MUMBAI EDUCATIONAL TRUST

MET INSTITUTE OF COMPUTER SCIENCE

Program Number	18
Roll Number	1545
Title of program	**CRUD OPERATION IN ANDROID**
Program	implementing CRUD (Create, Read, Update, Delete) operations using SQLite in Android Studio with Java.

Source Code:

DatabaseHelper.java

```
package com.example.sqlitecrudapp;

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

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "UserDB.db";
    private static final int DATABASE_VERSION = 1;

    private static final String TABLE_NAME = "users";
    private static final String COLUMN_ID = "id";
    private static final String COLUMN_NAME = "name";
    private static final String COLUMN_EMAIL = "email";

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

    // Create table
    @Override
```

```
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME + " (" +
                COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COLUMN_NAME + " TEXT, " +
                COLUMN_EMAIL + " TEXT)";
        db.execSQL(createTable);
    }

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

    // CREATE
    public boolean addUser(String name, String email) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COLUMN_NAME, name);
        values.put(COLUMN_EMAIL, email);

        long result = db.insert(TABLE_NAME, null, values);
        db.close();
        return result != -1;
    }

    // READ all records
    public Cursor getAllUsers() {
        SQLiteDatabase db = this.getReadableDatabase();
        return db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
    }

    // UPDATE
    public boolean updateUser(int id, String name, String email) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COLUMN_NAME, name);
        values.put(COLUMN_EMAIL, email);

        int rows = db.update(TABLE_NAME, values, COLUMN_ID + "=?", new
String[]{String.valueOf(id)});
        db.close();
        return rows > 0;
```

```
    }

    // DELETE
    public boolean deleteUser(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        int rows = db.delete(TABLE_NAME, COLUMN_ID + "=?", new
  String[]{String.valueOf(id)});
        db.close();
        return rows > 0;
    }
  }
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="16dp"
        android:orientation="vertical">

        <EditText
            android:id="@+id/editTextId"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter ID (for update/delete)"
            android:inputType="number" />

        <EditText
            android:id="@+id/editTextName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter Name"
            android:layout_marginTop="8dp" />

        <EditText
            android:id="@+id/editTextEmail"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Email"
        android:inputType="textEmailAddress"
        android:layout_marginTop="8dp" />

    <Button
        android:id="@+id/buttonAdd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Add"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/buttonView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="View All"
        android:layout_marginTop="8dp" />

    <Button
        android:id="@+id/buttonUpdate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Update"
        android:layout_marginTop="8dp" />

    <Button
        android:id="@+id/buttonDelete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delete"
        android:layout_marginTop="8dp" />

    <TextView
        android:id="@+id/textViewResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Results will appear here"
        android:layout_marginTop="16dp" />

  </LinearLayout>
</ScrollView>
```

MainActivity.java

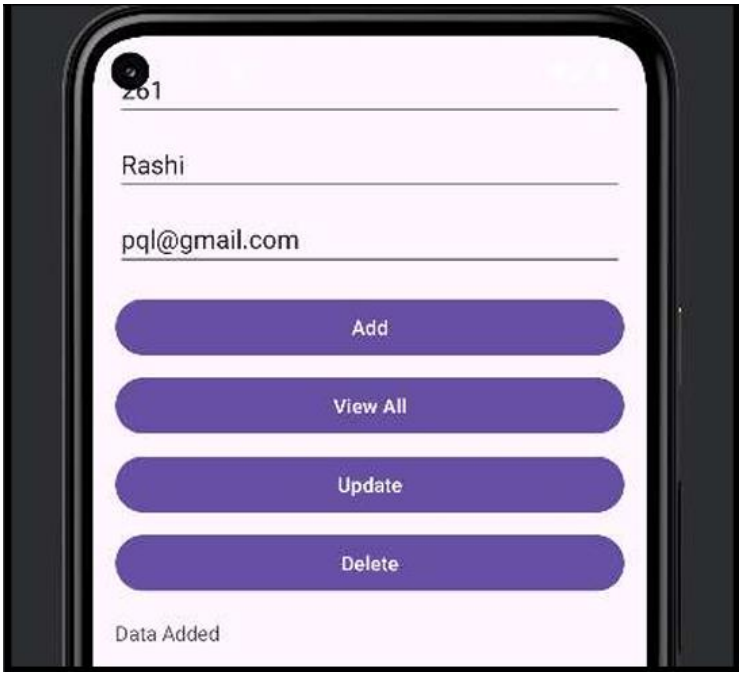
```
package com.example.sqlitecrudapp;

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.TextView;

public class MainActivity extends AppCompatActivity {

    EditText editTextId, editTextName, editTextEmail;
    Button buttonAdd, buttonView, buttonUpdate, buttonDelete;
    TextView textViewResult;
    DatabaseHelper dbHelper;

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

        editTextId = findViewById(R.id.editTextId);
        editTextName = findViewById(R.id.editTextName);
        editTextEmail = findViewById(R.id.editTextEmail);

        buttonAdd = findViewById(R.id.buttonAdd);
        buttonView = findViewById(R.id.buttonView);
        buttonUpdate = findViewById(R.id.buttonUpdate);
        buttonDelete = findViewById(R.id.buttonDelete);

        textViewResult = findViewById(R.id.textViewResult);

        dbHelper = new DatabaseHelper(this);

        buttonAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String name = editTextName.getText().toString().trim();
                String email = editTextEmail.getText().toString().trim();
```

```
            if (!name.isEmpty() && !email.isEmpty()) {
                boolean inserted = dbHelper.addUser(name, email);
                textViewResult.setText(inserted ? "Data Added" : "Insert Failed");
            } else {
                textViewResult.setText("Please fill name and email");
            }
        }
    });

    buttonView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Cursor cursor = dbHelper.getAllUsers();

            if (cursor.getCount() == 0) {
                textViewResult.setText("No data found");
                return;
            }

            StringBuilder builder = new StringBuilder();
            while (cursor.moveToNext()) {
                builder.append("ID: ").append(cursor.getInt(0)).append("\n");
                builder.append("Name: ").append(cursor.getString(1)).append("\n");
                builder.append("Email:
").append(cursor.getString(2)).append("\n\n");
            }

            textViewResult.setText(builder.toString());
            cursor.close();
        }
    });

    buttonUpdate.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String idStr = editTextId.getText().toString().trim();
            String name = editTextName.getText().toString().trim();
            String email = editTextEmail.getText().toString().trim();

            if (!idStr.isEmpty() && !name.isEmpty() && !email.isEmpty()) {
                int id = Integer.parseInt(idStr);
```

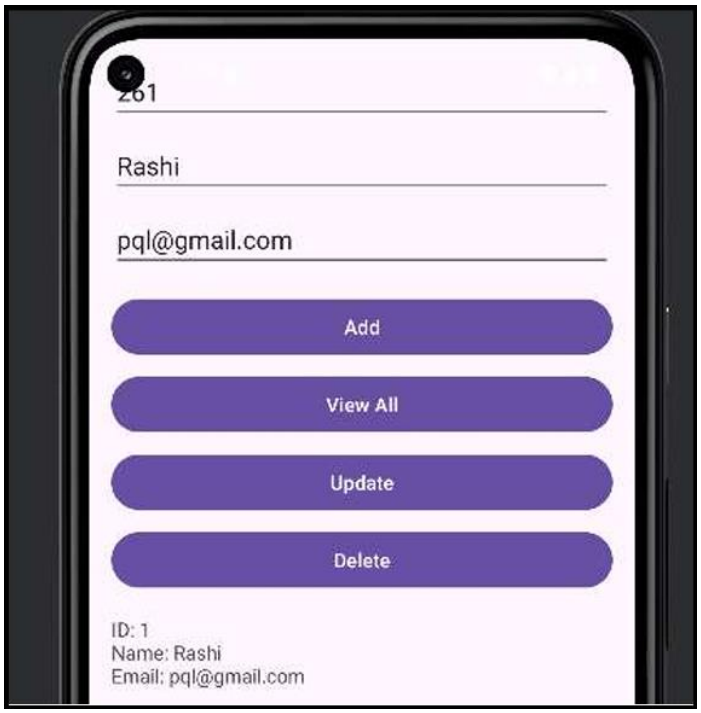
```
            boolean updated = dbHelper.updateUser(id, name, email);
            textViewResult.setText(updated ? "Data Updated" : "Update Failed");
          } else {
            textViewResult.setText("Please fill ID, name, and email");
          }
        }
      });

      buttonDelete.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
          String idStr = editTextId.getText().toString().trim();

          if (!idStr.isEmpty()) {
            int id = Integer.parseInt(idStr);
            boolean deleted = dbHelper.deleteUser(id);
            textViewResult.setText(deleted ? "Data Deleted" : "Delete Failed");
          } else {
            textViewResult.setText("Please enter ID to delete");
          }
        }
      });
    }
}
```

Output:

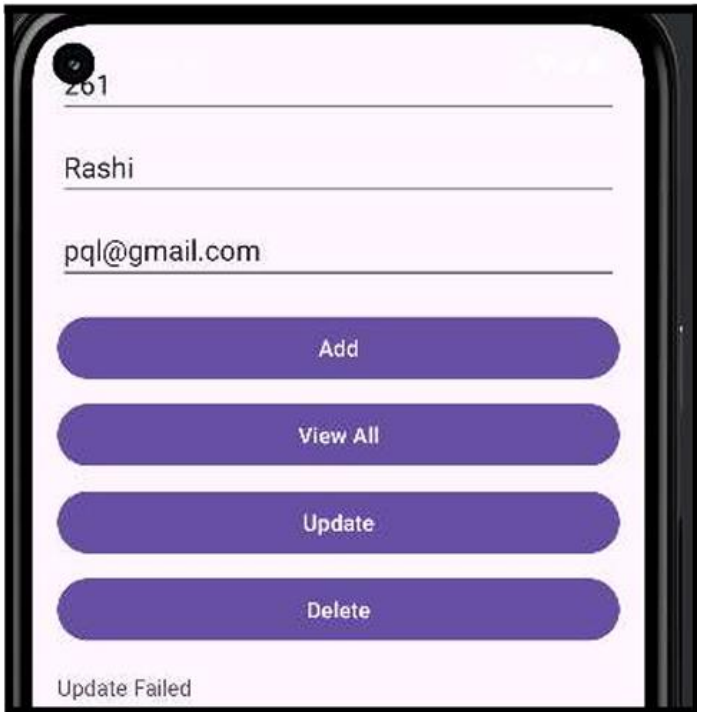
1. CREAT



2. READ



3. UPDATE



4. DELETE