|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_***ИУК «Информатика и управление»*\_\_**\_\_\_\_\_\_\_\_\_\_\_\_

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №4**

**«Использование БД в Android приложениях»**

**ДИСЦИПЛИНА: «Разработка мобильного программного обеспечения»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-62Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Калашников А.Н.)  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Шаматрин А.Г.)  (Подпись) (Ф.И.О.) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |
|  | | |

Калуга, 2023

**Цель:** формирование практических навыков разработки приложений с использованием СУБД SQLite, списков и файлов при разработке Android-приложений с несколькими Activity.

**Задачи:**

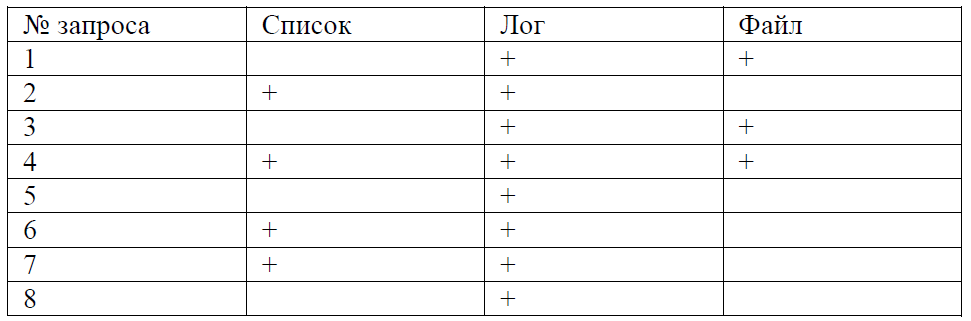
1. Научиться работать с СУБД SQLite.
2. Научиться сохранять результаты выполнения запросов к базе данных в списки, файлы и LogCat.
3. Понять особенности реализации Android-приложений с использованием списков и СУБД SQLite.

**Задание:**

Для всех вариантов создать базу данных согласно варианту задания. К таблице добавить поле id. Сформировать следующие запросы: 1) произвести сортировку всех полей таблицы по одному из числовых полей; 2) сгруппировать данные по нескольким одинаковым полям; 3) вычислить сумму значений одного из числовых полей; 4) вычислить средние значения по сгруппированным полям; 5) отобразить поле с максимальным значением числовой величины; 6) отобразить поля таблицы, в которых числовые величины больше заданной; 7) отобразить поля таблицы, в которых числовые величины меньше средней; 8) отобразить значения только одного из полей, для которых числовое значение больше заданного.

В таблице должно быть не менее 15 записей.

Результаты запроса представить в следующем виде согласно таблице:



При выполнении лабораторной работы использовать intent-объекты. Реализацию всех запросов выполнить в рамках одного приложения.

**Вариант 17. Задание варианта:**

Менеджер: пол, возраст, образование, должность, заработная плата, стаж

**Листинг реализованной программы:**

**MainActivity.kt**

package com.example.myapplication  
  
import android.content.Intent  
import android.os.Bundle  
import android.view.View  
import android.widget.Button  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
  
  
class MainActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_main)  
  
 DatabaseHelper(applicationContext)  
 }  
 fun ex1(view: View) {  
 val intent = Intent(this, ListActivity::class.java).apply {  
 putExtra("query\_name", "query1")  
 putExtra("query", "SELECT \* FROM " + DatabaseHelper.TABLE\_NAME +  
 " ORDER BY " + DatabaseHelper.COL\_WAGES)  
 }  
 startActivity(intent)  
 }  
  
 // Сгруппировать данные по нескольким одинаковым полям (MAKER и COST)  
 fun ex2(view: View) {  
 val intent = Intent(this, ListActivity::class.java).apply {  
 putExtra("query\_name", "query2")  
 putExtra("query", "SELECT \* FROM " + DatabaseHelper.TABLE\_NAME +  
 " GROUP BY " + DatabaseHelper.COL\_GENDER + ", " + DatabaseHelper.COL\_WAGES)  
 }  
 startActivity(intent)  
 }  
  
  
 // Вычислить сумму значений одного из числовых полей (RAM)  
 fun ex3(view: View) {  
 val intent = Intent(this, ListActivity::class.java).apply {  
 putExtra("query\_name", "query3")  
 putExtra("query", "SELECT SUM(" + DatabaseHelper.COL\_WAGES  
 + ") FROM " + DatabaseHelper.TABLE\_NAME)  
 }  
 startActivity(intent)  
 }  
  
  
 // Вычислить средние значения по сгруппированным полям (AVG(COST) планшетов Huawei)  
 fun ex4(view: View) {  
 val intent = Intent(this, ListActivity::class.java).apply {  
 putExtra("query\_name", "query4")  
 putExtra("query", "SELECT " + DatabaseHelper.COL\_GENDER + ", AVG(" + DatabaseHelper.COL\_AGE + ") FROM "  
 + DatabaseHelper.TABLE\_NAME + " GROUP BY " + DatabaseHelper.COL\_GENDER)  
 }  
 startActivity(intent)  
 }  
  
  
 // Отобразить поле с максимальным значением числовой величины  
 fun ex5(view: View) {  
 val intent = Intent(this, ListActivity::class.java).apply {  
 putExtra("query\_name", "query5")  
 putExtra("query", "SELECT \* FROM " + DatabaseHelper.TABLE\_NAME + " WHERE " +  
 DatabaseHelper.COL\_WAGES + " = (SELECT MAX(" + DatabaseHelper.COL\_WAGES  
 + ") FROM " + DatabaseHelper.TABLE\_NAME + ")")  
 }  
 startActivity(intent)  
 }  
  
  
 // Отобразить поля таблицы, в которых числовые величины больше заданной (COST > 21000)  
 fun ex6(view: View) {  
 val intent = Intent(this, ListActivity::class.java).apply {  
 putExtra("query\_name", "query6")  
 putExtra("query", "SELECT \* FROM " + DatabaseHelper.TABLE\_NAME + " WHERE " +  
 DatabaseHelper.COL\_WAGES + " > 21000")  
 }  
 startActivity(intent)  
 }  
  
  
 // Отобразить поля таблицы, в которых числовые величины меньше средней (COST < AVG(COST))  
 fun ex7(view: View) {  
 val intent = Intent(this, ListActivity::class.java).apply {  
 putExtra("query\_name", "query7")  
 putExtra("query", "SELECT \* FROM " + DatabaseHelper.TABLE\_NAME + " WHERE " +  
 DatabaseHelper.COL\_AGE + " < (SELECT AVG(" + DatabaseHelper.COL\_AGE  
 + ") FROM " + DatabaseHelper.TABLE\_NAME + ")")  
 }  
 startActivity(intent)  
 }  
  
  
 // Отобразить значения только одного из полей, для которых числовое значение больше заданного (RAM > 4096)  
 fun ex8(view: View) {  
 val intent = Intent(this, ListActivity::class.java).apply {  
 putExtra("query\_name", "query8")  
 putExtra("query", "SELECT \* FROM " + DatabaseHelper.TABLE\_NAME + " WHERE " +  
 DatabaseHelper.COL\_EXPERIENCE + " > 10 LIMIT 1")  
 }  
 startActivity(intent)  
 }  
}

**ListActivity.kt**

package com.example.md\_labwork4

import android.content.Context

import android.database.Cursor

import android.database.DatabaseUtils

import android.database.sqlite.SQLiteDatabase

import android.os.Bundle

import android.util.Log

import android.view.View

import android.widget.ListView

import android.widget.SimpleCursorAdapter

import android.widget.TextView

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

import java.io.IOException

import java.io.OutputStreamWriter

class ListViewActivity : AppCompatActivity() {

lateinit var header1: TextView

lateinit var header2: TextView

lateinit var list: ListView

lateinit var databaseHelper: DatabaseHelper

lateinit var db: SQLiteDatabase

lateinit var cursor: Cursor

lateinit var cursorAdapter: SimpleCursorAdapter

private lateinit var query\_name: String

private lateinit var query: String

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.list\_view\_activity)

header1 = findViewById<View>(R.id.header1) as TextView

header2 = findViewById<View>(R.id.header2) as TextView

list = findViewById<View>(R.id.list) as ListView

databaseHelper = DatabaseHelper(applicationContext)

query\_name = intent.extras!!.getString("query\_name").toString()

query = intent.extras!!.getString("query").toString()

header1.text = query

}

override fun onResume() {

super.onResume()

db = databaseHelper.readableDatabase

cursor = db.rawQuery(query, null)

writeToFile(query\_name, DatabaseUtils.dumpCursorToString(cursor))

writeToLogcat(query\_name, DatabaseUtils.dumpCursorToString(cursor))

when (query\_name) {

"query3" -> {

cursor.moveToFirst()

header2.text = "SUM(ram): " + cursor.getString(0)

}

"query4" -> {

var data: String = ""

cursor.moveToFirst()

while (cursor.moveToNext()) {

data += "Maker: " + cursor.getString(cursor.getColumnIndex(DatabaseHelper.COLUMN\_MAKER)) + "\n"

data += "AVG(cost): " + cursor.getString(cursor.getColumnIndex("AVG(cost)")) + "\n\n"

}

header2.text = data

}

else -> {

val headers = arrayOf<String>(

DatabaseHelper.COLUMN\_MAKER,

DatabaseHelper.COLUMN\_MEMCARD,

DatabaseHelper.COLUMN\_RAM,

DatabaseHelper.COLUMN\_COST)

cursorAdapter = SimpleCursorAdapter(

this,

R.layout.multi\_line\_list\_item,

cursor,

headers,

intArrayOf(R.id.line1, R.id.line2, R.id.line3, R.id.line4),

1)

header2.text = "Найдено элементов: " + cursor.count

list.adapter = cursorAdapter

}

}

}

override fun onDestroy() {

super.onDestroy()

db.close()

cursor.close()

}

private fun writeToFile(fileName: String, data: String) {

try {

val outputStreamWriter = OutputStreamWriter(this.openFileOutput("$fileName.txt", Context.MODE\_PRIVATE))

outputStreamWriter.write(data)

outputStreamWriter.close()

Toast.makeText(this, "Данные были успешно сохранены в файл", Toast.LENGTH\_LONG).show()

} catch (e: IOException) {

Log.e("Exception", "File write failed: $e")

}

}

private fun writeToLogcat(tag: String, data: String) {

Log.i(tag, data)

Toast.makeText(this, "Данные были успешно сохранены в LogCat", Toast.LENGTH\_LONG).show()

}

}

**DatabaseHelper.kt**

package com.example.myapplication  
  
import android.content.Context  
import android.database.sqlite.SQLiteDatabase  
import android.database.sqlite.SQLiteOpenHelper  
  
class DatabaseHelper(context: Context?) : SQLiteOpenHelper(context, DATABASE\_NAME, null, SCHEMA) {  
  
 companion object {  
  
 private const val DATABASE\_NAME = "mangers.db"  
 private const val SCHEMA = 1  
 const val TABLE\_NAME = "manager"  
  
 const val COL\_ID = "\_id"  
 const val COL\_GENDER = "gender"  
 const val COL\_AGE = "age"  
 const val COL\_EDUCATION = "education"  
 const val COL\_POST = "post"  
 const val COL\_WAGES = "wages"  
 const val COL\_EXPERIENCE = "experience"  
 }  
  
  
 override fun onCreate(db: SQLiteDatabase) {  
  
 db.execSQL("CREATE TABLE $TABLE\_NAME (" +  
 "$COL\_ID INTEGER PRIMARY KEY AUTOINCREMENT,"+  
 "$COL\_GENDER VARCHAR(256),"+  
 "$COL\_AGE INTEGER,"+  
 "$COL\_EDUCATION VARCHAR(256),"+  
 "$COL\_POST VARCHAR(256),"+  
 "$COL\_WAGES INTEGER,"+  
 "$COL\_EXPERIENCE INTEGER);")  
  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',26,'higher','director',30000,5);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',22,'higher','director',45000,4);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',27,'higher','director',60000,3);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',28,'higher','director',20000,2);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',19,'higher','director',15000,1);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',45,'higher','director',18000,9);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',34,'higher','director',19000,14);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',26,'higher','director',21000,20);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',42,'higher','director',45000,14);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('woman',27,'higher','director',22000,1);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',25,'higher','director',34000,2);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('woman',26,'higher','director',30000,6);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',29,'higher','director',19000,8);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('man',28,'higher','director',20000,7);")  
 db.execSQL("INSERT INTO $TABLE\_NAME ($COL\_GENDER, $COL\_AGE, $COL\_EDUCATION, $COL\_POST,$COL\_WAGES,$COL\_EXPERIENCE) " +  
 "VALUES ('woman',24,'higher','director',120000,4);")  
  
 }  
  
 override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {  
 db.execSQL("DROP TABLE IF EXISTS $TABLE\_NAME")  
 onCreate(db)  
 }  
}

**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">

<LinearLayout

android:id="@+id/buttons\_layout"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="vertical"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<Button

android:id="@+id/button\_query1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:width="200dp"

android:text="@string/button\_query1"

android:onClick="query1"/>

<Button

android:id="@+id/button\_query2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:width="200dp"

android:text="@string/button\_query2"

android:onClick="query2"/>

<Button

android:id="@+id/button\_query3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:width="200dp"

android:text="@string/button\_query3"

android:onClick="query3"/>

<Button

android:id="@+id/button\_query4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:width="200dp"

android:text="@string/button\_query4"

android:onClick="query4"/>

<Button

android:id="@+id/button\_query5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:width="200dp"

android:text="@string/button\_query5"

android:onClick="query5"/>

<Button

android:id="@+id/button\_query6"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:width="200dp"

android:text="@string/button\_query6"

android:onClick="query6"/>

<Button

android:id="@+id/button\_query7"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:width="200dp"

android:text="@string/button\_query7"

android:onClick="query7"/>

<Button

android:id="@+id/button\_query8"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

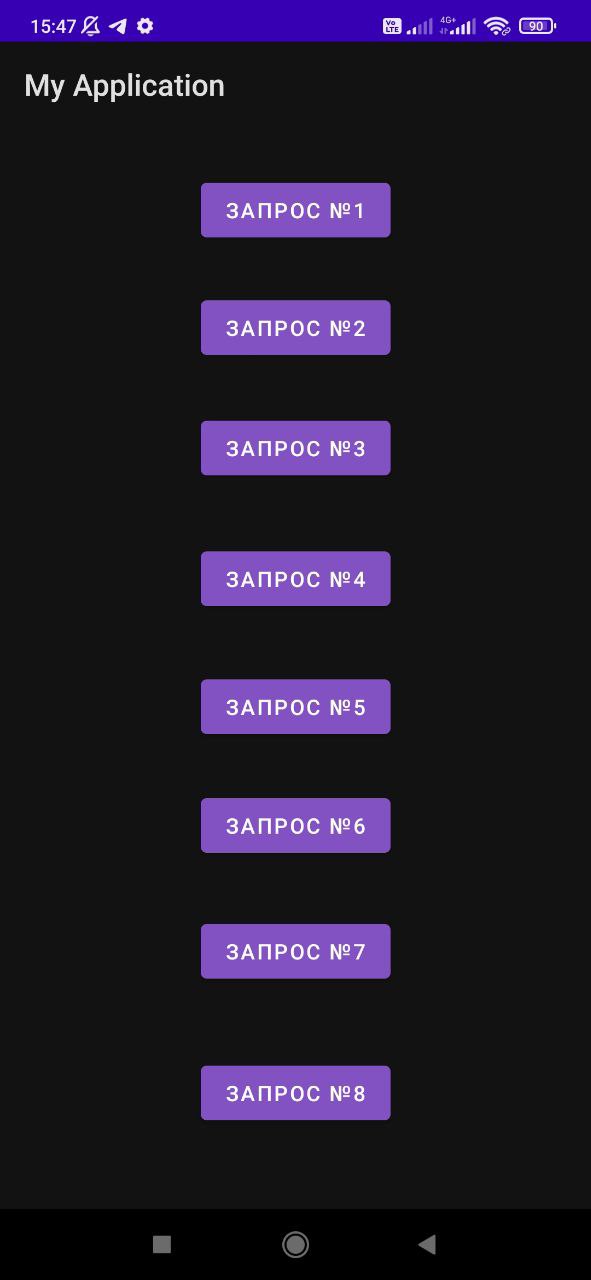
android:width="200dp"

android:text="@string/button\_query8"

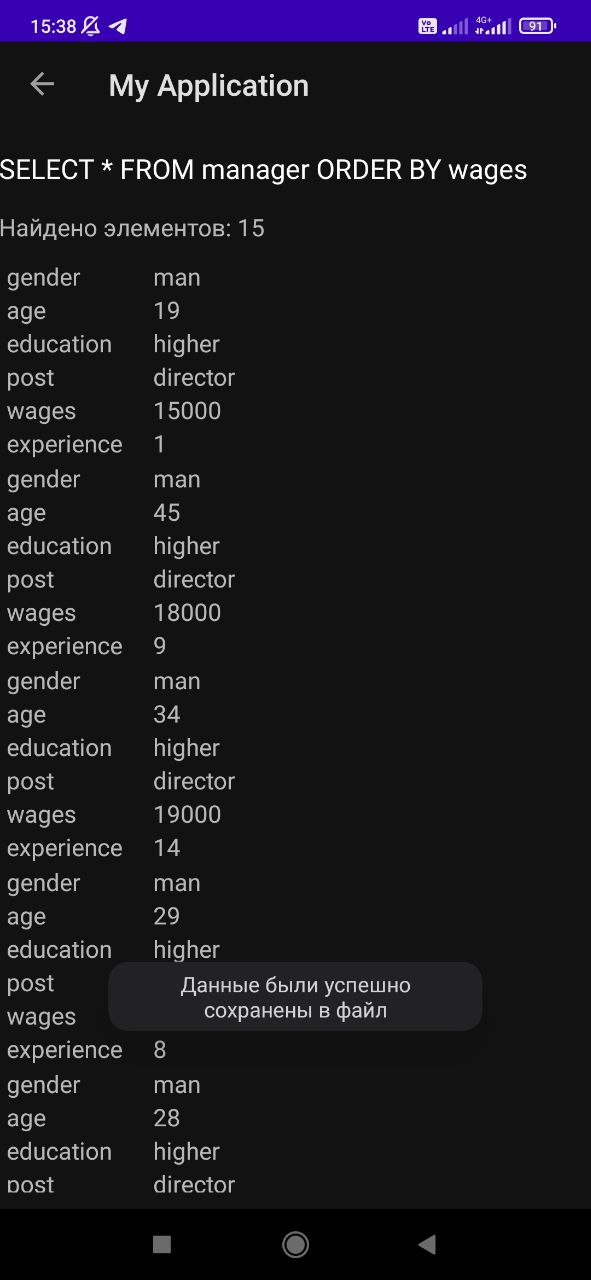
android:onClick="query8"/>

</LinearLayout>

**Результат выполнения работы:**

****

**Рис. 1.** Основная активность приложения

**** ****

**Рис. 2.** Запросы выводом данных в формате списка

D:\3-kurs-2-semestr\Mobile App\LR4_new\Картинки\4.png

**Рис. 3.** Файлы после выполнения

**Вывод:** в ходе выполнения данной лабораторной работы были приобретены практические навыки по разработке приложений с использованием СУБД SQLite, списков и файлов при разработке Android-приложений с несколькими Activity.