Lab Cycle: 04 Date:

EXPERIMENT NO: 12

AIM: Develop application using Fragments.

Program:

MainActivity.java

```
package com.example.fragmentapp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    private Button button1, button2;
    private Fragment fragment1, fragment2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Find the views from the layout file.
        button1 = findViewById(R.id.button_fragment1);
        button2 = findViewById(R.id.button_fragment2);
        // Initialize the fragment objects.
        fragment1 = new FragmentOne();
        fragment2 = new FragmentTwo();
        // Set up click listeners for the buttons.
        button1.setOnClickListener(this);
        button2.setOnClickListener(this);
        // Initially load the first fragment into the container.
        loadFragment(fragment1);
    }

    @Override
    public void onClick(View v) {
        // Use a switch statement to handle button clicks.
        int viewId = v.getId();
        if (viewId == R.id.button_fragment1) {
            loadFragment(fragment1); // Load the first fragment.
        } else if (viewId == R.id.button_fragment2) {
            loadFragment(fragment2); // Load the second fragment.
        }
    }

    // A helper method to handle the transaction of loading a fragment.
    private void loadFragment(Fragment fragment) {
        // Get the FragmentManager to manage fragment transactions.
        FragmentManager fragmentManager = getSupportFragmentManager();
        // Start a transaction.
        FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
        // Replace the current fragment in the container with the new fragment.
        fragmentTransaction.replace(R.id.fragment_container, fragment);
        // Commit the transaction to apply the changes.
        fragmentTransaction.commit();
    }
}
```

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">

    <!-- This is the container where the fragments will be displayed. -->
    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@+id/button_fragment1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <!-- Button to load the first fragment. -->
    <Button
        android:id="@+id/button_fragment1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="4dp"
        android:text="Fragment One"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/button_fragment2"
        app:layout_constraintStart_toStartOf="parent" />
    <!-- Button to load the second fragment. -->
    <Button
        android:id="@+id/button_fragment2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="4dp"
        android:layout_marginEnd="8dp"
        android:text="Fragment Two"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/button_fragment1" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

FragmentOne.java

```
package com.example.fragmentapp;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

// This is the first fragment class.
public class FragmentOne extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
@Nullable Bundle savedInstanceState) {
        // Inflate the layout for this fragment from its XML file.
        return inflater.inflate(R.layout.fragment_one, container, false);
    }
}
```

fragment_one.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- A simple layout for the first fragment. -->
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFCDD2"
    tools:context=".FragmentOne">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="This is Fragment One"
        android:textSize="24sp"
        android:textColor="@android:color/black" />
</FrameLayout>
```

FragmentTwo.java

```
package com.example.fragmentapp;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

// This is the second fragment class.
public class FragmentTwo extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
@Nullable Bundle savedInstanceState) {
        // Inflate the layout for this fragment from its XML file.
        return inflater.inflate(R.layout.fragment_two, container, false);
    }
}
```

fragment_two.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- A simple layout for the second fragment. -->
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#BBDEFB"
    tools:context=".FragmentTwo">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="This is Fragment Two"
        android:textSize="24sp"
        android:textColor="@android:color/black" />
```

```
</FrameLayout>
```

OUTPUT:

RESULT:

The program was executed successfully and the output was verified.

Lab Cycle: 04 Date:

EXPERIMENT NO: 13

AIM: Create database using SQLite and perform INSERT, UPDATE, SELECT and DELETE.

Program:

DatabaseHelper.java

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;

public class DatabaseHelper extends SQLiteOpenHelper {

    // Database name and version
    private static final String DATABASE_NAME = "Student.db";
    private static final int DATABASE_VERSION = 1;
    // Table name and columns
    private static final String TABLE_NAME = "student_table";
    public static final String COL_1 = "ID";
    public static final String COL_2 = "NAME";
    public static final String COL_3 = "EMAIL";
    // Constructor
    public DatabaseHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    // This method is called when the database is created for the first time.
    @Override
    public void onCreate(SQLiteDatabase db) {
```

```
    // SQL statement to create the table
    String CREATE_TABLE = "CREATE TABLE " + TABLE_NAME + " (" +
            COL_1 + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            COL_2 + " TEXT, " +
            COL_3 + " TEXT)";
    db.execSQL(CREATE_TABLE);
}
// This method is called when the database needs to be upgraded.
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // Drop the old table if it exists
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    // Create a new one
    onCreate(db);
}
// Method to insert a new record (CREATE operation)
public boolean insertData(String name, String email) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COL_2, name);
    contentValues.put(COL_3, email);

    // The insert() method returns the row ID of the newly inserted row, or -1 if an error occurred
    long result = db.insert(TABLE_NAME, null, contentValues);
    return result != -1;
}
// Method to get all records from the database (READ operation)
public Cursor getAllData() {
    SQLiteDatabase db = this.getReadableDatabase();
    // The rawQuery() method executes a raw SQL query and returns a Cursor
    Cursor res = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
```

```
        return res;
    }
    // Method to update an existing record (UPDATE operation)
    public boolean updateData(String id, String name, String email) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1, id);
        contentValues.put(COL_2, name);
        contentValues.put(COL_3, email);

        // The update() method returns the number of rows affected
        db.update(TABLE_NAME, contentValues, "ID = ?", new String[]{id});
        return true;
    }
    // Method to delete a record (DELETE operation)
    public Integer deleteData(String id) {
        SQLiteDatabase db = this.getWritableDatabase();
        // The delete() method returns the number of rows affected
        return db.delete(TABLE_NAME, "ID = ?", new String[]{id});
    }
}
```

MainActivity.java

```
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {

    DatabaseHelper myDb;
    EditText editTextId, editTextName, editTextEmail;
    Button btnAdd, btnView, btnUpdate, btnDelete;

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

        // Create an instance of the DatabaseHelper
        myDb = new DatabaseHelper(this);

        // Initialize UI components
        editTextId = findViewById(R.id.editTextId);
        editTextName = findViewById(R.id.editTextName);
        editTextEmail = findViewById(R.id.editTextEmail);
        btnAdd = findViewById(R.id.btnAdd);
        btnView = findViewById(R.id.btnView);
        btnUpdate = findViewById(R.id.btnUpdate);
        btnDelete = findViewById(R.id.btnDelete);

        // Set up button listeners
        addData();
        viewAll();
        updateData();
        deleteData();
    }
    // Method to handle the "ADD" button click
    public void addData() {
        btnAdd.setOnClickListener(
                new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        boolean isInserted = myDb.insertData(
                                editTextName.getText().toString(),
                                editTextEmail.getText().toString()
                        );
                        if (isInserted) {
                            Toast.makeText(MainActivity.this, "Data Inserted Successfully!",
Toast.LENGTH_SHORT).show();
                        } else {
                            Toast.makeText(MainActivity.this, "Data Insertion Failed",
Toast.LENGTH_SHORT).show();
                        }
                    }
                }
        );
    }

    // Method to handle the "VIEW ALL" button click
    public void viewAll() {
        btnView.setOnClickListener(
                new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Cursor res = myDb.getAllData();
                        if (res.getCount() == 0) {
                            // Show message if no data is found
                            showMessage("Error", "Nothing found");
                            return;
                        }

                        StringBuffer buffer = new StringBuffer();
                        while (res.moveToNext()) {
                            buffer.append("ID: " + res.getString(0) + "\n");
                            buffer.append("Name: " + res.getString(1) + "\n");
                            buffer.append("Email: " + res.getString(2) + "\n\n");
                        }
                        // Show all data in a dialog
                        showMessage("Data", buffer.toString());
                    }
                }
        );
    }

    // Method to handle the "UPDATE" button click
    public void updateData() {
        btnUpdate.setOnClickListener(
                new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        boolean isUpdated = myDb.updateData(
                                editTextId.getText().toString(),
                                editTextName.getText().toString(),
                                editTextEmail.getText().toString()
                        );
                        if (isUpdated) {
                            Toast.makeText(MainActivity.this, "Data Updated Successfully!",
Toast.LENGTH_SHORT).show();
                        } else {
                            Toast.makeText(MainActivity.this, "Data Update Failed",
Toast.LENGTH_SHORT).show();
                        }
                    }
                }
        );
    }

    // Method to handle the "DELETE" button click
    public void deleteData() {
        btnDelete.setOnClickListener(
                new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Integer deletedRows = myDb.deleteData(editTextId.getText().toString());
                        if (deletedRows > 0) {
                            Toast.makeText(MainActivity.this, "Data Deleted Successfully!",
Toast.LENGTH_SHORT).show();
                        } else {
                            Toast.makeText(MainActivity.this, "Data Deletion Failed",
Toast.LENGTH_SHORT).show();
                        }
                    }
                }
        );
    }

    // Generic method to show a message dialog
    public void showMessage(String title, String message) {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="[http://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)"
    xmlns:app="[http://schemas.android.com/apk/res-auto](http://schemas.android.com/apk/res-auto)"
    xmlns:tools="[http://schemas.android.com/tools](http://schemas.android.com/tools)"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:padding="16dp">

    <TextView
        android:id="@+id/textViewTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Student Database"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="24dp"/>

    <EditText
        android:id="@+id/editTextId"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="ID"
        android:inputType="number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textViewTitle"
        android:layout_marginTop="16dp"/>

    <EditText
        android:id="@+id/editTextName"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editTextId"
        android:layout_marginTop="8dp"/>

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editTextName"
        android:layout_marginTop="8dp"/>

    <Button
        android:id="@+id/btnAdd"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Add Data"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editTextEmail"
        android:layout_marginTop="24dp"/>

    <Button
        android:id="@+id/btnView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="View All Data"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/btnAdd"
        android:layout_marginTop="8dp"/>

    <Button
        android:id="@+id/btnUpdate"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Update Data"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/btnView"
        android:layout_marginTop="8dp"/>

    <Button
        android:id="@+id/btnDelete"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
```

```
        android:text="Delete Data"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/btnUpdate"
        android:layout_marginTop="8dp"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:

RESULT:

The program was executed successfully and the output was verified.