

**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



**Звіт**

**до лабораторної роботи №5**

**з дисципліни: «Програмні технології мобільних обчислень»**

**на тему: «Управління даними в ОС Android за допомогою СУБД SQLite»**

**Варіант №8**

**Виконав студент групи КІ-44:**

**Козлюк Д.С.**

**Перевірив:**

**Бачинський Р.В.**

**Львів 2023**

**Мета:** Оволодіти навичками роботи з СУБД SQLite в ОС Android.

**Завдання:**

- Створити базу даних навчальних предметів біжучого семестру (назва предмету, обсяг предмету (навчальні години), викладач, розклад, показники успішності) та тестову програму перегляду і редагування записів бази даних.
- Створити базу даних сучасних мобільних операційних систем (Android, iOS, BlackBerry OS, Symbian OS, Windows Phone, Bada та ін. з інформацією про компанію-власника, біжучу версію, архітектурні особливості (сімейство і т.п.), біжучу долю на ринку мобільних обчислювальних пристроїв та ін.) та тестову програму перегляду і редагування записів бази даних.

### **Хід виконання**

#### **1. Перелік основних методів для роботи з СУБД SQLite в ОС Android.**

В ОС Android для роботи з СУБД SQLite використовується абстрактний клас SQLiteOpenHelper. Цей клас надає основні методи для створення, відкриття, закриття та оновлення баз даних.

#### **Основні методи класу SQLiteOpenHelper:**

- onCreate(): викликається при створенні бази даних. У цьому методі можна створити необхідні таблиці та індекси.
- onUpgrade(): викликається при оновленні бази даних. У цьому методі можна модифікувати або видалити таблиці та індекси.
- getWritableDatabase(): повертає об'єкт SQLiteDatabase, який можна використовувати для запису в базу даних.
- getReadableDatabase(): повертає об'єкт SQLiteDatabase, який можна використовувати для читання з бази даних.
- close(): закриває базу даних.

Додаткові методи класу SQLiteOpenHelper:

- execSQL(): виконує SQL-запит.

- insert(): вставляє запис у таблицю.
- update(): оновлює запис у таблиці.
- delete(): видаляє запис з таблиці.

Приклад використання методів класу SQLiteOpenHelper:

```
public class MySQLiteOpenHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "my_database";
    private static final int DATABASE_VERSION = 1;

    public MySQLiteOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE users (id INTEGER PRIMARY KEY, name TEXT, age INTEGER)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Виконуємо оновлення бази даних
    }
}
```

Цей клас створює базу даних з однією таблицею users. Таблиця має три поля: id (первинний ключ), name (рядок) та age (ціле число).

## 2. Лістинг тестової програми

### 2.1. База даних сучасних мобільних операційних систем

#### DatabaseHelper.java

```
package com.example.lab5_sql;

import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import java.util.ArrayList;
import java.util.List;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "OperatingSystem.db";
    private static final String TABLE_NAME = "System_table";
    private static final String COL_1 = "ID";
    private static final String COL_2 = "SYSTEM";
    private static final String COL_3 = "COMPANY";
```

```

private static final String COL_4 = "VERSION";
private static final String COL_5 = "ARCHITECTURE";
private static final String COL_6 = "DESTINY";

public DatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, 1);
}

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table " + TABLE_NAME + " (ID INTEGER PRIMARY KEY AUTOIN-
CREMENT,SYSTEM TEXT,COMPANY TEXT,VERSION TEXT,ARCHITECTURE TEXT,DESTINY TEXT)");
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(db);
}

public boolean insertData(String subject_name, String scope, String teacher,
String schedule, String success_rate) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COL_2, subject_name);
    contentValues.put(COL_3, scope);
    contentValues.put(COL_4, teacher);
    contentValues.put(COL_5, schedule);
    contentValues.put(COL_6, success_rate);
    long result = db.insert(TABLE_NAME, null, contentValues);
    return result != -1;
}

public Cursor getAllData() {
    SQLiteDatabase db = this.getWritableDatabase();
    return db.rawQuery("select * from " + TABLE_NAME, null);
}

@SuppressLint("Range")
public List<OperatingSystem> fetchDataFromDatabase() {
    List<OperatingSystem> data = new ArrayList<>();
    Cursor cursor = this.getAllData();

    if (cursor != null) {
        while (cursor.moveToNext()) {
            @SuppressLint("Range") int id = cursor.getInt(cursor.getColumnIndex(COL_1));
            @SuppressLint("Range") String subjectName = cursor.getString(cur-
sor.getColumnIndex(COL_2));
            @SuppressLint("Range") String scope = cursor.getString(cur-
sor.getColumnIndex(COL_3));
            @SuppressLint("Range") String teacher = cursor.getString(cur-
sor.getColumnIndex(COL_4));
            @SuppressLint("Range") String schedule = cursor.getString(cur-
sor.getColumnIndex(COL_5));
            @SuppressLint("Range") String successRate = cursor.getString(cur-
sor.getColumnIndex(COL_6));

            // Create a Subject object and add it to the list

```

```

        OperatingSystem operatingSystem = new OperatingSystem(id, subject-
Name, scope, teacher, schedule, successRate);
        data.add(operatingSystem);
    }
    cursor.close();
}

return data;
}

public void deleteSubject(int id) {
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_NAME, COL_1 + " = ? ", new String[] { Integer.toString(id)
});
}

public boolean deleteSubjectById(int id) {
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete(TABLE_NAME, COL_1 + "=" + id, null) > 0;
}
}

```

## InsertData.java

```

package com.example.lab5_sql;

import android.content.ContentValues;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class InsertData extends AppCompatActivity {

    private EditText subjectNameEditText;
    private EditText scopeEditText;
    private EditText teacherEditText;
    private EditText scheduleEditText;
    private EditText successRateEditText;
    private Button saveButton;
    private DatabaseHelper dbHelper;

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

        subjectNameEditText = findViewById(R.id.editTextSubjectName);
        scopeEditText = findViewById(R.id.editTextScope);
        teacherEditText = findViewById(R.id.editTextTeacher);
        scheduleEditText = findViewById(R.id.editTextSchedule);
        successRateEditText = findViewById(R.id.editTextSuccessRate);
    }
}

```

```

        saveButton = findViewById(R.id.buttonSave);
        dbHelper = new DatabaseHelper(this);

        saveButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(insertData())
                {
                    Intent intent = new Intent(InsertData.this, Lab5.class);
                    startActivity(intent);
                }
            }
        });
    }

    private boolean insertData() {
        String subjectName = subjectNameEditText.getText().toString();
        String scope = scopeEditText.getText().toString();
        String teacher = teacherEditText.getText().toString();
        String schedule = scheduleEditText.getText().toString();
        String successRate = successRateEditText.getText().toString();

        if (subjectName.isEmpty() || scope.isEmpty() || teacher.isEmpty() || schedule.isEmpty() || successRate.isEmpty()) {
            // Show an error message or toast indicating that all fields must be filled.
            Toast.makeText(this, "Please fill in all fields",
                Toast.LENGTH_SHORT).show();
            return false;
        } else {
            // All fields are filled, so proceed with the insertion.
            return dbHelper.insertData(subjectName, scope, teacher, schedule, successRate);
        }
    }
}

```

## Lab5.java

```

package com.example.lab5_sql;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class Lab5 extends AppCompatActivity
{

```

```

// Define variables for UI elements
private Button insertButton;

private DatabaseHelper dbHelper;
private Button updateButton;
private Button deleteButton;
private List<OperatingSystem> operatingSystemList;
private RecyclerView recyclerView;

private OperatingSystemAdapter adapter;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
    setTitle("Lab5_Kozliuk_KI44");

    // Initialize UI elements

    dbHelper = new DatabaseHelper(this);

    insertButton = findViewById(R.id.buttonInsert);
    updateButton = findViewById(R.id.buttonUpdate);
    deleteButton = findViewById(R.id.buttonDelete);

    recyclerView = findViewById(R.id.subjectRecyclerView);

    recyclerView.setLayoutManager(new LinearLayoutManager(this));

    appendToRecycle();

    // Set click listeners for the buttons
    insertButton.setOnClickListener(view -> {
        // Launch the InsertDataActivity when the Insert button is clicked
        Intent intent = new Intent(Lab5.this, InsertData.class);
        startActivity(intent);
    });

    updateButton.setOnClickListener(view -> {
        // Launch the UpdateDataActivity when the Update button is clicked
        // Intent intent = new Intent(MainActivity.this, UpdateDataActiv-
ity.class);
        //
        startActivity(intent);
    });

    deleteButton.setOnClickListener(view -> {
        // Launch the DeleteDataActivity when the Delete button is clicked
        //
        Intent intent = new Intent(MainActivity.this, DeleteDataActiv-
ity.class);
        //
        startActivity(intent);
        int selectedItemPosition = adapter.getSelectedItemPosition();
        if (selectedItemPosition != -1) {

            // Delete the item from the data source (e.g., subjectList)
            // Refresh the RecyclerView
            // For example, if using a List<Subject>:
            //dbHelper.deleteSubject(selectedItemPosition);

```

```

        boolean isDeleted = dbHelper.deleteSubjectById((int) operatingSystem-
List.get(selectedItemPosition).getId());
        if (isDeleted) {
            Toast.makeText(getApplicationContext(), "Subject deleted success-
fully", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "Failed to delete sub-
ject", Toast.LENGTH_SHORT).show();
        }
        operatingSystemList.remove(selectedItemPosition);
        adapter.clearSelectedItem(); // Clear the selected item
        adapter.notifyItemRemoved(selectedItemPosition);
    }

});
}

void appendToRecycle()
{
    operatingSystemList = dbHelper.fetchDataFromDatabase();
    adapter = new OperatingSystemAdapter(operatingSystemList);
    recyclerView.setAdapter(adapter);
};
}

```

## OperatingSystem.java

```

package com.example.lab5_sql;

public class OperatingSystem {
    private long id; // Unique subject ID
    private String name;

    public OperatingSystem(long id, String name, String company, String version,
String rating, String architecture) {
        this.id = id;
        this.name = name;
        this.company = company;
        this.version = version;
        this.rating = rating;
        this.architecture = architecture;
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }
}

```



```
public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getCompany() {
    return company;
}

public void setCompany(String company) {
    this.company = company;
}

public String getVersion() {
    return version;
}

public void setVersion(String version) {
    this.version = version;
}

public String getRating() {
    return rating;
}

public void setRating(String rating) {
    this.rating = rating;
}

public String getArchitecture() {
    return architecture;
}

public void setArchitecture(String architecture) {
    this.architecture = architecture;
}

private String company;
private String version;
private String rating;
private String architecture;

// Constructors, getters, setters, and other methods go here
}
```

## OperatingSystemAdapter.java

```
package com.example.lab5_sql;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.Color;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class OperatingSystemAdapter extends RecyclerView.Adapter<OperatingSystem-
Adapter.SubjectViewHolder> {
    private List<OperatingSystem> operatingSystemList;
    private int selectedItem = -1;

    private boolean doubleClick = false;

    private long lastClickTime = 0;

    public int getSelectedItemPosition() {
        return selectedItem;
    }

    public void clearSelectedItem() {
        selectedItem = -1;
        notifyDataSetChanged();
    }

    Button deleteButton;
    public OperatingSystemAdapter(List<OperatingSystem> operatingSystemList) {
        this.operatingSystemList = operatingSystemList;
    }
    public void removeItem(int position) {
        operatingSystemList.remove(position);
        notifyItemRemoved(position);
    }
    @NonNull
    @Override
    public SubjectViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View itemView = LayoutInflater.from(parent.getContext()).inflate(R.lay-
out.item, parent, false);

        return new SubjectViewHolder(itemView);
    }

    @SuppressWarnings("SetTextI18n")
    @Override
    public void onBindViewHolder(@NonNull SubjectViewHolder holder, @Sup-
pressLint("RecyclerView") int position) {
```

```

        OperatingSystem operatingSystem = operatingSystemList.get(position);
        holder.subjectNameTextView.setText(operatingSystem.getName());
        //holder.scopeTextView.setText("Обсяг: " + operatingSystem.getCompany());
        //holder.schedule.setText("Розклад: \n" + operatingSystem.getRating());
        //holder.rate.setText("Успішність " + operatingSystem.getArchitecture());
        //holder.teacher.setText("Викладач: " + operatingSystem.getVersion());
        // Populate other fields as needed

        // Highlight the clicked item
        if (position == selectedItem) {
            holder.itemView.setBackgroundColor(ContextCompat.getColor(holder.itemView.getContext(), R.color.highlightColor));
        } else {
            holder.itemView.setBackgroundColor(Color.TRANSPARENT); // Reset background
        }
        // Handle item click
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                selectedItem = position; long clickTime = System.currentTimeMillis();
                notifyDataSetChanged(); // Refresh the RecyclerView to apply highlighting

                // Double-click detected
                doubleClick = true;
                onDoubleClick(view, operatingSystemList.get(selectedItem));
            }
        });
    }

    private void onDoubleClick(View view, OperatingSystem system) {
        // Handle double-click, switch to a new window (activity) here
        // Example: Start a new activity
        Intent intent = new Intent(view.getContext(), OSDetailsActivity.class);
        intent.putExtra("osName", system.getName());
        intent.putExtra("ownerCompany", system.getCompany());
        intent.putExtra("version", system.getVersion());
        intent.putExtra("features", system.getArchitecture());
        intent.putExtra("marketStatus", system.getRating());
        view.getContext().startActivity(intent);
    }

    @Override
    public int getItemCount() {
        return operatingSystemList.size();
    }

    public static class SubjectViewHolder extends RecyclerView.ViewHolder {
        TextView subjectNameTextView;
        TextView scopeTextView;

        TextView teacher;

        TextView rate;
    }

```

```

        TextView schedule;
        // Define other TextViews for teacher, schedule, success rate

        public SubjectViewHolder(@NonNull View itemView) {
            super(itemView);
            subjectNameTextView = itemView.findViewById(R.id.textViewSubjectName);

        }

    }
}

```

## OSDetailActivity.java

```

package com.example.lab5_sql;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class OSDetailsActivity extends AppCompatActivity {

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

        // Get references to the TextViews in the layout
        TextView osNameTextView = findViewById(R.id.textViewOSName);
        TextView ownerCompanyTextView = findViewById(R.id.textViewOwnerCompany);
        TextView versionTextView = findViewById(R.id.textViewVersion);
        TextView featuresTextView = findViewById(R.id.textViewFeatures);
        TextView marketStatusTextView = findViewById(R.id.textViewMarketStatus);

        // Retrieve OS details from the intent
        Intent intent = getIntent();
        if (intent != null) {
            String osName = intent.getStringExtra("osName");
            String ownerCompany = intent.getStringExtra("ownerCompany");
            String version = intent.getStringExtra("version");
            String features = intent.getStringExtra("features");
            String marketStatus = intent.getStringExtra("marketStatus");

            // Set the TextViews with the retrieved information
            osNameTextView.setText(osName);
            ownerCompanyTextView.setText("Owner Company: " + ownerCompany);
            versionTextView.setText("Current Version: " + version);
            featuresTextView.setText("Architectural Features: " + marketStatus);
            marketStatusTextView.setText("Market Status: " + features);

        }
    }
}

```

## 2.2. База даних навчальних предметів біжучого семестру

### DataBaseHelper.java

```
package com.example.lab5_sql;

import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import java.util.ArrayList;
import java.util.List;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "Subjects.db";
    private static final String TABLE_NAME = "Subjects_table";
    private static final String COL_1 = "ID";
    private static final String COL_2 = "SUBJECT_NAME";
    private static final String COL_3 = "SCOPE";
    private static final String COL_4 = "TEACHER";
    private static final String COL_5 = "SCHEDULE";
    private static final String COL_6 = "SUCCESS_RATE";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table " + TABLE_NAME + " (ID INTEGER PRIMARY KEY AUTOIN-  
CREMENT, SUBJECT_NAME TEXT, SCOPE TEXT, TEACHER TEXT, SCHEDULE TEXT, SUCCESS_RATE TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }

    public boolean insertData(String subject_name, String scope, String teacher,
String schedule, String success_rate) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_2, subject_name);
        contentValues.put(COL_3, scope);
        contentValues.put(COL_4, teacher);
        contentValues.put(COL_5, schedule);
        contentValues.put(COL_6, success_rate);
        long result = db.insert(TABLE_NAME, null, contentValues);
        return result != -1;
    }

    public Cursor getAllData() {
        SQLiteDatabase db = this.getWritableDatabase();
        return db.rawQuery("select * from " + TABLE_NAME, null);
    }
}
```

```

    }

    @SuppressWarnings("Range")
    public List<Subject> fetchDataFromDatabase() {
        List<Subject> data = new ArrayList<>();
        Cursor cursor = this.getAllData();

        if (cursor != null) {
            while (cursor.moveToNext()) {
                @SuppressWarnings("Range") int id = cursor.getInt(cursor.getColumnIndex(COL_1));
                @SuppressWarnings("Range") String subjectName = cursor.getString(cursor.getColumnIndex(COL_2));
                @SuppressWarnings("Range") String scope = cursor.getString(cursor.getColumnIndex(COL_3));
                @SuppressWarnings("Range") String teacher = cursor.getString(cursor.getColumnIndex(COL_4));
                @SuppressWarnings("Range") String schedule = cursor.getString(cursor.getColumnIndex(COL_5));
                @SuppressWarnings("Range") String successRate = cursor.getString(cursor.getColumnIndex(COL_6));

                // Create a Subject object and add it to the list
                Subject subject = new Subject(id, subjectName, scope, teacher, schedule, successRate);
                data.add(subject);
            }
            cursor.close();
        }

        return data;
    }

    public void deleteSubject(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_NAME, COL_1 + " = ? ", new String[] { Integer.toString(id)
    });
    }

    public boolean deleteSubjectById(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        return db.delete(TABLE_NAME, COL_1 + "=" + id, null) > 0;
    }
}

```

## InsertData.java

```

package com.example.lab5_sql;

import android.content.ContentValues;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

```

```

import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class InsertData extends AppCompatActivity {

    private EditText subjectNameEditText;
    private EditText scopeEditText;
    private EditText teacherEditText;
    private EditText scheduleEditText;
    private EditText successRateEditText;
    private Button saveButton;
    private DatabaseHelper dbHelper;

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

        subjectNameEditText = findViewById(R.id.editTextSubjectName);
        scopeEditText = findViewById(R.id.editTextScope);
        teacherEditText = findViewById(R.id.editTextTeacher);
        scheduleEditText = findViewById(R.id.editTextSchedule);
        successRateEditText = findViewById(R.id.editTextSuccessRate);
        saveButton = findViewById(R.id.buttonSave);
        dbHelper = new DatabaseHelper(this);

        saveButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(insertData())
                {
                    Intent intent = new Intent(InsertData.this, Lab5.class);
                    startActivity(intent);
                }
            }
        });
    }

    private boolean insertData() {
        String subjectName = subjectNameEditText.getText().toString();
        String scope = scopeEditText.getText().toString();
        String teacher = teacherEditText.getText().toString();
        String schedule = scheduleEditText.getText().toString();
        String successRate = successRateEditText.getText().toString();

        if (subjectName.isEmpty() || scope.isEmpty() || teacher.isEmpty() || schedule.isEmpty() || successRate.isEmpty()) {
            // Show an error message or toast indicating that all fields must be filled.
            Toast.makeText(this, "Please fill in all fields", Toast.LENGTH_SHORT).show();
            return false;
        } else {
            // All fields are filled, so proceed with the insertion.
            return dbHelper.insertData(subjectName, scope, teacher, schedule, successRate);
        }
    }
}

```

## Lab5.java

```
package com.example.lab5_sql;

import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.List;

public class Lab5 extends AppCompatActivity
{
    // Define variables for UI elements
    private Button insertButton;

    private DatabaseHelper dbHelper;
    private Button updateButton;
    private Button deleteButton;
    private List<Subject> subjectList;
    private RecyclerView recyclerView;

    private SubjectAdapter adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main_layout);
        setTitle("Lab5_Kozliuk_KI44");

        // Initialize UI elements

        dbHelper = new DatabaseHelper(this);

        insertButton = findViewById(R.id.buttonInsert);
        updateButton = findViewById(R.id.buttonUpdate);
        deleteButton = findViewById(R.id.buttonDelete);

        recyclerView = findViewById(R.id.subjectRecyclerView);

        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        appendToRecycle();

        // Set click listeners for the buttons
        insertButton.setOnClickListener(view -> {
            // Launch the InsertDataActivity when the Insert button is clicked
```



```

        Intent intent = new Intent(Lab5.this, InsertData.class);
        startActivity(intent);
    });

    updateButton.setOnClickListener(view -> {
        // Launch the UpdateDataActivity when the Update button is clicked
        // Intent intent = new Intent(MainActivity.this, UpdateDataActiv-
ity.class);
        //
        startActivity(intent);
    });

    deleteButton.setOnClickListener(view -> {
        // Launch the DeleteDataActivity when the Delete button is clicked
        // Intent intent = new Intent(MainActivity.this, DeleteDataActiv-
ity.class);
        //
        startActivity(intent);
        int selectedItemPosition = adapter.getSelectedItemPosition();
        if (selectedItemPosition != -1) {

            // Delete the item from the data source (e.g., subjectList)
            // Refresh the RecyclerView
            // For example, if using a List<Subject>:
            //dbHelper.deleteSubject(selectedItemPosition);

            boolean isDeleted = dbHelper.deleteSubjectById((int) subject-
List.get(selectedItemPosition).getId());
            if (isDeleted) {
                Toast.makeText(getApplicationContext(), "Subject deleted success-
fully", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(getApplicationContext(), "Failed to delete sub-
ject", Toast.LENGTH_SHORT).show();
            }
            subjectList.remove(selectedItemPosition);
            adapter.clearSelectedItem(); // Clear the selected item
            adapter.notifyItemRemoved(selectedItemPosition);

        }

    });
}

void appendToRecycle()
{
    subjectList = dbHelper.fetchDataFromDatabase();
    adapter = new SubjectAdapter(subjectList);
    recyclerView.setAdapter(adapter);
};
}

```

## Subject

```
package com.example.lab5_sql;
```

```
public class Subject {
    private long id; // Unique subject ID
    private String name;

    public Subject(long id, String name, String studyHours, String teacher, String
schedule, String successRate) {
        this.id = id;
        this.name = name;
        this.studyHours = studyHours;
        this.teacher = teacher;
        this.schedule = schedule;
        this.successRate = successRate;
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getStudyHours() {
        return studyHours;
    }

    public void setStudyHours(String studyHours) {
        this.studyHours = studyHours;
    }

    public String getTeacher() {
        return teacher;
    }

    public void setTeacher(String teacher) {
        this.teacher = teacher;
    }

    public String getSchedule() {
        return schedule;
    }

    public void setSchedule(String schedule) {
        this.schedule = schedule;
    }

    public String getSuccessRate() {
        return successRate;
    }

    public void setSuccessRate(String successRate) {
```

```

        this.successRate = successRate;
    }

    private String studyHours;
    private String teacher;
    private String schedule;
    private String successRate;

    // Constructors, getters, setters, and other methods go here
}

```

## SubjectAdapter

```

package com.example.lab5_sql;

import android.annotation.SuppressLint;
import android.graphics.Color;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class SubjectAdapter extends RecyclerView.Adapter<SubjectAdapter.SubjectViewHolder> {
    private List<Subject> subjectList;
    private int selectedItem = -1;

    public int getSelectedItemPosition() {
        return selectedItem;
    }

    public void clearSelectedItem() {
        selectedItem = -1;
        notifyDataSetChanged();
    }

    Button deleteButton;
    public SubjectAdapter(List<Subject> subjectList) {
        this.subjectList = subjectList;
    }
    public void removeItem(int position) {
        subjectList.remove(position);
        notifyItemRemoved(position);
    }
    @NonNull
    @Override
    public SubjectViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.item, parent, false);
    }

```

```

        return new SubjectViewHolder(itemView);
    }

    @SuppressWarnings("SetTextI18n")
    @Override
    public void onBindViewHolder(@NonNull SubjectViewHolder holder, @SuppressWarnings("RecyclerView") int position) {
        Subject subject = subjectList.get(position);
        holder.subjectNameTextView.setText(subject.getName());
        holder.scopeTextView.setText("Обсяг: " + subject.getStudyHours());
        holder.schedule.setText("Розклад: \n" + subject.getSchedule());
        holder.rate.setText("Успішність " + subject.getSuccessRate());
        holder.teacher.setText("Викладач: " + subject.getTeacher());
        // Populate other fields as needed

        // Highlight the clicked item
        if (position == selectedItem) {
            holder.itemView.setBackgroundColor(ContextCompat.getColor(holder.itemView.getContext(), R.color.highlightColor));
        } else {
            holder.itemView.setBackgroundColor(Color.TRANSPARENT); // Reset background
        }
        // Handle item click
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                selectedItem = position;
                notifyDataSetChanged(); // Refresh the RecyclerView to apply highlighting
            }
        });
    }

    @Override
    public int getItemCount() {
        return subjectList.size();
    }

    public static class SubjectViewHolder extends RecyclerView.ViewHolder {
        TextView subjectNameTextView;
        TextView scopeTextView;

        TextView teacher;

        TextView rate;

        TextView schedule;
        // Define other TextViews for teacher, schedule, success rate

        public SubjectViewHolder(@NonNull View itemView) {
            super(itemView);
            subjectNameTextView = itemView.findViewById(R.id.textViewSubjectName);
            scopeTextView = itemView.findViewById(R.id.textViewScope);
            teacher = itemView.findViewById(R.id.textViewTeacher);
            rate = itemView.findViewById(R.id.textViewSuccessRate);
        }
    }

```

```

        schedule = itemView.findViewById(R.id.textViewSchedule);


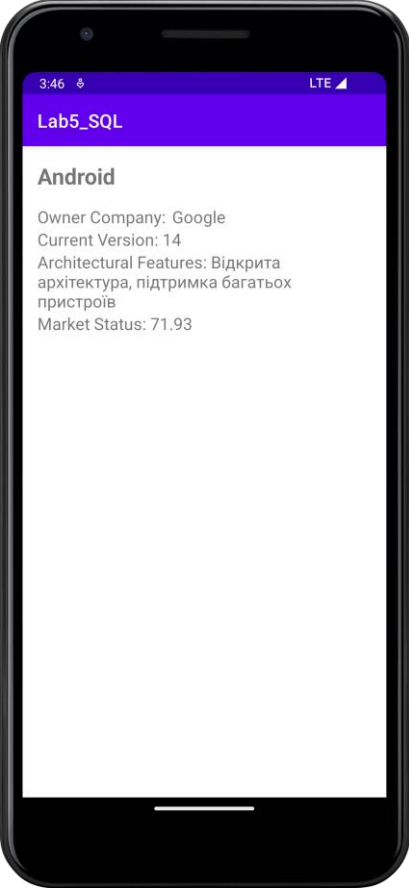
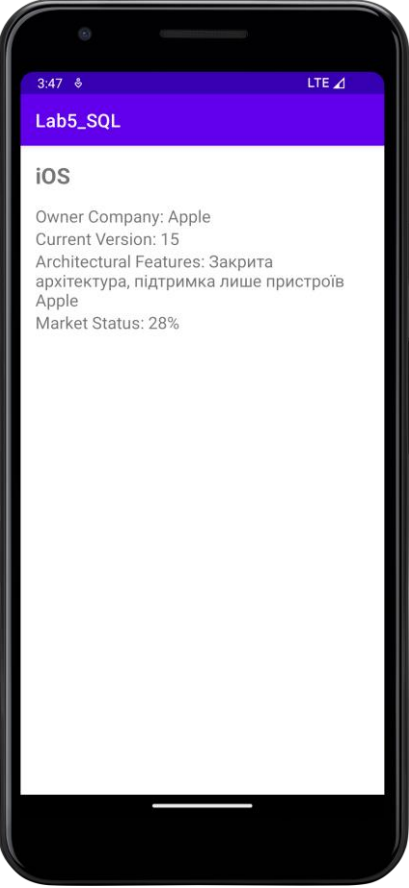
    }

}
}

```

### 3. Результати виконання тестової програми

#### 3.1. База даних сучасних мобільних операційних систем

 <p>Рис.1. Вигляд БД OS</p>	 <p>Рис.2. Інформація про Android при натисканні ПКМ на Android</p>	 <p>Рис.3. Інформація про IOS при натисканні ПКМ на IOS</p>
---	--	---

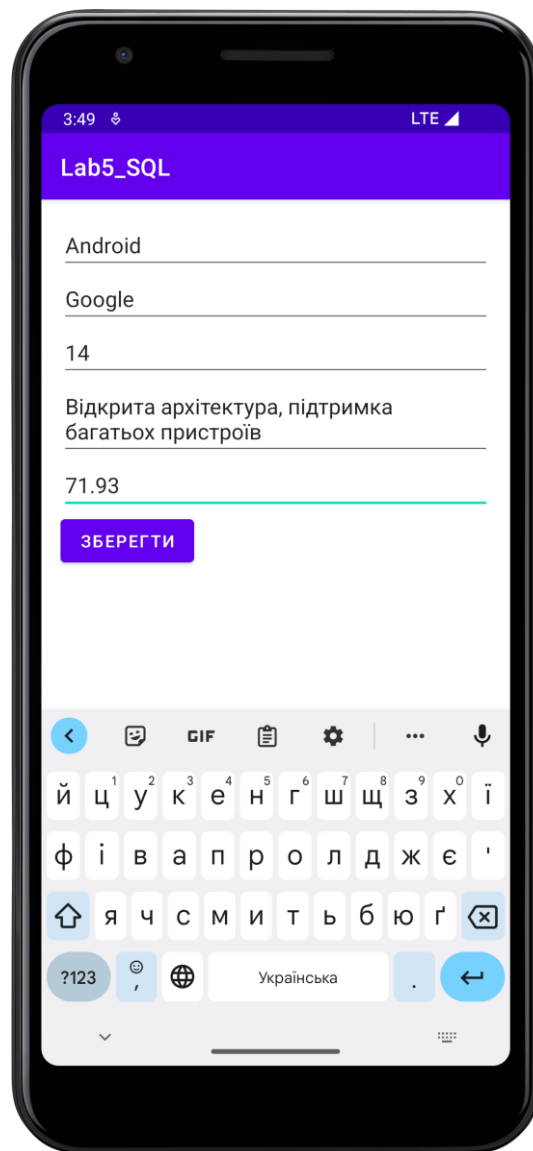


Рис.4. Додавання даних в БД про OS Android

### 3.2. База даних навчальних предметів біжучого семестру



Рис.5. Вигляд БД: Навчальні предмети

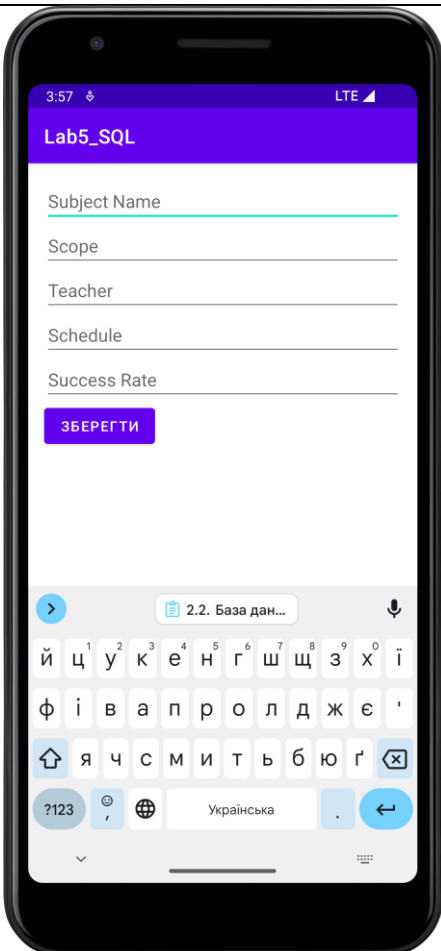


Рис.6. Вигляд вікна для додавання предмету

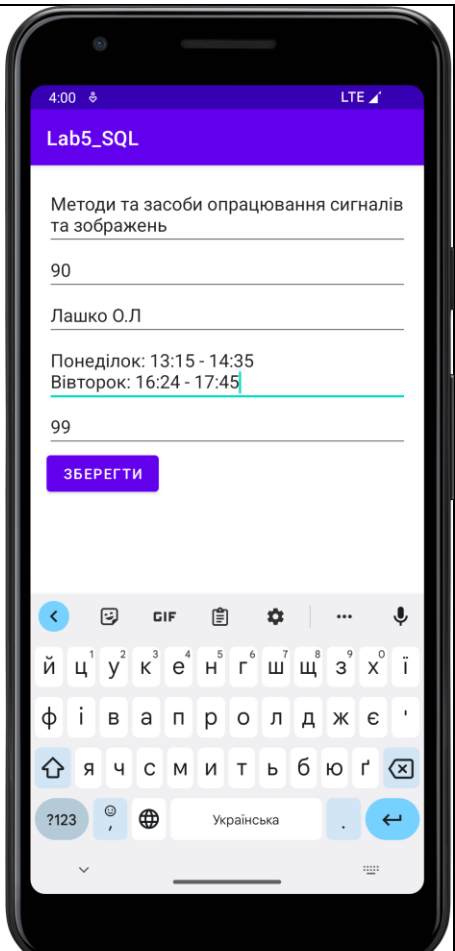


Рис.7. Додавання предмету

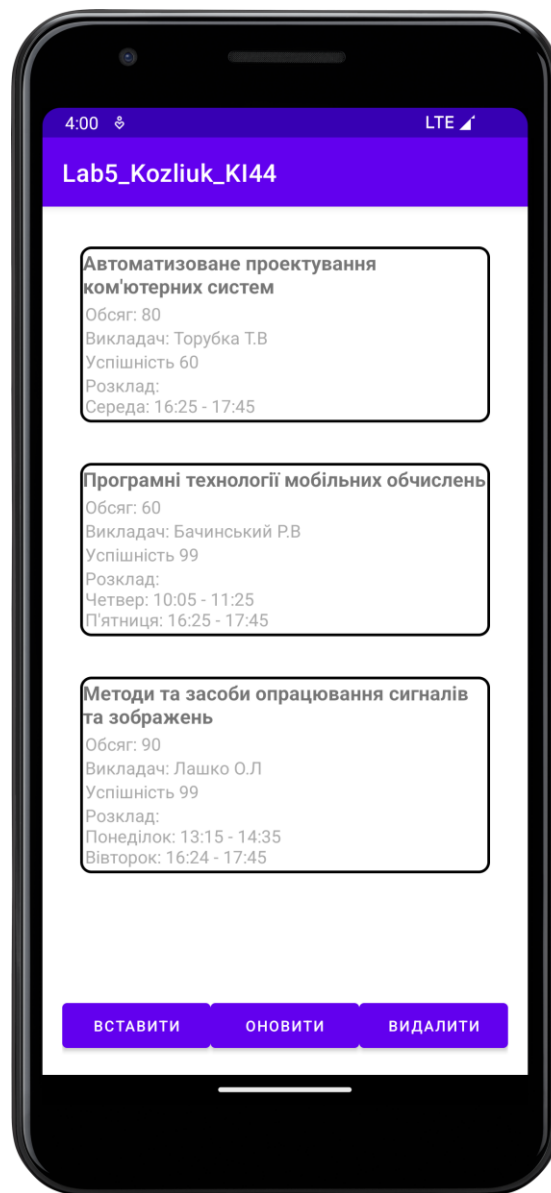


Рис.8. Вигляд БД після додавання нового предмету

**Висновок:** Під час виконання даної лабораторної роботи, я оволодів навичками роботи з СУБД SQLite в OS Android. Створив дві бази даних: база даних сучасних мобільних операційних систем й навчальних предметів біжучого семестру та тестові аплікації для взаємодії з БД.