

Использование базы данных SQLite в Android Studio

Андрюенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

студент

Аннотация

Целью данной статьи является подключение и использование базы данных SQLite в Android Studio. Для создания записей в базе данных было создано приложение на языке программирования Java, сохраняющее записи пользователя. Результатом работы стало приложение, создающее контактные данные, введенные пользователем.

Ключевые слова: SQLite, база данных, Android studio

Using SQLite database in Android Studio

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to connect and use the SQLite database in Android Studio. To create records in the database, an application was created in the Java programming language that saves user records. The result of the work was an application that creates contact data entered by the user.

Keywords: SQLite, database, Android studio

1 Введение

1.1 Актуальность

Базы данных составляют в настоящее время основу компьютерного обеспечения информационных процессов, входящих практически во все сферы человеческой деятельности. База данных SQLite не исключение. SQLite является бесплатной библиотекой. Она создает автономные базы данных, не требовательные к ресурсам и не нуждающиеся в серверах, надежные и компактные. SQLite доступен на каждом устройстве с операционной системой Android, его не нужно устанавливать отдельно.

1.2 Обзор исследований

В своей работе Л.А. Березуцкая рассмотрела основные типы данных SQLite. Также были рассмотрены наиболее значимые функции SQLite [1]. М.М. Лебедев, М.В. Раскатова описали варианты работы устройства Android с локальными базами данных. Также рассматриваются различные режимы использования и функции для работы с данными. Проводятся тесты и

измерения времени выполнения операций в различных условиях [2]. А.Д. Тищенко реализовала программу-интерфейс для портативной системы анализа слов на основе словаря для операционной системы android. Программа содержит xml-разметку для общего вида отображения информации, структуру SQLite-базы и код для доступа к словарным характеристикам. [3]. В своей работе Ю.В. Мурашко рассмотрел методы хранения данных в операционной системе Android. Также рассматриваются особенности хранения данных во внутреннем и внешнем хранилищах. Выделяются достоинства и недостатки каждого из хранилищ [4]. В своей работе Изварина А.А., Маевский В.К. описали процесс подключения и создание базы данных sqlite в android studio [5].

1.3 Цель исследования

Цель исследования – подключить и создать записи с помощью базы данных SQLite.

2 Материалы и методы

Для создания записей использовалась реляционная база данных SQLite, импортированная в мобильное приложение.

3 Результаты и обсуждения

Для того, чтобы протестировать базу данных создадим небольшое приложение. Приложение будет брать данные из полей ввода, заполненные пользователем. Так же будет создано три кнопки. Первая будет отвечать за добавление записи в базу данных, вторая за извлечение записи из базы данных, а третья будет очищать всю базу данных.

Изначально создадим макет приложения, это поможет лучше ориентироваться при написании кода функционала приложения. Добавим два поля ввода для имени и email, и три кнопки (рис. 1,2).

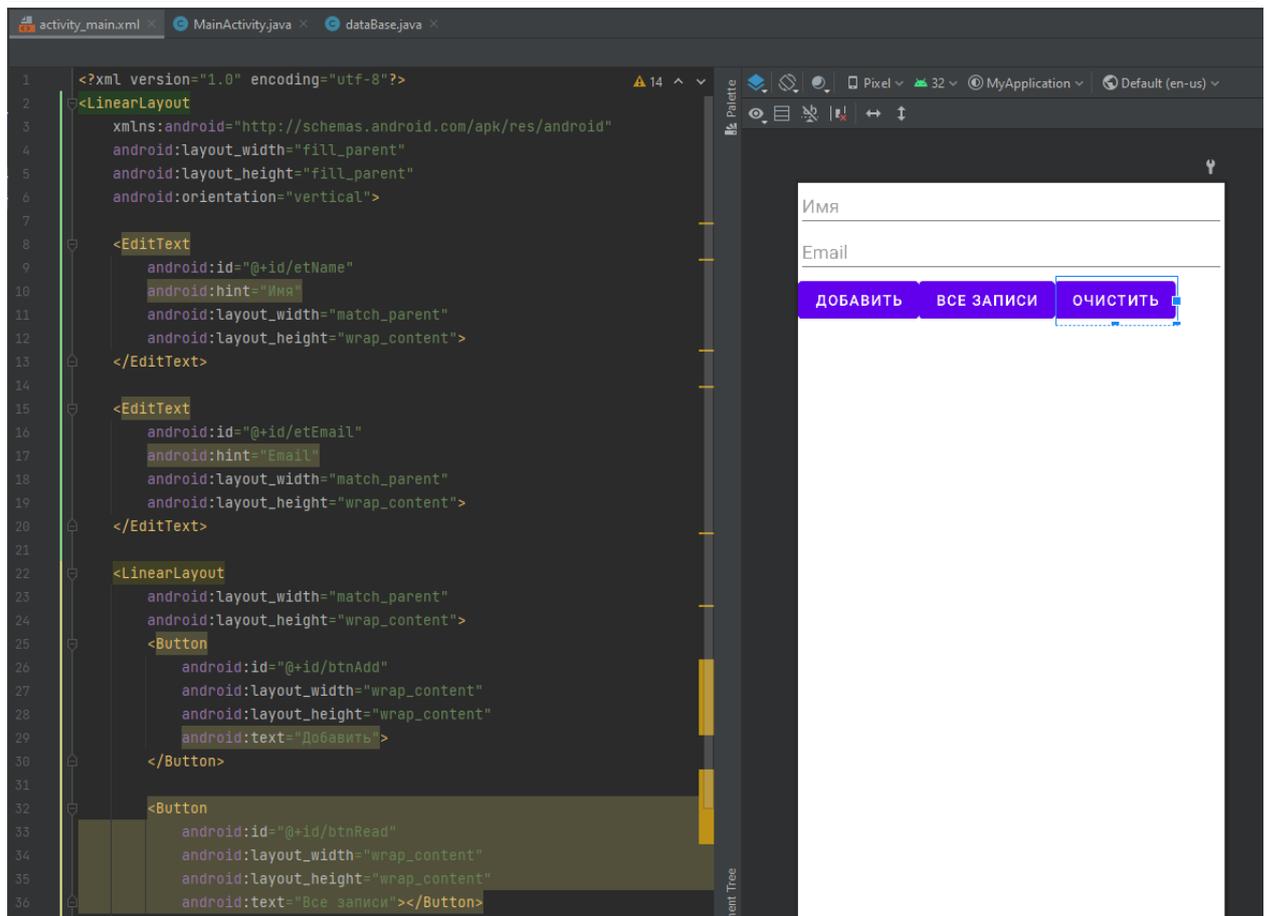


Рисунок 1 – Создание макета приложения

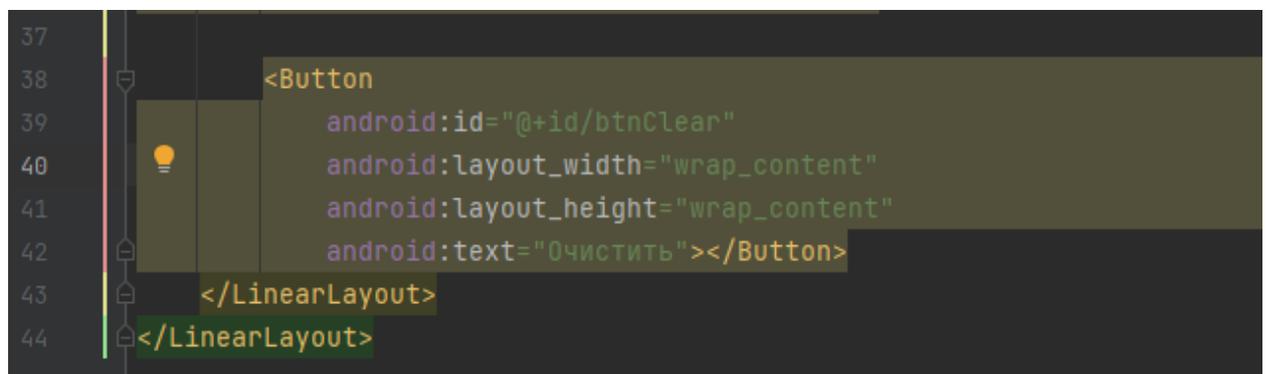


Рисунок 2 – Создание макета приложения

Далее создадим дополнительный класс для работы с базой данных. Для этого в левой панели в разделе «Project» добавляем Java.class в папку проекта. Переименуем его в «dataBase» (рис. 3).

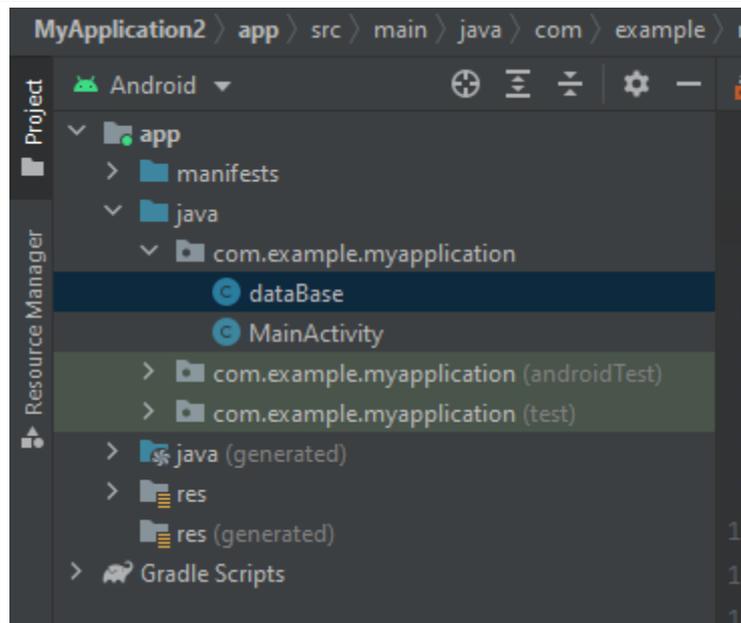


Рисунок 3 – Создание дополнительного класса «database»

Возвращаемся к основному классу. Импортируем необходимые для работы библиотеки.

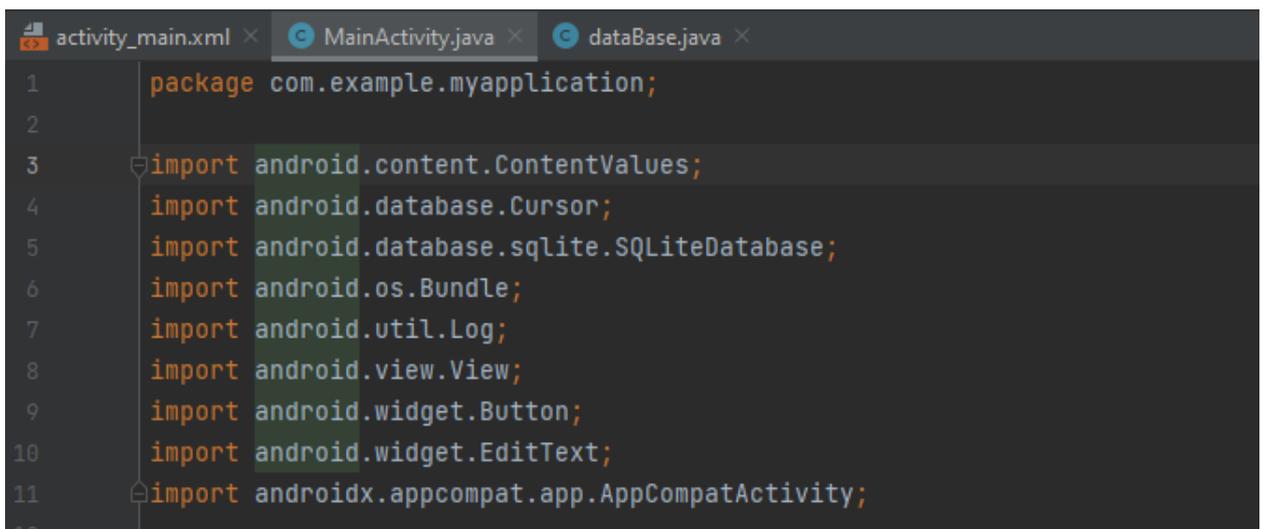


Рисунок 3 – Импорт библиотек

Далее создадим элементы экрана. Для этого пропишем кнопки и обработчики (рис. 4). Также найдем поля ввода.

```
13 public class MainActivity extends AppCompatActivity implements View.OnClickListener{
14
15     Button btnAdd, btnRead, btnClear;
16     EditText etName, etEmail;
17
18     DataBase dataBase;
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_main);
24
25         btnAdd = (Button) findViewById(R.id.btnAdd);
26         btnAdd.setOnClickListener(this);
27
28         btnRead = (Button) findViewById(R.id.btnRead);
29         btnRead.setOnClickListener(this);
30
31         btnClear = (Button) findViewById(R.id.btnClear);
32         btnClear.setOnClickListener(this);
33
34         etName = (EditText) findViewById(R.id.etName);
35         etEmail = (EditText) findViewById(R.id.etEmail);
36
37         dataBase = new DataBase( context: this);
38     }
39
```

Рисунок 4 – Создание элементов экрана

Далее необходимо в методе `onClick` создать объект класса `SQLiteDatabase`. Он предназначен для управления базой данных `SQLite` (рис. 5). Так же необходимо добавить функционал объектам на экране. Для этого пропишем в `case` пропишем код (рис.6).

```
39
40     @Override
41     public void onClick(View v) {
42
43         String name = etName.getText().toString();
44         String email = etEmail.getText().toString();
45
46         SQLiteDatabase database = dataBase.getWritableDatabase();
47
48         ContentValues contentValues = new ContentValues();
49
```

Рисунок 5 – Создание вспомогательного объекта

```

51 switch (v.getId()) {
52
53     case R.id.btnAdd:
54         contentValues.put(com.example.myapplication.dataBase.KEY_NAME, name);
55         contentValues.put(com.example.myapplication.dataBase.KEY_MAIL, email);
56
57         database.insert(com.example.myapplication.dataBase.TABLE_CONTACTS, nullColumnHack: null, contentValues);
58         break;
59
60     case R.id.btnRead:
61         Cursor cursor = database.query(com.example.myapplication.dataBase.TABLE_CONTACTS, columns: null, selection: null, selectionArgs: null,
62                                     groupBy: null, having: null, orderBy: null);
63
64         if (cursor.moveToFirst()) {
65             int idIndex = cursor.getColumnIndex(com.example.myapplication.dataBase.KEY_ID);
66             int nameIndex = cursor.getColumnIndex(com.example.myapplication.dataBase.KEY_NAME);
67             int emailIndex = cursor.getColumnIndex(com.example.myapplication.dataBase.KEY_MAIL);
68             do {
69                 Log.d(tag: "nLog", msg: "ID = " + cursor.getInt(idIndex) +
70                       ", name = " + cursor.getString(nameIndex) +
71                       ", email = " + cursor.getString(emailIndex));
72             } while (cursor.moveToNext());
73         } else
74             Log.d(tag: "nLog", msg: "0 rows");
75
76         cursor.close();
77         break;
78
79     case R.id.btnClean:
80         database.delete(com.example.myapplication.dataBase.TABLE_CONTACTS, whereClause: null, whereArgs: null);
81         break;
82 }
83 database.close();
84 }

```

Рисунок 6 – Добавления функционала для кнопок

Теперь работаем с классом `dataBase`, созданным в начале. Здесь будут созданы константы, а также работа с ними (рис.7).

```

1 package com.example.myapplication;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6
7 public class dataBase extends SQLiteOpenHelper{
8
9     public static final int DATABASE_VERSION = 1;
10    public static final String DATABASE_NAME = "contactDb";
11    public static final String TABLE_CONTACTS = "contacts";
12
13    public static final String KEY_ID = "_id";
14    public static final String KEY_NAME = "name";
15    public static final String KEY_MAIL = "mail";
16
17    public dataBase(Context context) { super(context, DATABASE_NAME, factory: null, DATABASE_VERSION); }
18
19
20
21    @Override
22    public void onCreate(SQLiteDatabase db) {
23        db.execSQL("create table " + TABLE_CONTACTS + "(" + KEY_ID
24                + " integer primary key," + KEY_NAME + " text," + KEY_MAIL + " text" + ")");
25    }
26
27
28    @Override
29    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
30        db.execSQL("drop table if exists " + TABLE_CONTACTS);
31
32        onCreate(db);
33    }
34 }
35 }

```

Рисунок 7 – Работа с классом `database`

Теперь необходимо проверить созданное приложение. Создадим пару записей (рис. 8).

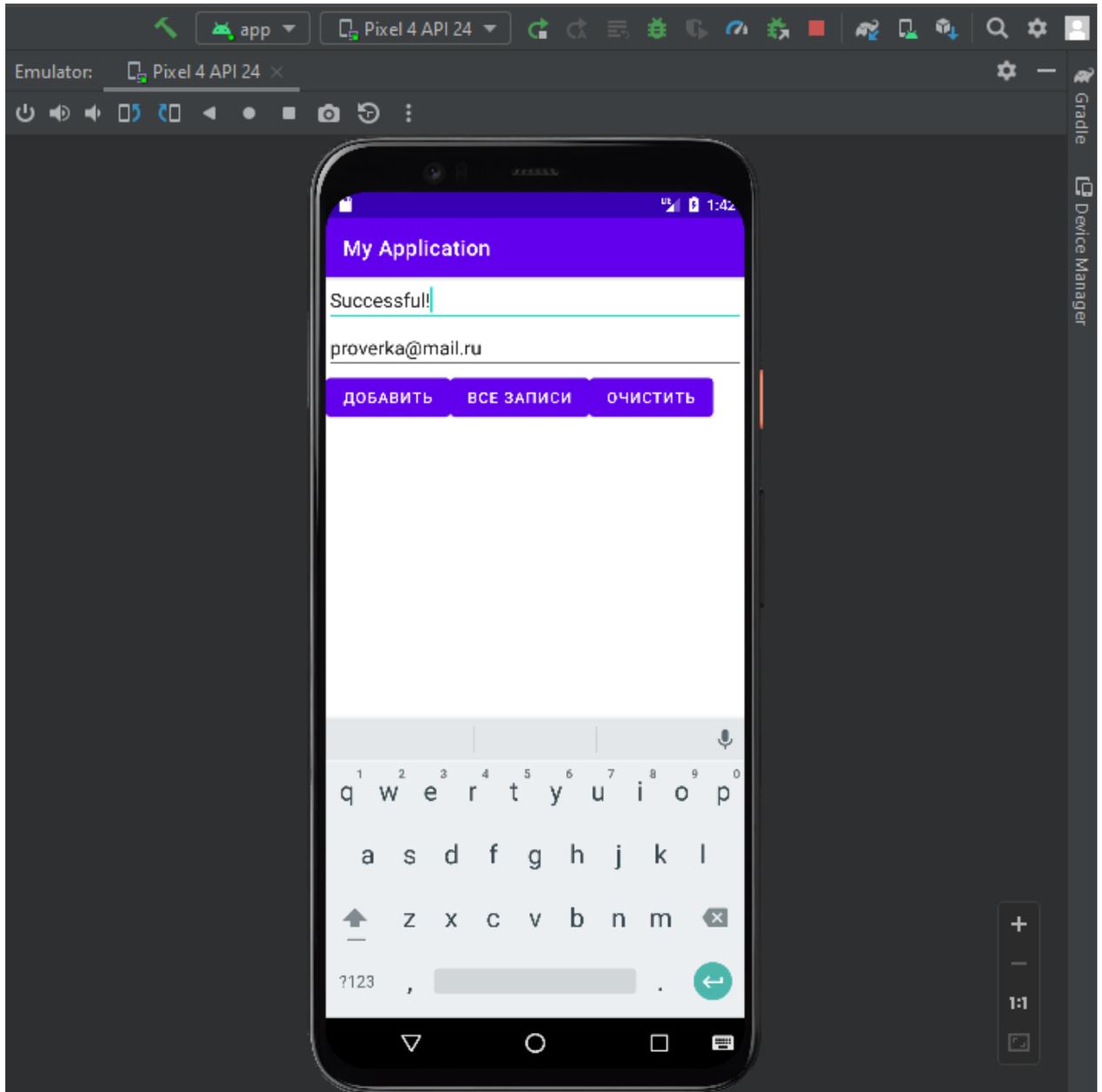
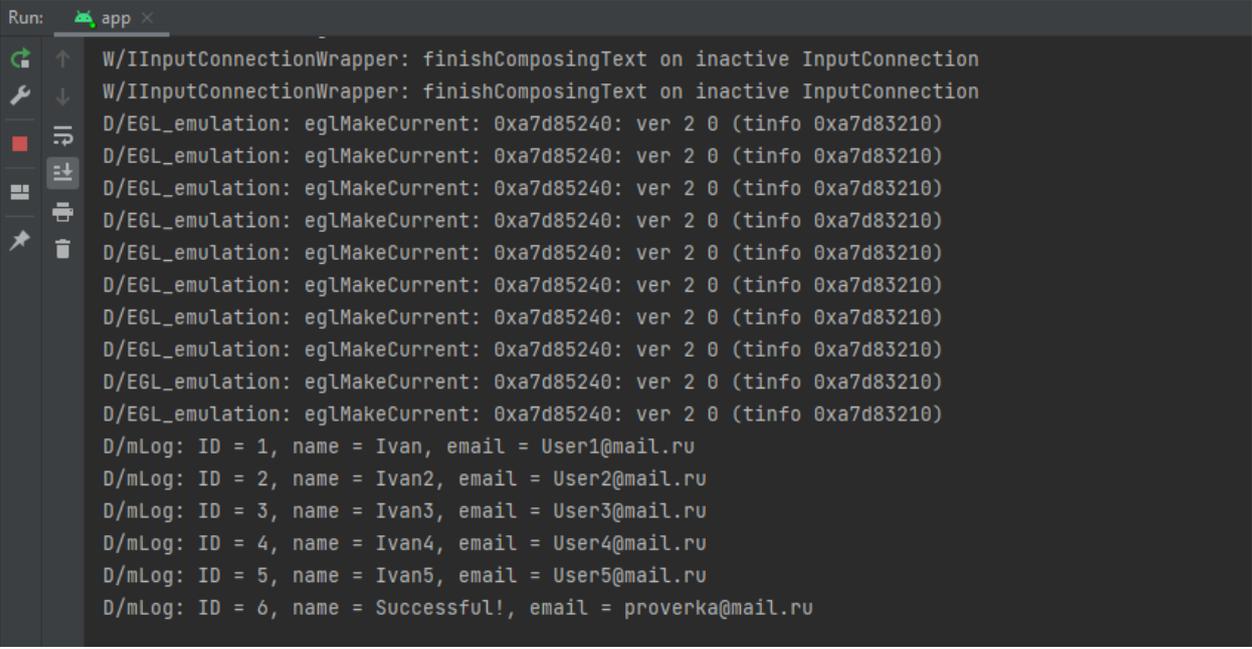


Рисунок 8 – Добавление записей

Теперь выведем все добавленные записи в консоль, с помощью кнопки все записи (рис. 9).

The image shows a screenshot of the Android Studio interface, specifically the Logcat window. The window title is "Run: app". The log output is as follows:

```
W/IInputConnectionWrapper: finishComposingText on inactive InputConnection
W/IInputConnectionWrapper: finishComposingText on inactive InputConnection
D/EGL_emulation: eglMakeCurrent: 0xa7d85240: ver 2 0 (tinfo 0xa7d83210)
D/mLog: ID = 1, name = Ivan, email = User1@mail.ru
D/mLog: ID = 2, name = Ivan2, email = User2@mail.ru
D/mLog: ID = 3, name = Ivan3, email = User3@mail.ru
D/mLog: ID = 4, name = Ivan4, email = User4@mail.ru
D/mLog: ID = 5, name = Ivan5, email = User5@mail.ru
D/mLog: ID = 6, name = Successful!, email = proverka@mail.ru
```

Рисунок 9 – Вывод добавленных записей

Выводы

В данной работе было подключена и использована база данных SQLite. Итогом работы стало работающее мобильное приложение, сохраняющая имя и email введенный пользователем. Также в приложение добавлена возможность просмотра и очистка списка добавленных данных.

Библиографический список

1. Березуцкая Л.А. SQLite main commands and their comparison to sql for an android app. // LinguaNet. Сборник материалов Всероссийской молодежной научно-практической конференции с международным участием «LinguaNet». под общ. ред. Ю.А. Иванцовой, Н.С. Руденко. 2019. С. 289-293.
2. Лебедев М.М., Раскатова М.В. Оптимизация скорости работы android устройств с локальной базой данных. // Международный журнал информационных технологий и энергоэффективности. 2020. Т. 5. № 2 (16). С. 3-21.
3. Тищенко А.Д. Программный интерфейс для системы словарного анализа для android – приложений. // Свидетельство о регистрации программы для ЭВМ RU 2015617959, 27.07.2015. Заявка № 2015614917 от 04.06.2015.
4. Мурашко Ю.В. Методы хранения данных в ос android. // Решение. 2019. Т. 1. С. 144-146.
5. Изварина А.А., Маевский В.К. Разработка android приложения с встроенной базой данных. // Семьдесят четвертая всероссийская научно-техническая конференция студентов, магистрантов и аспирантов высших учебных заведений с международным участием. Сборник материалов конференции. Ярославль, 2021. С. 692-695.