



Министерство образования и науки 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 "Фундаментальные науки"

КАФЕДРА "Программное обеспечение ЭВМ, информационные
технологии и прикладная математика"

О Т Ч Е Т

Лабораторная работа №4

ДИСЦИПЛИНА: " Мобильные операционные системы "

ТЕМА: " ИСПОЛЬЗОВАНИЕ БД В ANDROID ПРИЛОЖЕНИЯХ "

Выполнил: студент гр. ИТД-81 Турченко С.А. _____

Проверил: Гришунов С.С. _____

Дата сдачи (защиты) лабораторной работы:

Результаты сдачи (защиты):

Количество рейтинговых баллов

Оценка

Калуга, 2018 г.

Цель:

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

Задача:

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 2

Автомобиль2: тип (седан, кабриолет и пр.), производитель, марка, вместимость багажника, наличие ABS, количество подушек безопасности, средний расход топлива.

Код программы

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    SQLiteDatabase database;
    int count = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        this.initializeDatabase();

        String query = "SELECT * FROM cars";
        Cursor elements = this.database.rawQuery(query, null);
        if (elements.getCount() == 0)
            this.onAddExampleDatabase();
    }

    private void initializeDatabase() {
        this.database = getBaseContext().openOrCreateDatabase
            ("car.db", MODE_PRIVATE, null);
        this.database.execSQL("CREATE TABLE IF NOT EXISTS cars (" +
            "id INTEGER, " +
            "type TEXT, " +
            "manufacture TEXT, " +
            "model TEXT, " +
            "baggage INTEGER, " +
            "abs BOOLEAN, " +
            "safety INTEGER, " +
            "consumption REAL)");
        Log.i("database", "created or opened");
    }

    private boolean addExampleRecord(
        String type,
        String manufacture,
        String model,
        int baggage,
        boolean abs,
        int safety,
        float consumption) {
        ContentValues value = new ContentValues();
        value.put("type", type);
        value.put("manufacture", manufacture);
        value.put("model", model);
        value.put("baggage", baggage);
        value.put("abs", abs);
        value.put("safety", safety);
        value.put("consumption", consumption);
        this.count++;
        return this.database.insert("cars", null, value) >= 0;
    }

    private boolean addExampleData() {
        boolean success = true;

        success &= this.addExampleRecord("легковая", "ваз", "лада", 5, false, 2,
3);
        success &= this.addExampleRecord("легковая", "toyota", "камри", 7, true,
10, 2);
    }
}
```

```

        success &= this.addExampleRecord("легковая", "chevrolet", "нисан", 3,
false, 4, 1);
        success &= this.addExampleRecord("легковая", "ford", "генри", 4, false, 7,
4);
        success &= this.addExampleRecord("легковая", "kia", "корея", 9, true, 6,
3);

        success &= this.addExampleRecord("фургон", "kia", "форгон", 40, true, 4,
9);
        success &= this.addExampleRecord("фургон", "toyota", "камри", 30, false, 5,
8);
        success &= this.addExampleRecord("фургон", "kia", "форгон", 30, true, 3,
7);
        success &= this.addExampleRecord("фургон", "ваз", "газель", 40, true, 1,
10);
        success &= this.addExampleRecord("фургон", "ford", "мандео", 35, false, 4,
8);

        success &= this.addExampleRecord("седан", "ford", "генриХ", 6, true, 8, 6);
        success &= this.addExampleRecord("седан", "kia", "лада", 7, true, 7, 7);
        success &= this.addExampleRecord("седан", "chevrolet", "нисани", 4, false,
6, 5);
        success &= this.addExampleRecord("седан", "ваз", "седан", 6, false, 3, 6);
        success &= this.addExampleRecord("седан", "toyota", "камриХ", 10, true, 15,
7);

        return success;
    }

    public void onAddExampleDatabase() {
        boolean success = this.addExampleData();
        if (success)
            Toast.makeText(this, R.string.database_message_onCreateExample,
Toast.LENGTH_SHORT).show();
        else
            Toast.makeText(this, R.string.database_message_onCreateExample_fail,
Toast.LENGTH_SHORT).show();
    }

    public void onQuery(View view) {
        Intent intent = new Intent(MainActivity.this, QueryViewActivity.class);

        int id = 0;
        boolean isLog = false;
        boolean isFile = false;
        boolean isView = false;
        String query = "* FROM cars";
        eOutputMode mode = eOutputMode.ALL;

        switch (view.getId()) {
            case R.id.activity_query_button_1:
                id = 1;
                isView = false;
                isLog = true;
                isFile = true;
                query = "* FROM cars ORDER BY safety";
                break;
            case R.id.activity_query_button_2:
                id = 2;
                isView = true;
                isLog = true;
                isFile = false;
                query = "* FROM cars GROUP BY safety, abs";
                break;
        }
    }

```

```

        case R.id.activity_query_button_3:
            id = 3;
            isView = false;
            isLog = true;
            isFile = true;
            query = "SUM(baggage) FROM cars";
            mode = eOutputMode.NUMBER;
            break;
        case R.id.activity_query_button_4:
            id = 4;
            isView = true;
            isLog = true;
            isFile = true;
            query = "type, manufacture, model, AVG(baggage) AS baggage, abs,
AVG(safety) AS safety, AVG(consumption) AS consumption " +
                "FROM cars GROUP BY manufacture";
            break;
        case R.id.activity_query_button_5:
            id = 5;
            isView = false;
            isLog = true;
            isFile = false;
            query = "* FROM cars WHERE safety = (SELECT MAX(safety) FROM
cars)";
            break;
        case R.id.activity_query_button_6:
            id = 6;
            isView = true;
            isLog = true;
            isFile = false;
            String value =
((EditText)findViewById(R.id.activity_main_input)).getText().toString();
            query = "* FROM cars WHERE " +
                "consumption > " + value +
                " AND baggage > " +value +
                " AND safety > " +value;
            break;
        case R.id.activity_query_button_7:
            id = 7;
            isView = true;
            isLog = true;
            isFile = false;
            query = "* FROM cars WHERE " +
                "consumption < (SELECT AVG(consumption) FROM cars)" +
                "AND baggage < (SELECT AVG(baggage) FROM cars)" +
                "AND safety < (SELECT AVG(safety) FROM cars)";
            break;
        case R.id.activity_query_button_8:
            id = 8;
            isView = false;
            isLog = true;
            isFile = false;
            query = "* FROM cars WHERE consumption > 5";
            mode = eOutputMode.FIRST;
            break;
    }

    intent.putExtra("query", "SELECT rowid _id, " + query);
    intent.putExtra("log", isLog);
    intent.putExtra("file", isFile);
    intent.putExtra("view", isView);
    intent.putExtra("id", id);
    intent.putExtra("mode", mode);

    String logRecord = "query" +

```

```

        "request query #" + Integer.toString(id) +
        "\nlog output: " + Boolean.toString(isLog) +
        "\nfile output: " + Boolean.toString(isFile) +
        "\nview output: " + Boolean.toString(isView) +
        "\nquery: " + query;

        Log.i("query", logRecord);

        startActivity(intent);

    }

}

```

QueryViewActivity.java

```

public class QueryViewActivity extends Activity {

    SQLiteDatabase database;
    String query;
    boolean logOutput;
    boolean fileOutput;
    boolean viewOutput;
    eOutputMode mode;
    int id;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_query_view);

        Bundle extras = getIntent().getExtras();

        this.query = extras.getString("query");
        this.logOutput= extras.getBoolean("log");
        this.fileOutput= extras.getBoolean("file");
        this.viewOutput= extras.getBoolean("view");
        this.mode = (eOutputMode) extras.getSerializable("mode");
        this.id = extras.getInt("id");

        this.openDatabase();

        this.requestStoragePermission();

        TextView label = findViewById(R.id.activity_query_view_label);
        String newLabel = label.getText().toString().replace("%id%",
Integer.toString(this.id));
        label.setText(newLabel);

        this.startQuery();
    }

    private void openDatabase() {
        this.database = getBaseContext().openOrCreateDatabase
            ("car.db", MODE_PRIVATE, null);
        Log.i("database", "opened");
    }

    public void requestStoragePermission() {
        ActivityCompat.requestPermissions(this,
            new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
            0);
    }

    private String recordToString(Cursor cursor) {

```

```

        StringBuilder result = new StringBuilder();

        result.append("\n---");
        result.append("\nrecord #" +
Integer.toString(cursor.getInt(cursor.getColumnIndex("_id"))));
        result.append("\ntype: " +
cursor.getString(cursor.getColumnIndex("type")));
        result.append("\nmanufacture: " +
cursor.getString(cursor.getColumnIndex("manufacture")));
        result.append("\nmodel: " +
cursor.getString(cursor.getColumnIndex("model")));
        result.append("\nbaggage: " +
Integer.toString(cursor.getInt(cursor.getColumnIndex("baggage"))));
        //result.append("abs" +
Integer.toString(cursor.getInt(cursor.getColumnIndex("abs"))));
        result.append("\nsafety: " +
Integer.toString(cursor.getInt(cursor.getColumnIndex("safety"))));
        result.append("\nconsumption: " +
Float.toString(cursor.getFloat(cursor.getColumnIndex("consumption"))));
        result.append("\n---");

        return result.toString();
    }

    private void logCursor(Cursor cursor) {
        do {
            Log.i("database/cursor", this.recordToString(cursor));
        } while (cursor.moveToNext());
        Toast.makeText(this, R.string.database_message_output_log_success,
Toast.LENGTH_SHORT).show();
    }

    private void logCursorFirst (Cursor cursor) {
        Log.i("database/cursor", this.recordToString(cursor));
        Toast.makeText(this, R.string.database_message_output_log_success,
Toast.LENGTH_SHORT).show();
    }

    private void writeToFile(String content) {
        File cursorFile = new File(Environment.getExternalStoragePublicDirectory(
            Environment.DIRECTORY_DOWNLOADS), "query" + this.id + ".txt");
        Log.i("file", "write in `" + cursorFile.getAbsolutePath() + "` folder");
        try {

            cursorFile.createNewFile();
            FileOutputStream outputStream = new FileOutputStream(cursorFile,
false);
            outputStream.write(content.getBytes(), 0, content.getBytes().length);
            outputStream.close();
        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(this, R.string.database_message_output_file_fail,
Toast.LENGTH_SHORT).show();
            return;
        }
        Toast.makeText(this, R.string.database_message_output_file_success,
Toast.LENGTH_SHORT).show();
    }

    public void fileCursor(Cursor cursor) {
        StringBuilder result = new StringBuilder();

        do {
            result.append(this.recordToString(cursor));
        } while (cursor.moveToNext());
    }

```

```

        this.writeToFile(result.toString());
    }

    public void fileCursorFirst(Cursor cursor) {
        this.writeToFile(this.recordToString(cursor));
    }

    public void viewCursor(Cursor cursor) {
        Log.i("view", "binding adapter");
        ListView lvItems = (ListView) findViewById(R.id.activity_query_view_items);
        QueryAdapter adapter= new QueryAdapter(this, cursor);
        lvItems.setAdapter(adapter);
        Toast.makeText(this, R.string.database_message_output_view_success,
Toast.LENGTH_SHORT).show();
    }

    public String numbersToString(float[] numbers) {
        if (numbers.length == 1)
            return Float.toString(numbers[0]);
        return Arrays.toString(numbers);
    }

    public void logNumber(float[] numbers) {
        Log.i("database/cursor", this.numbersToString(numbers));
    }

    public void fileNumber(float[] numbers) {
        this.writeToFile(this.numbersToString(numbers));
    }

    public void viewNumber(float[] numbers) {
        TextView label = findViewById(R.id.activity_query_view_label);
        String newLabel = label.getText().toString() + "\nresult " +
this.numbersToString(numbers);
        label.setText(newLabel);
    }

    public void startQuery() {
        Cursor result = this.database.rawQuery(this.query, null);

        if (result.getCount() == 0)
            return;

        if (this.logOutput) {
            result.moveToFirst();
            switch (this.mode) {
                case ALL:
                    this.logCursor(result);
                    break;
                case FIRST:
                    this.logCursorFirst(result);
                    break;
                case NUMBER:
                    float r = result.getFloat(0);
                    this.logNumber(new float[] { r });
                    break;
            }
        }

        if (this.fileOutput) {
            result.moveToFirst();
            switch (this.mode) {
                case ALL:
                    this.fileCursor(result);

```



```

        break;
    case FIRST:
        this.fileCursorFirst(result);
        break;
    case NUMBER:
        float r = result.getFloat(0);
        this.fileNumber(new float[] { r });
        break;
    }
}

if (this.viewOutput) {
    result.moveToFirst();
    switch (this.mode) {
        case ALL:
            this.viewCursor(result);
            break;
        case FIRST:
            result.moveToLast();
            this.viewCursor(result);
            break;
        case NUMBER:
            float r = result.getFloat(0);
            this.viewNumber(new float[] { r });
            break;
    }
}
}
}
}

```

QueryAdapter.java

```

public class QueryAdapter extends CursorAdapter {
    public QueryAdapter(Context context, Cursor cursor) {
        super(context, cursor, 0);
    }

    @Override
    public View newView(Context context, Cursor cursor, ViewGroup parent) {
        return LayoutInflater.from(context).inflate(R.layout.quey_output, parent,
false);
    }

    @Override
    public void bindView(View view, Context context, Cursor cursor) {
        TextView dataId = view.findViewById(R.id.query_id_value);
        int idValue = cursor.getInt(cursor.getColumnIndexOrThrow("_id"));
        dataId.setText(Integer.toString(idValue));

        TextView dataType = view.findViewById(R.id.query_type_value);
        String typeValue = cursor.getString(cursor.getColumnIndexOrThrow("type"));
        dataType.setText(typeValue);

        TextView dataManufacture = view.findViewById(R.id.query_manufacture_value);
        String manufactureValue =
cursor.getString(cursor.getColumnIndexOrThrow("manufacture"));
        dataManufacture.setText(manufactureValue);

        TextView dataModel = view.findViewById(R.id.query_model_value);
        String modelValue =
cursor.getString(cursor.getColumnIndexOrThrow("model"));
        dataModel.setText(modelValue);
    }
}

```

```

        TextView dataBaggage = view.findViewById(R.id.query_baggage_value);
        int baggageValue = cursor.getInt(cursor.getColumnIndexOrThrow("baggage"));
        dataBaggage.setText(Integer.toString(baggageValue));

        TextView dataAbs = view.findViewById(R.id.query_abs_value);
        int absValue = cursor.getInt(cursor.getColumnIndexOrThrow("abs"));
        dataAbs.setText(Integer.toString(absValue));

        TextView dataSafety = view.findViewById(R.id.query_safety_value);
        int safetyValue = cursor.getInt(cursor.getColumnIndexOrThrow("safety"));
        dataSafety.setText(Integer.toString(safetyValue));

        TextView dataConsumption = view.findViewById(R.id.query_consumption_value);
        float consumptionValue =
        cursor.getFloat(cursor.getColumnIndexOrThrow("consumption"));
        dataConsumption.setText(Float.toString(consumptionValue));

    }
}

```

eOutputMode.java

```

public enum eOutputMode {

    ALL,
    FIRST,
    NUMBER

}

```

Результат

lab4	view output of query #2 --- начало записи ---		view output of query #7 --- начало записи ---	
	id	9	id	1
	тип	фургон	тип	легковая
	производитель	ваз	производитель	ваз
ЗАПРОС 4	модель	газель	модель	лада
ЗАПРОС 5	багажник	40	багажник	5
	есть abs	1	есть abs	0
	подушки безопасности	1	подушки безопасности	2
	потребление топлива	10.0	потребление топлива	3.0
ЗАПРОС 6	--- конец записи ---		--- конец записи ---	
	--- начало записи ---		--- начало записи ---	
	id	1	id	3
	тип	легковая	тип	легковая
ЗАПРОС 7	производитель	ваз	производитель	chevrolet
	модель	лада	модель	нисан
	багажник	5	багажник	3
	есть abs	0	есть abs	0
ЗАПРОС 8	подушки безопасности	2	подушки безопасности	4
	потребление топлива	3.0	потребление топлива	1.0
	--- конец записи ---		--- конец записи ---	
	--- начало записи ---			
5	id	14		
	тип	седан		
	производитель	ваз		
	модель	седан		
	багажник	6		
	есть abs	0		
	подушки безопасности	3		
	потребление топлива	6.0		

Вывод

Получены практические навыки разработки приложений с использованием СУБД SQLite, списков, файлов и несколькими Activity.