



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 \_\_\_\_\_ (Подпись) (Калашников А.Н.)  
(Ф.И.О.)

Проверил: \_\_\_\_\_ (Подпись) (Шаматрин А.Г.)  
(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:

№ запроса	Список	Лог	Файл
1		+	+
2	+	+	
3		+	+
4	+	+	+
5		+	
6	+	+	
7	+	+	
8		+	

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"/>

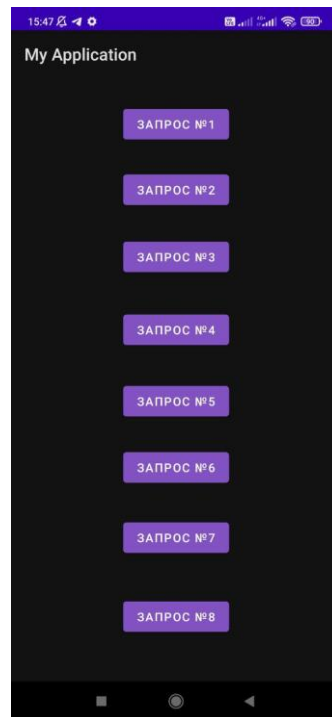
<Button

```

```
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

