные из полей ввода

String id = etID.getText().toString();

String name = etName.getText().toString();

// подключаемся к БД

SQLiteDatabase db = dbHelper.getWritableDatabase();

switch (v.getId()) {

case R.id.btnAdd:

Log.d(LOG\_TAG, "--- Insert in mytable: ---");

// подготовим данные для вставки в виде пар: наименование столбца - значение

cv.put("id", id);

cv.put("imgname", name);

// вставляем запись и получаем ее ID

long rowID = db.insert(DATABASE\_TABLE, null, cv);

Log.d(LOG\_TAG, "row inserted, ID = " + rowID);

break;

case R.id.btnRead:

Log.d(LOG\_TAG, "--- Rows in mytable: ---");

// делаем запрос всех данных из таблицы mytable, получаем Cursor

Cursor c = db.query(DATABASE\_TABLE, null, null, null, null, null, null);

// ставим позицию курсора на первую строку выборки

// если в выборке нет строк, вернется false

if (c.moveToFirst()) {

// определяем номера столбцов по имени в выборке

int idColIndex = c.getColumnIndex("id");

int nameColIndex = c.getColumnIndex("imgname");

do {

// получаем значения по номерам столбцов и пишем все в лог

Log.d(LOG\_TAG,

"ID = " + c.getInt(idColIndex) +

", name = " + c.getString(nameColIndex));

// переход на следующую строку

// а если следующей нет (текущая - последняя), то false - выходим из цикла

} while (c.moveToNext());

} else

Log.d(LOG\_TAG, "0 rows");

c.close();

break;

case R.id.btnClear:

Log.d(LOG\_TAG, "--- Clear mytable: ---");

// удаляем все записи

int clearCount = db.delete(DATABASE\_TABLE, null, null);

Log.d(LOG\_TAG, "deleted rows count = " + clearCount);

break;

}

// закрываем подключение к БД

dbHelper.close();

}

class DBHelper extends SQLiteOpenHelper {

public DBHelper(Context context) {

// конструктор суперкласса

super(context, DATABASE\_NAME, null, 1);

}

@Override

public void onCreate(SQLiteDatabase db) {

Log.d(LOG\_TAG, "--- onCreate database ---");

// создаем таблицу с полями

// db.execSQL("create table "+ DATABASE\_TABLE + " ("

// + "id integer primary key autoincrement,"

// + "imgname text" + ");");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

}

}

}