Practical No. 5

Title: Android program to perform CRUD operation using SQLite DB

Aim: Create an application to demonstrate CRUD operations using SQLite DB

Introduction

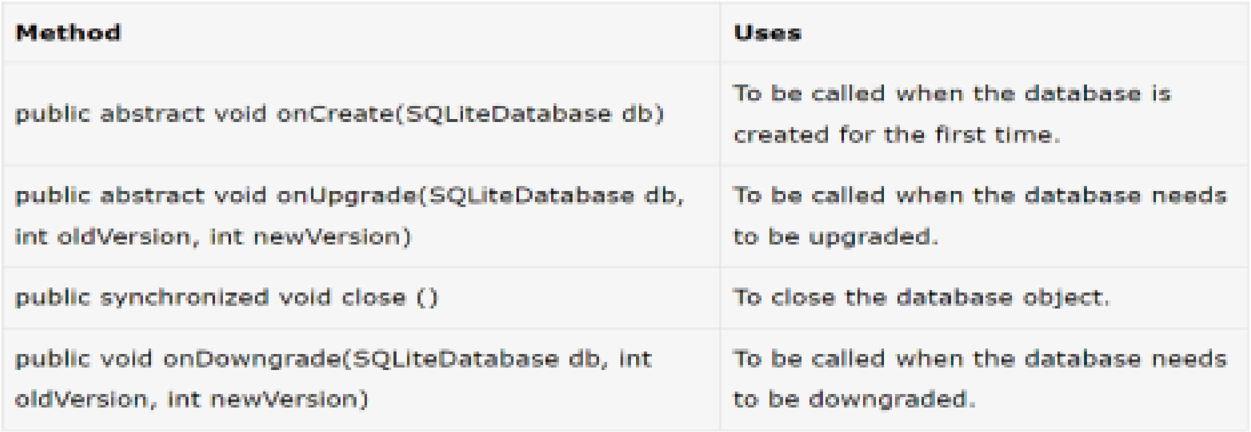
What is SQLite?

SQLite is another data storage available in Android where we can store data in the user's device and can use it any time when required. We can perform so many operations on this data such as adding new data, updating, reading, and deleting this data. SQLite is an offline database that is locally stored in the user's device and we do not have to create any connection to connect to this database.

SQLiteOpenHelper class:

The functionality to use the SQLite database is provided by the android.database.sqlite.SQLiteOpenHelper class which is used for database creation and version management. The implementation of onCreate() and onUpgrade() methods of SQLiteOpenHelper class is required to be provided to perform any database operation.

The SQLiteOpenHelper class has many methods. Some of the important methods of the SQLiteOpenHelper class are:



Method	Uses
public abstract void onCreate(SQLiteDatabase db)	To be called when the database is created for the first time.
public abstract void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)	To be called when the database needs to be upgraded.
public synchronized void close ()	To close the database object.
public void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion)	To be called when the database needs to be downgraded.

What is CRUD?

CRUD is nothing but an abbreviation for the basic operations that we perform in any database. And the operations are

- **Create**
- **Read**
- **Update**
- **Delete**

Exercise - Create a database and perform CRUD operations on it

Implementation:
Program:

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:id="@+id/main"
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">


<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_marginStart="50dp"
  android:layout_marginTop="20dp"
  android:layout_marginBottom="10dp"
  android:text="@string/roll_no"
  android:textSize="20sp"
  android:textStyle="bold" />

<EditText
  android:id="@+id/rollNo"
  android:layout_width="250dp"
  android:layout_height="48dp"
  android:layout_marginStart="50dp"
  android:layout_marginBottom="10dp"
  android:hint="@string/enter_your_roll_number"
  android:inputType="number"
  android:paddingStart="20dp"


  />

<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/name"
  android:textStyle="bold"
  android:layout_marginBottom="10dp"
  android:layout_marginStart="50dp"
  android:textSize="20sp" />
<EditText
  android:layout_width="250dp"
  android:layout_height="50dp"
```

```
    android:hint="@string/enter_your_name"
    android:id="@+id/txtName"
    android:layout_marginBottom="10dp"
    android:layout_marginStart="50dp"
    />
  <TextView
    android:layout_marginBottom="10dp"
    android:layout_marginStart="50dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/email_id"
    android:textStyle="bold"
    android:textSize="20sp" />
  <EditText
    android:layout_width="250dp"
    android:layout_height="50dp"
    android:hint="@string/enter_your_email"
    android:id="@+id/txtEmail"
    android:layout_marginBottom="10dp"
    android:layout_marginStart="50dp"
    />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/course"
    android:layout_marginBottom="10dp"
    android:layout_marginStart="50dp"
    android:textStyle="bold"
    android:textSize="20sp" />
  <EditText
    android:layout_width="250dp"
    android:layout_height="50dp"
    android:hint="@string/enter_your_course"
    android:id="@+id/txtCourse"
    android:layout_marginBottom="10dp"
    android:layout_marginStart="50dp"
    />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:text="@string/contact"
        android:layout_marginBottom="10dp"
        android:layout_marginStart="50dp"
        android:textStyle="bold"
        android:textSize="20sp" />
    <EditText
        android:layout_width="250dp"
        android:layout_height="50dp"
        android:layout_marginBottom="10dp"
        android:layout_marginStart="50dp"
        android:id="@+id/txtcontact"
        android:hint="@string/enter_your_contact_number"
        />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/address"
        android:layout_marginBottom="10dp"
        android:layout_marginStart="50dp"
        android:textStyle="bold"
        android:textSize="20sp" />
    <EditText
        android:layout_width="250dp"
        android:layout_height="50dp"
        android:hint="@string/enter_your_address"
        android:id="@+id/txtAddress"
        android:layout_marginBottom="10dp"
        android:layout_marginStart="50dp"
        />
    <LinearLayout
        style="?android:attr/buttonBarStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp">

        <Button
            style="?android:attr/buttonBarButtonStyle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/read"
            android:onClick="loadStudents"
            android:id="@+id/BtnRead"
            android:layout_weight="1" />

        <Button
            style="?android:attr/buttonBarButtonStyle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/insert"
            android:onClick="addStudents"
            android:id="@+id/BtnInsert"
            android:layout_weight="1" />

        <Button
            style="?android:attr/buttonBarButtonStyle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/delete"
            android:onClick="deleteStudents"
            android:id="@+id/BtnDelete"
            android:layout_weight="1" />

        <Button
            style="?android:attr/buttonBarButtonStyle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/update"
            android:onClick="updateStudents"
            android:id="@+id/BtnUpdate"
            android:layout_weight="1" />

    </LinearLayout>

    <TextView
        android:id="@+id/result"
        android:layout_width="185dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:hint="@string/result" />
```

```
</LinearLayout>
```

Student.java:

```
        package com.example.database;

public class Student {
  private  int rno, contact;
  private String name, email, course, address;
  Student(){}
  Student(int rno, String name, String email, int contact, String course, String address){
     this.rno = rno;
     this.name = name;
     this.email = email;
     this.contact = contact;
     this.course = course;
     this.address = address;
  }

   int getId() {return this.rno;}

   void setId(int id) {this.rno = id;}


   String getName() {return this.name;}

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


   String getEmail() {return this.email;}

   void setEmail(String email) {this.email = email;}


   int getContact() {return this.contact;}

   void setContact(int contact) {this.contact = contact;}


   String getCourse() {return this.course;}
```

```
    void setCourse(String course) {this.course = course;}


    String getAddress() {return this.address;}

    void setAddress(String address) {this.address = address;}

}
```

MyDbHandler.java:

```
package com.example.database;

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

public class MyDBHandler extends SQLiteOpenHelper {

   private static final int DATABASE_VERSION = 3;
   private static final String DATABASE_NAME = "studentDB.db";
   private static final String TABLE_STUDENT = "newstudent";
   private static final String COLUMN_RNO = "rno";
   private static final String COLUMN_NAME = "name";
   private static final String COLUMN_EMAIL = "email";
   private static final String COLUMN_COURSE = "course";
   private static final String COLUMN_CONTACT = "contact";
   private static final String COLUMN_ADDRESS = "address";

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


   @Override
   public void onCreate(SQLiteDatabase db) {
          String CREATE_STUDENT_TABLE = "CREATE TABLE " + TABLE_STUDENT + "(" +
COLUMN_RNO + " INTEGER PRIMARY KEY, " + COLUMN_NAME
           + " TEXT, " + COLUMN_EMAIL + " TEXT, " +
                COLUMN_COURSE + " TEXT, " + COLUMN_CONTACT + " INTEGER, " +
COLUMN_ADDRESS + " TEXT" + ")";
    db.execSQL(CREATE_STUDENT_TABLE);
  }

  @Override
  public void onUpgrade(SQLiteDatabase db, int i, int i1) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_STUDENT);
    onCreate(db);
  }

  //load data
  String loadHandler() {
    StringBuilder result = new StringBuilder();
    String query = "Select * From " + TABLE_STUDENT;
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(query, null);
    while (cursor.moveToNext()) {
      int result_0 = cursor.getInt(0);
      String result_1 = cursor.getString(1);
      String result_2 = cursor.getString(2);
      String result_3 = cursor.getString(3);
      int result_4 = cursor.getInt(4);
      String result_5 = cursor.getString(5);
                        result.append(result_0).append("    ").append(result_1).append("
").append(result_2).append("    ").append(result_3).append("    ").append(result_4).append("
").append(result_5); //append("\n");
      System.lineSeparator();
    }
    cursor.close();
    db.close();
    if (result.toString().isEmpty())
      result = new StringBuilder("No Records Found");
    return result.toString();
  }

  //add data
  long addHandler(Student student) {
    long id;
    ContentValues values = new ContentValues();
```

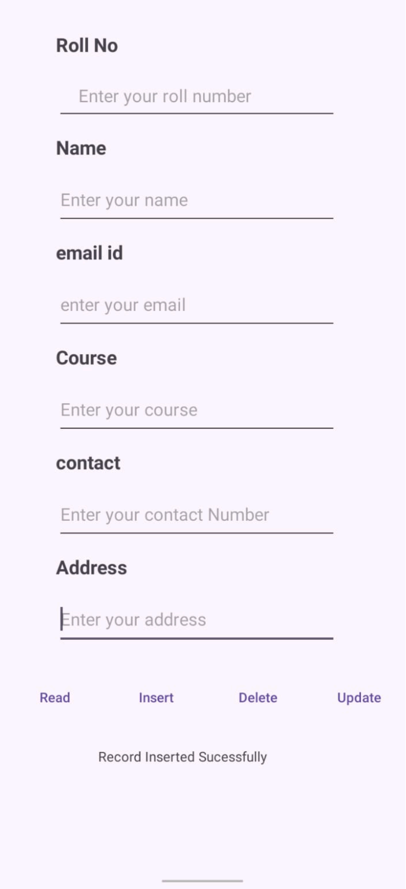
```
        values.put(COLUMN_RNO, student.getId());
        values.put(COLUMN_NAME, student.getName());
        values.put(COLUMN_EMAIL, student.getEmail());
        values.put(COLUMN_COURSE, student.getCourse());
        values.put(COLUMN_CONTACT, student.getContact());
        values.put(COLUMN_ADDRESS, student.getAddress());
        SQLiteDatabase db = this.getWritableDatabase();
        id = db.insert(TABLE_STUDENT, null, values);
        db.close();
        return id;
    }

    //update handler
      boolean updateHandler(int rno, String name, String email, String course, int contact,
String address){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues args = new ContentValues();
        args.put(COLUMN_RNO, rno);
        args.put(COLUMN_NAME, name);
        args.put(COLUMN_EMAIL, email);
        args.put(COLUMN_COURSE, course);
        args.put(COLUMN_CONTACT, contact);
        args.put(COLUMN_ADDRESS, address);
        return db.update(TABLE_STUDENT, args, COLUMN_RNO + "=" +rno, null) > 0;
    }
    //delete handler
    boolean deleteHandler(int rno) {
        boolean result = false;
         String query = "Select * From " + TABLE_STUDENT + " Where " + COLUMN_RNO + "='" +
rno + "'";
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(query, null);
        Student student = new Student();
        if (cursor.moveToFirst()) {
            student.setId(Integer.parseInt(cursor.getString(0)));
                                                db.delete(TABLE_STUDENT, COLUMN_RNO + "=?" , new
String[]{String.valueOf(student.getId())});
            cursor.close();
            result = true;
        }
        return result;
    }
}
```

MainActivity.java:

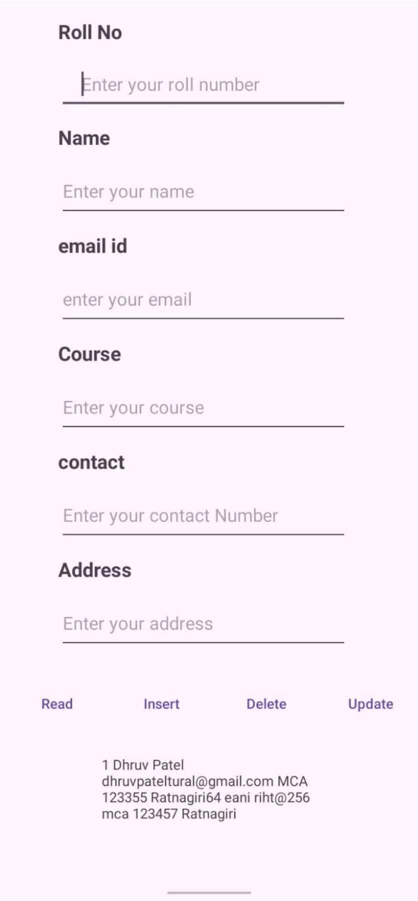
```
package com.example.database;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    EditText rno, name, email, contact, course, address;
    TextView output;
    MyDBHandler dbHandler;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) ->
{
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
        rno = findViewById(R.id.rollNo);
        name = findViewById(R.id.txtName);
        email = findViewById(R.id.txtEmail);
        contact = findViewById(R.id.txtcontact);
        course = findViewById(R.id.txtCourse);
        address = findViewById(R.id.txtAddress);
        output = findViewById(R.id.result);
        dbHandler = new MyDBHandler(this);
    }

    public void loadStudents(View view){
        output.setText(dbHandler.loadHandler());
        rno.setText("");
        name.setText("");
        email.setText("");
        contact.setText("");
        course.setText("");
        address.setText("");
    }

    //insert method
    public void addStudents(View view){
          if(!rno.getText().toString().isEmpty()  &&  !name.getText().toString().isEmpty()  &&
!email.getText().toString().isEmpty()     &&     !contact.getText().toString().isEmpty()     &&
!course.getText().toString().isEmpty() && !address.getText().toString().isEmpty()){
            int id = Integer.parseInt(rno.getText().toString());
            String nm = name.getText().toString();
            String em = email.getText().toString();
            String ad = address.getText().toString();
            String cr = course.getText().toString();
            int cn = Integer.parseInt(contact.getText().toString());
            Student student = new Student(id, nm, em, cn, cr, ad);
            long insertId = dbHandler.addHandler(student);
            output.setText(R.string.record_inserted_sucessfully);
            if(insertId == -1){
          output.setText(R.string.record_already_exists);
            }
            else{
          rno.setText("");
          name.setText("");
          email.setText("");
          contact.setText("");
          course.setText("");
          address.setText("");
            }
        }
        else{
      output.setText(R.string.please_enter_all_feilds);

    }
  }
  //update method
  public void updateStudents(View view){
        if(!rno.getText().toString().isEmpty()  &&  !name.getText().toString().isEmpty()  &&
!email.getText().toString().isEmpty()     &&     !contact.getText().toString().isEmpty()     &&
!course.getText().toString().isEmpty() && !address.getText().toString().isEmpty()) {
                                                                boolean       result       =
dbHandler.updateHandler(Integer.parseInt(rno.getText().toString()),
name.getText().toString(),                  email.getText().toString(),course.getText().toString(),
Integer.parseInt(contact.getText().toString()) , address.getText().toString());
      if(result){
        rno.setText("");
        name.setText("");
        email.setText("");
        contact.setText("");
        course.setText("");
        address.setText("");
        output.setText(R.string.record_updated_sucessfully);

      }
      else{
        output.setText(R.string.record_not_found);
      }
    }
    else{
      output.setText(R.string.please_enter_all_feilds);
    }
  }

  //delete record
  public void deleteStudents(View view){
    if(!rno.getText().toString().isEmpty()){
                                                                boolean       result       =
dbHandler.deleteHandler(Integer.parseInt(rno.getText().toString()));
      if(result){
        rno.setText("");
        name.setText("");
```

```
            email.setText("");
            contact.setText("");
            course.setText("");
            address.setText("");
            output.setText(R.string.record_deleted_sucessfully);
        }
        else{
            output.setText(R.string.record_not_found);
        }
    }
    else{
        output.setText(R.string.please_enter_all_feilds);
    }
}

@Override
public void onDestroy() {
    super.onDestroy();
    dbHandler.close();
}


}
```

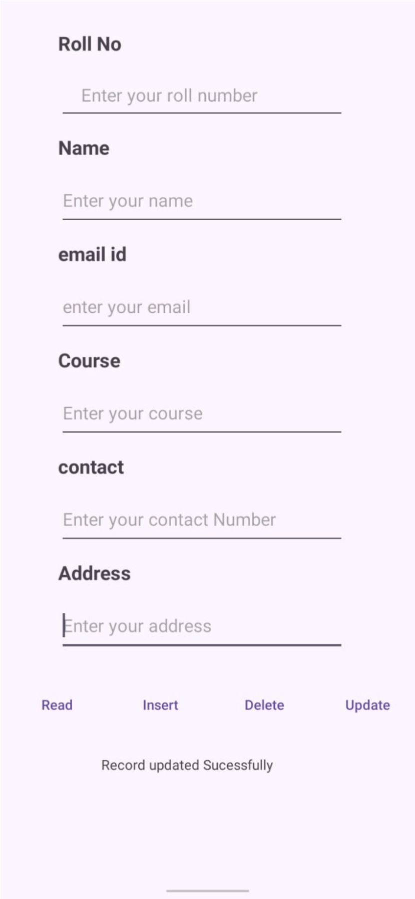
Output:



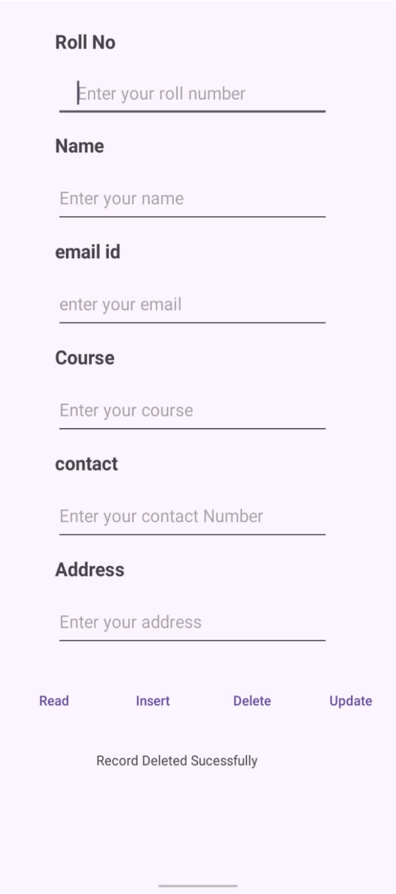
```
Record Inserted Sucessfully
```



```
1 Dhruv Patel
dhruvpateltural@gmail.com MCA
123355 Ratnagiri64 eani riht@256
mca 123457 Ratnagiri
```



Record updated Sucessfully



Record Deleted Sucessfully

Conclusion - This practical explored the basics of SQLite database operations in Android. We learned how to create a database, insert, retrieve, update, and delete data using the SQLiteOpenHelper class. This knowledge is essential for building Android apps that require local data storage.