МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Новосибирский национальный исследовательский государственный университет»

(Новосибирский государственный университет, НГУ)

Структурное подразделение Новосибирского государственного университета –

Высший колледж информатики Университета (ВКИ НГУ)

КАФЕДРА ИНФОРМАТИКИ

|  |  |
| --- | --- |
|  | Отчёт защищён с оценкой |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

ПРЕДМЕТ: Программирование мобильных приложений

ТЕМА: Работа с базой данных SQLite

|  |  |
| --- | --- |
| Преподаватель  к.т.н., ассистент | Пахомов Иван Владимирович  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2024 г. |
| Студент 3 курса,  гр. 107г2 | Фортуна Максим  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2024 г |

Новосибирск

2024

Для начала работы необходимо:

1. Создание нового проекта “Empty Views Activity”.

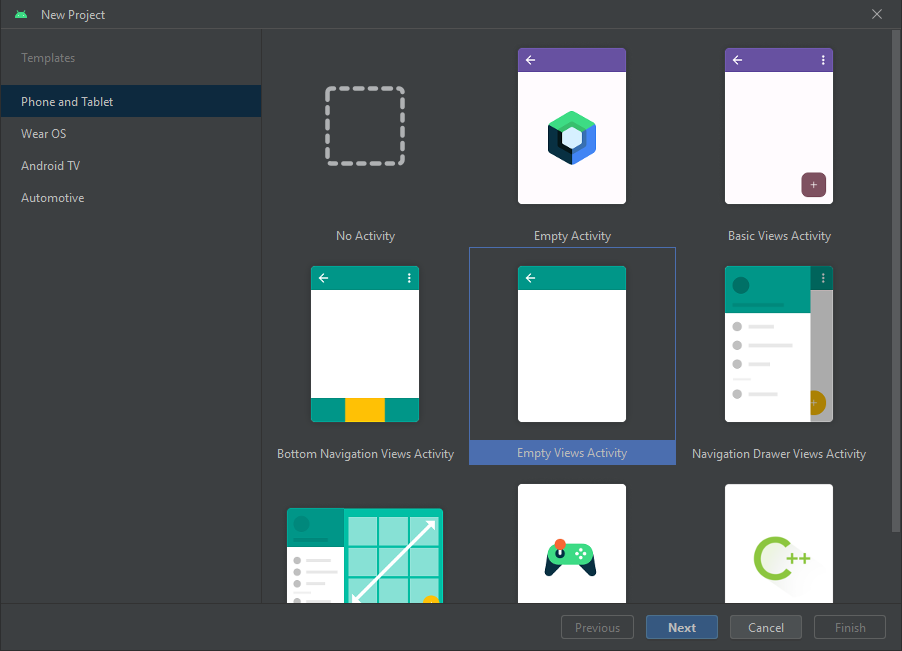


Рисунок - Выбор шаблона проекта

1. Добавление полей EditText, а также button,

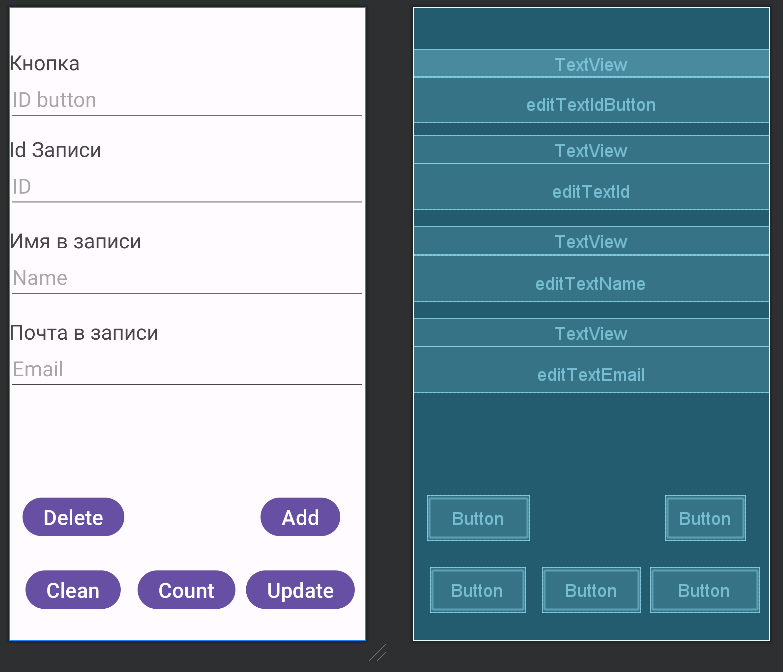
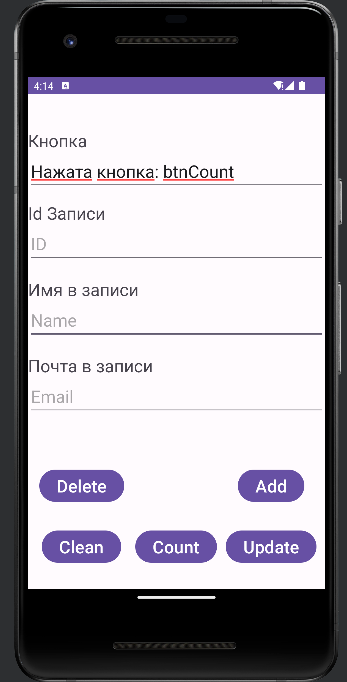
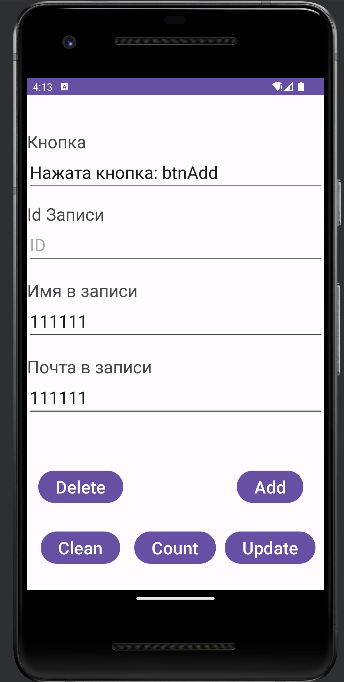


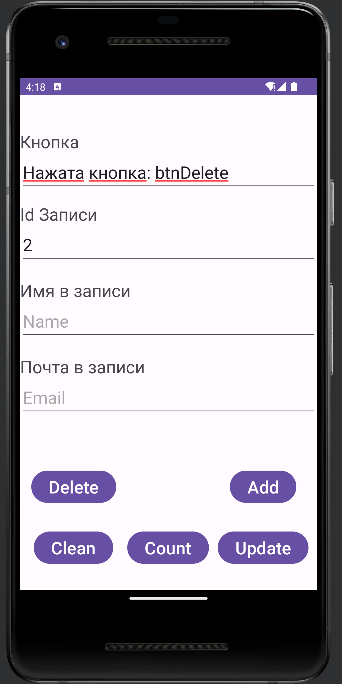
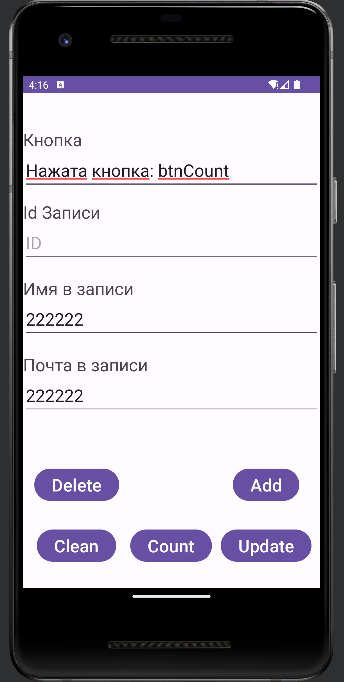
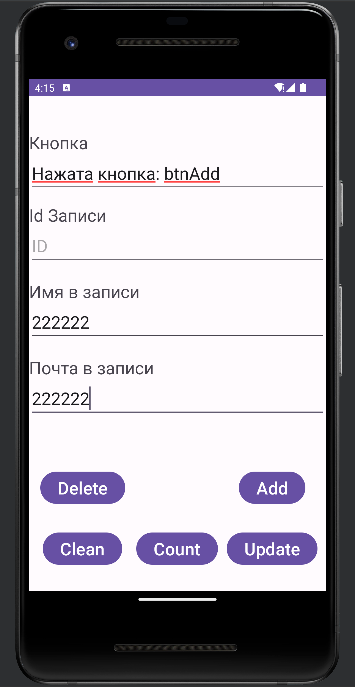
Рисунок - Размещение элементов на рабочую область

1. Оформив код, мы получим такой функционал.
   1. Добавление записи:

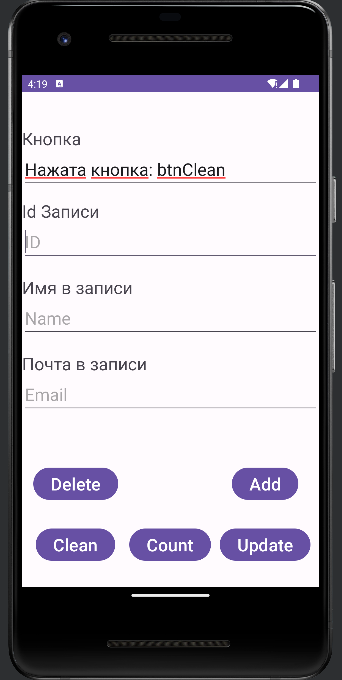


if (v.getId() == R.id.*btnAdd*) {  
 editText.setText(null);  
 String buttonName = getResources().getResourceEntryName(v.getId());  
 editText.setText("Нажата кнопка: " + buttonName);  
  
 contentValues.put(DBHelper.*KEY\_NAME*, name);  
 contentValues.put(DBHelper.*KEY\_MAIL*, email);  
 database.insert(DBHelper.*TABLE\_PERSONS*, null, contentValues);

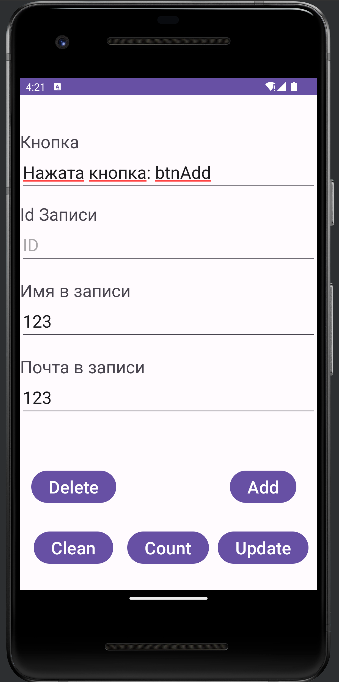
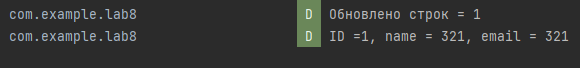
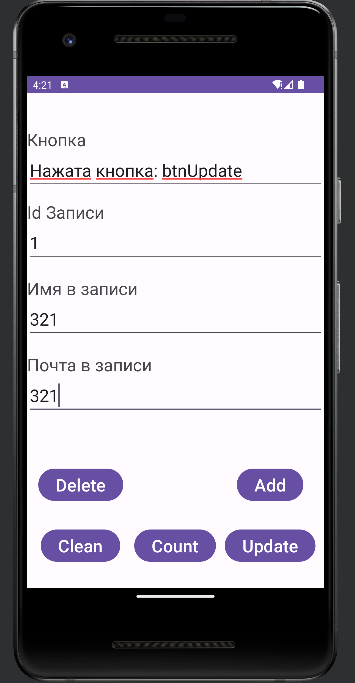
1. Удаление конкретной записи:

 else if (v.getId() == R.id.*btnDelete*) {  
 editText.setText(null);  
 String buttonName = getResources().getResourceEntryName(v.getId());  
 editText.setText("Нажата кнопка: " + buttonName);  
  
 if (!ID.equalsIgnoreCase(""))  
 {  
 int delCount = database.delete(DBHelper.*TABLE\_PERSONS*, DBHelper.*KEY\_ID* + "= " + ID, null);  
 Log.*d*("mLog", "Удалено строк = " + delCount);  
 }

1. Очистка всех записей:

 else if (v.getId() == R.id.*btnClean*) {  
 editText.setText(null);  
 String buttonName = getResources().getResourceEntryName(v.getId());  
 editText.setText("Нажата кнопка: " + buttonName);  
  
 database.delete(DBHelper.*TABLE\_PERSONS*, null, null);

1. Обновление конкретной записи:

  else if (v.getId() == R.id.*btnUpdate*) {  
 editText.setText(null);  
 String buttonName = getResources().getResourceEntryName(v.getId());  
 editText.setText("Нажата кнопка: " + buttonName);  
  
 if (!ID.equalsIgnoreCase(""))  
 {  
 contentValues.put(DBHelper.*KEY\_MAIL*, email);  
 contentValues.put(DBHelper.*KEY\_NAME*, name);  
 int updCount = database.update(DBHelper.*TABLE\_PERSONS*, contentValues, DBHelper.*KEY\_ID* + "= ?", new String[]{ID});  
 Log.*d*("mLog", "Обновлено строк = " + updCount);  
 }

**ПРИЛОЖЕНИЕ**

* activity\_main.xml
* <?xml version="1.0" encoding="utf-8"?>  
  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   xmlns:tools="http://schemas.android.com/tools"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   tools:context=".MainActivity">  
    
   <EditText  
   android:id="@+id/editTextId"  
   android:layout\_width="0dp"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginTop="100dp"  
   android:ems="10"  
   android:hint="ID"  
   android:inputType="text"  
   android:textSize="24sp"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
   <EditText  
   android:id="@+id/editTextIdButton"  
   android:layout\_width="0dp"  
   android:layout\_height="wrap\_content"  
   android:ems="10"  
   android:hint="ID button"  
   android:inputType="text"  
   android:textSize="24sp"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.0"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
   <EditText  
   android:id="@+id/editTextName"  
   android:layout\_width="0dp"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginTop="50dp"  
   android:ems="10"  
   android:hint="Name"  
   android:inputType="text"  
   android:textSize="24sp"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/editTextId" />  
    
   <EditText  
   android:id="@+id/editTextEmail"  
   android:layout\_width="0dp"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginTop="50dp"  
   android:ems="10"  
   android:hint="Email"  
   android:inputType="text"  
   android:textSize="24sp"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/editTextName" />  
    
   <Button  
   android:id="@+id/btnAdd"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginBottom="32dp"  
   android:text="Add"  
   android:textSize="24sp"  
   app:layout\_constraintBottom\_toTopOf="@+id/btnUpdate"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toEndOf="@+id/btnCount" />  
    
   <Button  
   android:id="@+id/btnDelete"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginBottom="32dp"  
   android:text="Delete"  
   android:textSize="24sp"  
   app:layout\_constraintBottom\_toTopOf="@+id/btnClean"  
   app:layout\_constraintEnd\_toStartOf="@+id/btnCount"  
   app:layout\_constraintStart\_toStartOf="parent" />  
    
   <Button  
   android:id="@+id/btnClean"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginBottom="32dp"  
   android:text="Clean"  
   android:textSize="24sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toStartOf="@+id/btnCount"  
   app:layout\_constraintHorizontal\_bias="0.493"  
   app:layout\_constraintStart\_toStartOf="parent" />  
    
   <Button  
   android:id="@+id/btnCount"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginBottom="32dp"  
   android:text="Count"  
   android:textSize="24sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.498"  
   app:layout\_constraintStart\_toStartOf="parent" />  
    
   <Button  
   android:id="@+id/btnUpdate"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginBottom="32dp"  
   android:text="Update"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toEndOf="@+id/btnCount"  
   android:textSize="24sp" />  
    
  </androidx.constraintlayout.widget.ConstraintLayout>
* MainActivity.java

package com.example.lab8;  
  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.annotation.SuppressLint;  
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.widget.Button;  
import android.widget.EditText;  
  
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
 Button buttonAdd, buttonDelete, buttonClear, buttonRead, buttonUpdate;  
 EditText etName, etEmail, etID;  
 DBHelper dbHelper;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 buttonAdd = (Button) findViewById(R.id.*btnAdd*);  
 buttonAdd.setOnClickListener(this);  
  
 buttonRead = (Button) findViewById(R.id.*btnCount*);  
 buttonRead.setOnClickListener(this);  
  
 buttonClear = (Button) findViewById(R.id.*btnClean*);  
 buttonClear.setOnClickListener(this);  
  
 buttonUpdate = (Button) findViewById(R.id.*btnUpdate*);  
 buttonUpdate.setOnClickListener(this);  
  
 buttonDelete = (Button) findViewById(R.id.*btnDelete*);  
 buttonDelete.setOnClickListener(this);  
  
 etID = (EditText) findViewById(R.id.*editTextId*);  
 etName = (EditText) findViewById(R.id.*editTextName*);  
 etEmail = (EditText) findViewById(R.id.*editTextEmail*);  
  
 dbHelper = new DBHelper(this);  
  
 }  
  
 public static class DBHelper extends SQLiteOpenHelper  
 {  
 public static final int *DATABASE\_VERSION* = 1;  
 public static final String *DATABASE\_NAME* = "myBase";  
 public static final String *TABLE\_PERSONS* = "persons";  
 public static final String *KEY\_ID* = "\_id";  
 public static final String *KEY\_NAME* = "name";  
 public static final String *KEY\_MAIL* = "mail";  
  
 public DBHelper(@Nullable Context context)  
 {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db)  
 {  
 db.execSQL("create table " + *TABLE\_PERSONS* + "(" + *KEY\_ID* + " integer primary key," + *KEY\_NAME* + " text," + *KEY\_MAIL* + " text" + ")");  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)  
 {  
 db.execSQL("drop table if exists " + *TABLE\_PERSONS*);  
 onCreate(db);  
 }  
 }  
  
 @SuppressLint("SetTextI18n")  
 @Override  
 public void onClick(View v) {  
 EditText editText = (EditText) findViewById(R.id.*editTextIdButton*);  
  
 String ID = etID.getText().toString();  
 String name = etName.getText().toString();  
 String email = etEmail.getText().toString();  
  
 SQLiteDatabase database = dbHelper.getWritableDatabase();  
 ContentValues contentValues = new ContentValues(); // класс для добавления новых строк в таблицу  
  
 if (v.getId() == R.id.*btnAdd*) {  
 editText.setText(null);  
 String buttonName = getResources().getResourceEntryName(v.getId());  
 editText.setText("Нажата кнопка: " + buttonName);  
  
 contentValues.put(DBHelper.*KEY\_NAME*, name);  
 contentValues.put(DBHelper.*KEY\_MAIL*, email);  
 database.insert(DBHelper.*TABLE\_PERSONS*, null, contentValues);  
 } else if (v.getId() == R.id.*btnCount*) {  
 editText.setText(null);  
 String buttonName = getResources().getResourceEntryName(v.getId());  
 editText.setText("Нажата кнопка: " + buttonName);  
  
 Cursor cursor = database.query(DBHelper.*TABLE\_PERSONS*, null, null, null,  
 null, null, null); // все поля без сортировки и группировки  
  
 if (cursor.moveToFirst())  
 {  
 int idIndex = cursor.getColumnIndex(DBHelper.*KEY\_ID*);  
 int nameIndex = cursor.getColumnIndex(DBHelper.*KEY\_NAME*);  
 int emailIndex = cursor.getColumnIndex(DBHelper.*KEY\_MAIL*);  
 do {  
 Log.*d*("mLog", "ID =" + cursor.getInt(idIndex) +  
 ", name = " + cursor.getString(nameIndex) +  
 ", email = " + cursor.getString(emailIndex));  
  
 } while (cursor.moveToNext());  
 } else  
 Log.*d*("mLog", "0 rows");  
  
 cursor.close(); // освобождение памяти  
 } else if (v.getId() == R.id.*btnClean*) {  
 editText.setText(null);  
 String buttonName = getResources().getResourceEntryName(v.getId());  
 editText.setText("Нажата кнопка: " + buttonName);  
  
 database.delete(DBHelper.*TABLE\_PERSONS*, null, null);  
 } else if (v.getId() == R.id.*btnDelete*) {  
 editText.setText(null);  
 String buttonName = getResources().getResourceEntryName(v.getId());  
 editText.setText("Нажата кнопка: " + buttonName);  
  
 if (!ID.equalsIgnoreCase(""))  
 {  
 int delCount = database.delete(DBHelper.*TABLE\_PERSONS*, DBHelper.*KEY\_ID* + "= " + ID, null);  
 Log.*d*("mLog", "Удалено строк = " + delCount);  
 }  
 } else if (v.getId() == R.id.*btnUpdate*) {  
 editText.setText(null);  
 String buttonName = getResources().getResourceEntryName(v.getId());  
 editText.setText("Нажата кнопка: " + buttonName);  
  
 if (!ID.equalsIgnoreCase(""))  
 {  
 contentValues.put(DBHelper.*KEY\_MAIL*, email);  
 contentValues.put(DBHelper.*KEY\_NAME*, name);  
 int updCount = database.update(DBHelper.*TABLE\_PERSONS*, contentValues, DBHelper.*KEY\_ID* + "= ?", new String[]{ID});  
 Log.*d*("mLog", "Обновлено строк = " + updCount);  
 }  
 }  
 dbHelper.close(); // закрываем соединение с БД  
 }  
}

**Вывод:**

В данной лабораторной работе был рассмотрен способ сохранения, редактирования и обновления записей данных в файл базы данных, а также вывод на экран приложения этих данных во внутренней среде IDE Android Studio.