Министерство науки и высшего образования Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

По дисциплине: «Разработка мобильных приложений»

Лабораторная работа №5

«Исследование взаимодействия android-приложения с базой данных»

Выполнил

ст. гр. ИС/б-17-2о

Волобуев Ю.С.

Проверил:

Шишкевич В.Е.

Севастополь

2020

1 ЦЕЛЬ

Освоить взаимодействие Android-приложения с SQLite.

2 ВАРИАНТ ЗАДАНИЯ

Вариант 9:

|  |  |
| --- | --- |
| № варианта | Задание |
| 9 |  |

3 ХОД РАБОТЫ

1. Листинг программы:

MainActivity.java:

package com.example.databaseapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentValues;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.RelativeLayout;

import android.widget.TextView;

import static com.example.databaseapplication.DBHelper.KIND\_PRODUCT\_TABLE\_NAME;

import static com.example.databaseapplication.DBHelper.MANUFACTURER\_TABLE\_NAME;

import static com.example.databaseapplication.DBHelper.PRODUCT\_TABLE\_NAME;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

final String LOG\_TAG = "myLogs";

private Button btnTableManuf, btnTableKindProducts, btnTableProducts, btnBack1, btnBack2, btnBack3;

private TextView manLOG, manReadLOG, kindProdLOG, kindProdReadLOG, prodLOG, prodReadLOG;

private Button btnManUpdate;

private Button btnManAdd;

private Button btnManRead;

private Button btnManClear;

private Button btnManDelete;

private EditText manEtID, manEtName, manEtCountry;

private Button btnKindProdAdd, btnKindProdRead, btnKindProdClear, btnKindProdUpdate, btnKindProdDelete;

private EditText kindProdEtID, kindProdEtName;

private Button btnProdAdd, btnProdRead, btnProdClear, btnProdUpdate, btnProdDelete;

private EditText prodEtID, prodEtName, prodEtPackCount, prodEtPrice, prodEtReserve, prodEtManID, prodEtKindPrID;

private RelativeLayout relativeLayoutBase, relativeLayoutMan, relativeLayoutKindProd, relativeLayoutProducts;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

btnTableManuf = (Button) findViewById(R.id.manufacturers);

btnTableKindProducts = (Button) findViewById(R.id.kindProducts);

btnTableProducts = (Button) findViewById(R.id.products);

btnBack1 = (Button) findViewById(R.id.Back);

btnBack2 = (Button) findViewById(R.id.Back2);

btnBack3 = (Button) findViewById(R.id.Back3);

manLOG = (TextView) findViewById(R.id.LOG);

manReadLOG = (TextView) findViewById(R.id.readLOG);

kindProdLOG = (TextView) findViewById(R.id.LOG2);

kindProdReadLOG = (TextView) findViewById(R.id.readLOG2);

prodLOG = (TextView) findViewById(R.id.LOG3);

prodReadLOG = (TextView) findViewById(R.id.readLOG3);

//------Manufacturer------

btnManUpdate = (Button) findViewById(R.id.ManUpdate);

btnManUpdate.setOnClickListener(this);

btnManAdd = (Button) findViewById(R.id.ManAdd);

btnManAdd.setOnClickListener(this);

btnManRead = (Button) findViewById(R.id.ManRead);

btnManRead.setOnClickListener(this);

btnManClear = (Button) findViewById(R.id.ManClear);

btnManClear.setOnClickListener(this);

btnManDelete = (Button) findViewById(R.id.ManDelete);

btnManDelete.setOnClickListener(this);

manEtID = (EditText) findViewById(R.id.ManEtID);

manEtName = (EditText) findViewById(R.id.ManEtName);

manEtCountry = (EditText) findViewById(R.id.ManEtCountry);

//------Kind Products------

btnKindProdAdd = (Button) findViewById(R.id.KindProdAdd);

btnKindProdAdd.setOnClickListener(this);

btnKindProdRead = (Button) findViewById(R.id.KindProdRead);

btnKindProdRead.setOnClickListener(this);

btnKindProdClear = (Button) findViewById(R.id.KindProdClear);

btnKindProdClear.setOnClickListener(this);

btnKindProdUpdate = (Button) findViewById(R.id.KindProdUpdate);

btnKindProdUpdate.setOnClickListener(this);

btnKindProdDelete = (Button) findViewById(R.id.KindProdDelete);

btnKindProdDelete.setOnClickListener(this);

kindProdEtID = (EditText) findViewById(R.id.KindProdEtID);

kindProdEtName = (EditText) findViewById(R.id.KindProdEtName);

//------Product------

btnProdUpdate = (Button) findViewById(R.id.ProdUpdate);

btnProdUpdate.setOnClickListener(this);

btnProdAdd = (Button) findViewById(R.id.ProdAdd);

btnProdAdd.setOnClickListener(this);

btnProdRead = (Button) findViewById(R.id.ProdRead);

btnProdRead.setOnClickListener(this);

btnProdClear = (Button) findViewById(R.id.ProdClear);

btnProdClear.setOnClickListener(this);

btnProdDelete = (Button) findViewById(R.id.ProdDelete);

btnProdDelete.setOnClickListener(this);

prodEtID = (EditText) findViewById(R.id.ProdEtID);

prodEtName = (EditText) findViewById(R.id.ProdEtName);

prodEtPackCount = (EditText) findViewById(R.id.ProdEtPackCount);

prodEtPrice = (EditText) findViewById(R.id.ProdEtPrice);

prodEtReserve = (EditText) findViewById(R.id.ProdEtReserve);

prodEtManID = (EditText) findViewById(R.id.ProdEtManID);

prodEtKindPrID = (EditText) findViewById(R.id.ProdEtKindPrID);

relativeLayoutBase = (RelativeLayout) findViewById(R.id.RelativeLayoutBase);

relativeLayoutMan = (RelativeLayout) findViewById(R.id.RelativeLayoutMan);

relativeLayoutKindProd = (RelativeLayout) findViewById(R.id.RelativeLayoutKindProd);

relativeLayoutProducts = (RelativeLayout) findViewById(R.id.RelativeLayoutProducts);

}

public void WorkDB(View v) {

if(btnTableManuf.isPressed()){

btnTableManuf.setVisibility(View.INVISIBLE);

btnTableKindProducts.setVisibility(View.INVISIBLE);

btnTableProducts.setVisibility(View.INVISIBLE);

relativeLayoutMan.setVisibility(View.VISIBLE);

}else

if(btnTableKindProducts.isPressed()){

btnTableManuf.setVisibility(View.INVISIBLE);

btnTableKindProducts.setVisibility(View.INVISIBLE);

btnTableProducts.setVisibility(View.INVISIBLE);

relativeLayoutKindProd.setVisibility(View.VISIBLE);

}else if(btnTableProducts.isPressed()){

btnTableManuf.setVisibility(View.INVISIBLE);

btnTableKindProducts.setVisibility(View.INVISIBLE);

btnTableProducts.setVisibility(View.INVISIBLE);

relativeLayoutProducts.setVisibility(View.VISIBLE);

}

}

public void Back(View v){

if(btnBack1.isPressed() || btnBack2.isPressed() || btnBack3.isPressed()){

relativeLayoutMan.setVisibility(View.INVISIBLE);

relativeLayoutKindProd.setVisibility(View.INVISIBLE);

relativeLayoutProducts.setVisibility(View.INVISIBLE);

btnTableManuf.setVisibility(View.VISIBLE);

btnTableKindProducts.setVisibility(View.VISIBLE);

btnTableProducts.setVisibility(View.VISIBLE);

}

}

public void onClick(View v) {

ContentValues cv = new ContentValues();

String idMan = manEtID.getText().toString();

String nameMan = manEtName.getText().toString();

String countryMan = manEtCountry.getText().toString();

String idKindProd = kindProdEtID.getText().toString();

String nameKindProd = kindProdEtName.getText().toString();

String idProd = prodEtID.getText().toString();

String nameProd = prodEtName.getText().toString();

String packCountProd = prodEtPackCount.getText().toString();

String priceProd = prodEtPrice.getText().toString();

String reserveProd = prodEtReserve.getText().toString();

String manIdProd = prodEtManID.getText().toString();

String kindPrIdProd = prodEtKindPrID.getText().toString();

String data, title;

DBHelper dbHelper = new DBHelper(this);

SQLiteDatabase db;

db = dbHelper.getWritableDatabase();

Cursor cursor;

switch (v.getId()) {

case R.id.ManAdd:

cv.clear();

title = "--- Insert in table manufacturers: ---";

manLOG.setText(title);

cv.put(DBHelper.MANUFACTURER\_COLUMN\_NAME, nameMan);

cv.put(DBHelper.MANUFACTURER\_COLUMN\_COUNTRY, countryMan);

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

data = "row inserted, ID = " + rowID;

manReadLOG.setText(data);

break;

case R.id.ManRead:

data = "";

title = "--- Rows in table manufacturers: ---";

manLOG.setText(title);

cursor = db.query(MANUFACTURER\_TABLE\_NAME, null, null, null, null, null, null);

if (cursor.moveToFirst()) {

int idColIndex = cursor.getColumnIndex(DBHelper.MANUFACTURER\_COLUMN\_ID);

int nameColIndex = cursor.getColumnIndex(DBHelper.MANUFACTURER\_COLUMN\_NAME);

int countryColIndex = cursor.getColumnIndex(DBHelper.MANUFACTURER\_COLUMN\_COUNTRY);

do {

data += "ID = " + cursor.getInt(idColIndex) + ", name = "

+ cursor.getString(nameColIndex) + ", country = "

+ cursor.getString(countryColIndex) + "\n";

manReadLOG.setText(data);

} while (cursor.moveToNext());

} else {

data = "0 rows";

manReadLOG.setText(data);

}

cursor.close();

break;

case R.id.ManClear:

title = "--- Clear table manufacturers: ---";

manLOG.setText(title);

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

data = "deleted rows count = " + clearCount;

manReadLOG.setText(data);

break;

case R.id.ManUpdate:

if (idMan.equalsIgnoreCase("")) {

break;

}

title = "--- Update table manufacturers: ---";

manLOG.setText(title);

cv.put(DBHelper.MANUFACTURER\_COLUMN\_NAME, nameMan);

cv.put(DBHelper.MANUFACTURER\_COLUMN\_COUNTRY, countryMan);

int updCount = db.update(MANUFACTURER\_TABLE\_NAME, cv, DBHelper.MANUFACTURER\_COLUMN\_ID + " = ?",

new String[]{idMan});

data = "updated rows count = " + updCount;

manReadLOG.setText(data);

break;

case R.id.ManDelete:

if (idMan.equalsIgnoreCase("")) {

break;

}

title = "--- Delete row in table manufacturers: ---";

manLOG.setText(title);

int delCount = db.delete(MANUFACTURER\_TABLE\_NAME, DBHelper.MANUFACTURER\_COLUMN\_ID + " = " + idMan, null);

data = "deleted rows count = " + delCount;

manReadLOG.setText(data);

break;

case R.id.KindProdAdd:

cv.clear();

title = "--- Insert in table kind of products: ---";

kindProdLOG.setText(title);

cv.put(DBHelper.KIND\_PRODUCT\_COLUMN\_NAME, nameKindProd);

long rowId = db.insert(KIND\_PRODUCT\_TABLE\_NAME, null, cv);

data = "row inserted, ID = " + rowId;

kindProdReadLOG.setText(data);

break;

case R.id.KindProdRead:

data = "";

title = "--- Rows in table kind of products: ---";

kindProdLOG.setText(title);

cursor = db.query(KIND\_PRODUCT\_TABLE\_NAME, null, null, null, null, null, null);

if (cursor.moveToFirst()) {

int idColIndex = cursor.getColumnIndex(DBHelper.KIND\_PRODUCT\_COLUMN\_ID);

int nameColIndex = cursor.getColumnIndex(DBHelper.KIND\_PRODUCT\_COLUMN\_NAME);

do {

data += "ID = " + cursor.getInt(idColIndex) + ", name = "

+ cursor.getString(nameColIndex) + "\n";

kindProdReadLOG.setText(data);

} while (cursor.moveToNext());

} else {

data = "0 rows";

kindProdReadLOG.setText(data);

}

cursor.close();

break;

case R.id.KindProdClear:

title = "--- Clear table kind of products: ---";

kindProdLOG.setText(title);

int clearCountRow = db.delete(KIND\_PRODUCT\_TABLE\_NAME, null, null);

data = "deleted rows count = " + clearCountRow;

kindProdReadLOG.setText(data);

break;

case R.id.KindProdUpdate:

if (idKindProd.equalsIgnoreCase("")) {

break;

}

title = "--- Update table kind of products: ---";

kindProdLOG.setText(title);

cv.put(DBHelper.KIND\_PRODUCT\_COLUMN\_NAME, nameKindProd);

int updCountRow = db.update(KIND\_PRODUCT\_TABLE\_NAME, cv, DBHelper.KIND\_PRODUCT\_COLUMN\_ID + " = ?",

new String[]{idKindProd});

data = "updated rows count = " + updCountRow;

kindProdReadLOG.setText(data);

break;

case R.id.KindProdDelete:

if (idKindProd.equalsIgnoreCase("")) {

break;

}

title = "--- Delete row in table products: ---";

kindProdLOG.setText(title);

int DelCount = db.delete(KIND\_PRODUCT\_TABLE\_NAME, DBHelper.KIND\_PRODUCT\_COLUMN\_ID + " = " + idKindProd, null);

data = "deleted rows count = " + DelCount;

kindProdReadLOG.setText(data);

break;

case R.id.ProdAdd:

cv.clear();

title = "--- Insert in table products: ---";

prodLOG.setText(title);

cv.put(DBHelper.PRODUCT\_COLUMN\_NAME, nameProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_PACKAGE\_COUNT, packCountProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_PRICE, priceProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_RESERVE, reserveProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_MANUFACTURER\_ID, manIdProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_KIND\_PRODUCT\_ID, kindPrIdProd);

long row = db.insert(PRODUCT\_TABLE\_NAME, null, cv);

data = "row inserted, ID = " + row;

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

prodReadLOG.setText(data);

break;

case R.id.ProdRead:

data = "";

title = "--- Rows in table products: ---";

prodLOG.setText(title);

cursor = db.query(PRODUCT\_TABLE\_NAME, null, null, null, null, null, null);

if (cursor.moveToFirst()) {

int idColIndex = cursor.getColumnIndex(DBHelper.PRODUCT\_COLUMN\_ID);

int nameColIndex = cursor.getColumnIndex(DBHelper.PRODUCT\_COLUMN\_NAME);

int packageCountColIndex = cursor.getColumnIndex(DBHelper.PRODUCT\_COLUMN\_PACKAGE\_COUNT);

int priceColIndex = cursor.getColumnIndex(DBHelper.PRODUCT\_COLUMN\_PRICE);

int reserveColIndex = cursor.getColumnIndex(DBHelper.PRODUCT\_COLUMN\_RESERVE);

int manIDColIndex = cursor.getColumnIndex(DBHelper.PRODUCT\_COLUMN\_MANUFACTURER\_ID);

int kindPrIDColIndex = cursor.getColumnIndex(DBHelper.PRODUCT\_COLUMN\_KIND\_PRODUCT\_ID);

do {

data += "ID = " + cursor.getInt(idColIndex) + ", name = " + cursor.getString(nameColIndex) +

", package\_count = " + cursor.getString(packageCountColIndex) + ", price = " + cursor.getString(priceColIndex) +

", reserve = " + cursor.getString(reserveColIndex) + ", manufacturer id = " + cursor.getString(manIDColIndex) +

", kind product id = " + cursor.getString(kindPrIDColIndex) + "\n";

prodReadLOG.setText(data);

} while (cursor.moveToNext());

} else {

data = "0 rows";

prodReadLOG.setText(data);

}

cursor.close();

break;

case R.id.ProdClear:

title = "--- Clear table products: ---";

prodLOG.setText(title);

int clearRowCount = db.delete(PRODUCT\_TABLE\_NAME, null, null);

data = "deleted rows count = " + clearRowCount;

prodReadLOG.setText(data);

break;

case R.id.ProdUpdate:

if (idProd.equalsIgnoreCase("")) {

break;

}

title = "--- Update table products: ---";

prodLOG.setText(title);

cv.put(DBHelper.PRODUCT\_COLUMN\_NAME, nameProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_PACKAGE\_COUNT, packCountProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_PRICE, priceProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_RESERVE, reserveProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_MANUFACTURER\_ID, manIdProd);

cv.put(DBHelper.PRODUCT\_COLUMN\_KIND\_PRODUCT\_ID, kindPrIdProd);

int updateCountRow = db.update(PRODUCT\_TABLE\_NAME, cv, DBHelper.PRODUCT\_COLUMN\_KIND\_PRODUCT\_ID + " = ?",

new String[]{idProd});

data = "updated rows count = " + updateCountRow;

Log.d(LOG\_TAG, "updated rows count = " + updateCountRow);

prodReadLOG.setText(data);

break;

case R.id.ProdDelete:

if (idProd.equalsIgnoreCase("")) {

break;

}

title = "--- Delete row in table products: ---";

prodLOG.setText(title);

int deleteCount = db.delete(PRODUCT\_TABLE\_NAME, DBHelper.PRODUCT\_COLUMN\_KIND\_PRODUCT\_ID + " = " + idProd, null);

data = "deleted rows count = " + deleteCount;

prodReadLOG.setText(data);

break;

}

dbHelper.close();

}

}

В методе onCreate() мы определяем объекты и присваиваем обработчики на все необходимые кнопки.

В методе onClick() мы создаём объект dbHelper класса DBHelper для управления БД, а также обрабатываем события нажатия на кнопки, инициирующие добавление, удаление, чтение и обновление данных из БД.

С помощью метода getWritableDatabase производим подключение к БД и получаем объект SQLiteDatabase, позволяющий нам работать с БД. При обработке кнопок добавления, чтения и удаления мы используем методы этого объекта соответственно insert – вставка записи, query – чтение, delete – удаление.

Класс DBHelper, наследующий SQLiteOpenHelper, в котором мы прописываем логику работы с БД, а именно создание таблиц в методе onCreate(), обновление базы, если версия оказалась выше предыдущей в методе onUpgrade(), или аналогично наоборот, если версия старее нынешней (что случается реже) используем метод onDowngrade(). В конструкторе данного класса вызываем конструктор супер-класса и указываем имя и версию БД.

Листинг DBHelper.java:

package com.example.databaseapplication;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class DBHelper extends SQLiteOpenHelper {

public static final int DATABASE\_VERSION = 1;

public static final String DATABASE\_NAME = "myDB";

public static final String PRODUCT\_TABLE\_NAME = "products";

public static final String PRODUCT\_COLUMN\_ID = "id";

public static final String PRODUCT\_COLUMN\_NAME = "name";

public static final String PRODUCT\_COLUMN\_PACKAGE\_COUNT = "package\_count";

public static final String PRODUCT\_COLUMN\_PRICE = "price";

public static final String PRODUCT\_COLUMN\_RESERVE = "reserve";

public static final String PRODUCT\_COLUMN\_MANUFACTURER\_ID = "manufacturer\_id";

public static final String PRODUCT\_COLUMN\_KIND\_PRODUCT\_ID = "kind\_product\_id";

public static final String MANUFACTURER\_TABLE\_NAME = "manufacturers";

public static final String MANUFACTURER\_COLUMN\_ID = "id";

public static final String MANUFACTURER\_COLUMN\_NAME = "name";

public static final String MANUFACTURER\_COLUMN\_COUNTRY = "country";

public static final String KIND\_PRODUCT\_TABLE\_NAME = "kind\_products";

public static final String KIND\_PRODUCT\_COLUMN\_ID = "\_id";

public static final String KIND\_PRODUCT\_COLUMN\_NAME = "\_name";

public DBHelper(Context context) {

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

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("CREATE TABLE " + MANUFACTURER\_TABLE\_NAME + " (id integer primary key, name text, country text); ");

db.execSQL("CREATE TABLE " + KIND\_PRODUCT\_TABLE\_NAME + " (\_id integer primary key, \_name text); ");

db.execSQL(

"CREATE TABLE " + PRODUCT\_TABLE\_NAME +

" (id integer primary key, name text, package\_count integer, " +

"price real CHECK (price > 0), reserve integer, " +

"manufacturer\_id integer, kind\_product\_id integer, " +

"FOREIGN KEY (manufacturer\_id) REFERENCES " + MANUFACTURER\_TABLE\_NAME + "(id) ON DELETE CASCADE ON UPDATE CASCADE, " +

"FOREIGN KEY (kind\_product\_id) REFERENCES " + KIND\_PRODUCT\_TABLE\_NAME + "(\_id) ON DELETE CASCADE ON UPDATE CASCADE); ");

}

@Override

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

db.execSQL("DROP TABLE IF EXISTS " + MANUFACTURER\_TABLE\_NAME);

db.execSQL("DROP TABLE IF EXISTS " + KIND\_PRODUCT\_TABLE\_NAME);

db.execSQL("DROP TABLE IF EXISTS " + PRODUCT\_TABLE\_NAME);

onCreate(db);

}

@Override

public void onDowngrade(SQLiteDatabase db, int OldVersion, int newVersion){

db.execSQL("DROP TABLE IF EXISTS " + MANUFACTURER\_TABLE\_NAME);

db.execSQL("DROP TABLE IF EXISTS " + KIND\_PRODUCT\_TABLE\_NAME);

db.execSQL("DROP TABLE IF EXISTS " + PRODUCT\_TABLE\_NAME);

onCreate(db);

}

}

Разметка приложения описана в activity\_main.xml:

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

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

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/RelativeLayoutBase"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:visibility="visible"

tools:context=".MainActivity">

<Button

android:id="@+id/manufacturers"

style="@style/ButtonBase"

android:text="MANUFACTURERS" />

<Button

android:id="@+id/kindProducts"

style="@style/ButtonBase"

android:layout\_below="@id/manufacturers"

android:text="KINDS OF PRODUCTS" />

<Button

android:id="@+id/products"

style="@style/ButtonBase"

android:layout\_below="@id/kindProducts"

android:text="PRODUCTS" />

<RelativeLayout

android:id="@+id/RelativeLayoutMan"

style="@style/RelativeLayout"

tools:context=".MainActivity">

<LinearLayout style="@style/LinearLayout">

<TextView

android:id="@+id/ManID"

style="@style/TextView"

android:text="ID: " />

<EditText

android:id="@+id/ManEtID"

android:layout\_width="100sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp" />

<Button

android:id="@+id/ManUpdate"

android:layout\_marginBottom="10dp"

android:text="Update"

style="@style/ButtonDB" />

</LinearLayout>

<LinearLayout

style="@style/LinearLayout"

android:layout\_marginTop="70dp">

<TextView

android:id="@+id/ManName"

style="@style/TextView"

android:text="Name: " />

<EditText

android:id="@+id/ManEtName"

android:layout\_width="200sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp"

android:width="70dp" />

</LinearLayout>

<LinearLayout

style="@style/LinearLayout"

android:layout\_marginTop="140dp">

<TextView

android:id="@+id/ManCountry"

style="@style/TextView"

android:text="Country: " />

<EditText

android:id="@+id/ManEtCountry"

android:layout\_width="220sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp"

android:width="70dp" />

</LinearLayout>

<LinearLayout

android:id="@+id/LL"

style="@style/LinearLayout"

android:layout\_marginTop="210dp">

<Button

android:id="@+id/ManAdd"

android:text="Add"

style="@style/ButtonDB" />

<Button

android:id="@+id/ManRead"

android:text="Read"

style="@style/ButtonDB" />

<Button

android:id="@+id/ManClear"

android:text="Clear"

style="@style/ButtonDB" />

<Button

android:id="@+id/ManDelete"

android:text="Delete"

style="@style/ButtonDB" />

</LinearLayout>

<Button

android:id="@+id/Back"

android:layout\_alignTop="@id/LL"

style="@style/ButtonBack" />

<TextView

android:id="@+id/LOG"

style="@style/EditTextTitle" />

<TextView

android:id="@+id/readLOG"

android:layout\_below="@id/LOG"

style="@style/EditTextLOG" />

</RelativeLayout>

<RelativeLayout

android:id="@+id/RelativeLayoutKindProd"

style="@style/RelativeLayout"

tools:context=".MainActivity">

<LinearLayout style="@style/LinearLayout">

<TextView

android:id="@+id/KindProdID"

style="@style/TextView"

android:text="ID: " />

<EditText

android:id="@+id/KindProdEtID"

android:layout\_width="100sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp" />

<Button

android:id="@+id/KindProdUpdate"

android:layout\_marginBottom="10dp"

android:text="Update"

style="@style/ButtonDB" />

</LinearLayout>

<LinearLayout

style="@style/LinearLayout"

android:layout\_marginTop="70dp">

<TextView

android:id="@+id/KindProdName"

android:text="Name: "

style="@style/TextView" />

<EditText

android:id="@+id/KindProdEtName"

android:layout\_width="200sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp"

android:width="70dp" />

</LinearLayout>

<LinearLayout

android:id="@+id/LL2"

style="@style/LinearLayout"

android:layout\_marginTop="140dp">

<Button

android:id="@+id/KindProdAdd"

android:text="Add"

style="@style/ButtonDB" />

<Button

android:id="@+id/KindProdRead"

android:text="Read"

style="@style/ButtonDB" />

<Button

android:id="@+id/KindProdClear"

android:text="Clear"

style="@style/ButtonDB" />

<Button

android:id="@+id/KindProdDelete"

android:text="Delete"

style="@style/ButtonDB" />

</LinearLayout>

<Button

android:id="@+id/Back2"

android:layout\_alignTop="@id/LL2"

style="@style/ButtonBack" />

<TextView

android:id="@+id/LOG2"

style="@style/EditTextTitle" />

<TextView

android:id="@+id/readLOG2"

android:layout\_below="@id/LOG2"

style="@style/EditTextLOG" />

</RelativeLayout>

<RelativeLayout

android:id="@+id/RelativeLayoutProducts"

style="@style/RelativeLayout"

tools:context=".MainActivity">

<LinearLayout

android:id="@+id/l1"

style="@style/LinearLayout">

<TextView

android:id="@+id/ProdID"

android:text="ID: "

style="@style/TextView" />

<EditText

android:id="@+id/ProdEtID"

android:layout\_width="100sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp" />

<Button

android:id="@+id/ProdUpdate"

android:text="Update"

style="@style/ButtonDB" />

</LinearLayout>

<LinearLayout

android:id="@+id/l2"

style="@style/LinearLayout"

android:layout\_below="@id/l1">

<TextView

android:id="@+id/ProdName"

android:text="Name: "

style="@style/TextView" />

<EditText

android:id="@+id/ProdEtName"

android:layout\_width="200sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp"

android:width="70dp" />

</LinearLayout>

<LinearLayout

android:id="@+id/l3"

style="@style/LinearLayout"

android:layout\_below="@id/l2">

<TextView

android:id="@+id/ProdPackageCount"

android:text="Package count: "

style="@style/TextView" />

<EditText

android:id="@+id/ProdEtPackCount"

android:layout\_width="110sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp"

android:width="70dp" />

</LinearLayout>

<LinearLayout

android:id="@+id/l4"

style="@style/LinearLayout"

android:layout\_below="@id/l3">

<TextView

android:id="@+id/ProdPrice"

android:text="Price: "

style="@style/TextView" />

<EditText

android:id="@+id/ProdEtPrice"

android:layout\_width="110sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp"

android:width="70dp" />

</LinearLayout>

<LinearLayout

android:id="@+id/l5"

style="@style/LinearLayout"

android:layout\_below="@id/l4">

<TextView

android:id="@+id/ProdReserve"

android:text="Reserve: "

style="@style/TextView" />

<EditText

android:id="@+id/ProdEtReserve"

android:layout\_width="110sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp"

android:width="70dp" />

</LinearLayout>

<LinearLayout

android:id="@+id/l6"

style="@style/LinearLayout"

android:layout\_below="@id/l5">

<TextView

android:id="@+id/ProdManID"

android:text="Manufacturer id: "

style="@style/TextView" />

<EditText

android:id="@+id/ProdEtManID"

android:layout\_width="110sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp"

android:width="70dp" />

</LinearLayout>

<LinearLayout

android:id="@+id/l7"

style="@style/LinearLayout"

android:layout\_below="@id/l6">

<TextView

android:id="@+id/ProdKindPrID"

android:text="Kind product id: "

style="@style/TextView" />

<EditText

android:id="@+id/ProdEtKindPrID"

android:layout\_width="110sp"

android:layout\_height="wrap\_content"

android:layout\_marginRight="25dp"

android:width="70dp" />

</LinearLayout>

<LinearLayout

android:id="@+id/LL3"

style="@style/LinearLayout"

android:layout\_below="@id/l7">

<Button

android:id="@+id/ProdAdd"

android:text="Add"

style="@style/ButtonDB" />

<Button

android:id="@+id/ProdRead"

android:text="Read"

style="@style/ButtonDB" />

<Button

android:id="@+id/ProdClear"

android:text="Clear"

style="@style/ButtonDB" />

<Button

android:id="@+id/ProdDelete"

android:text="Delete"

style="@style/ButtonDB" />

</LinearLayout>

<Button

android:id="@+id/Back3"

android:layout\_alignTop="@id/LL3"

style="@style/ButtonBack" />

<TextView

android:id="@+id/LOG3"

android:layout\_below="@id/Back3"

style="@style/EditTextTitle" />

<TextView

android:id="@+id/readLOG3"

android:layout\_below="@id/LOG3"

style="@style/EditTextLOG" />

</RelativeLayout>

</RelativeLayout>

1. Протестируем работу приложения.

При загрузке приложения на первоначальной странице расположено три кнопки (см. рисунок 1), при нажатии на которые открывается интерфейс для работы с конкретной таблицей. На рисунках 2-4 представлена работа приложения с таблицей Производитель. При нажатии на кнопку «Back» можно вернуться на начальную страницу, чтобы выбрать другую таблицу, не выходя из приложения.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1 – Скриншот начальной рабочей области

Изображение выглядит как снимок экрана

Автоматически созданное описание Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 2 – Добавление строки в таблицу «Производитель» и чтение из неё введённых данных

Также можно обновить или удалить строку таблицы по введённому айди (см. рисунок 3).

Изображение выглядит как снимок экрана

Автоматически созданное описание Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3 – Обновление и удаление строки таблицы «Производитель»

Изображение выглядит как снимок экрана

Автоматически созданное описание Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 4 – Чтение данных после обновления и удаления строк соответственно

Аналогично все эти операции производятся с другими таблицами «Вид продукта» и «Продукт». На рисунке 5 изображено считывание заполненных ранее данных из указанных таблиц соответственно.

Изображение выглядит как снимок экрана

Автоматически созданное описание Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 5 – Чтение таблиц «Вид товара» и «Товар»

Также можно удалить все строки таблицы. Удаление содержимого также произойдёт в случае изменения версии БД.

ВЫВОДЫ

В ходе данной лабораторной работы были освоены взаимодействие Android-приложения с SQLite. Созданы таблицы и реалиована стандартная работа с базой данных на основе простейших запросов, таких, как удаление, добавление, обновление данных.