# **Name: Abdurrahman Qureshi**

# **Roll No: 210451**

Practical No: 26

1. Write a program to perform CRUD operation in SQLite

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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"  
 android:orientation="vertical">  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/id"  
 android:layout\_marginHorizontal="30dp"  
 android:hint="Enter your ID"  
 android:layout\_marginTop="100dp"/>  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/name"  
 android:layout\_marginHorizontal="30dp"  
 android:hint="Enter your Name"  
 android:layout\_marginTop="30dp"/>  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/rollno"  
 android:layout\_marginHorizontal="30dp"  
 android:hint="Enter your Roll No"  
 android:layout\_marginTop="30dp"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/insert"  
 android:layout\_gravity="center"  
 android:text="Insert "  
 android:layout\_marginTop="100dp"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/update"  
 android:layout\_gravity="center"  
 android:text="Update "  
 android:layout\_marginTop="22dp"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/delete"  
 android:layout\_gravity="center"  
 android:text="Delete "  
 android:layout\_marginTop="22dp"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/get"  
 android:layout\_gravity="center"  
 android:text="Get "  
 android:layout\_marginTop="22dp"/>  
</LinearLayout>

MainActivity.java

package com.example.mad\_playground;  
  
import android.database.Cursor;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
 CRUD myDb;  
 EditText rollno, name, id;  
 Button insert, update, delete, get;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 myDb = new CRUD(this);  
 insert = findViewById(R.id.*insert*);  
 update = findViewById(R.id.*update*);  
 delete = findViewById(R.id.*delete*);  
 get = findViewById(R.id.*get*);  
 name = findViewById(R.id.*name*);  
 rollno = findViewById(R.id.*rollno*);  
 id = findViewById(R.id.*id*);  
 insert.setOnClickListener(view -> insertdata());  
 update.setOnClickListener(view -> updateData());  
 delete.setOnClickListener(view -> deleteData());  
 get.setOnClickListener(view -> getdata());  
 }  
  
 private void updateData() {  
 boolean updated = myDb.updateData(id.getText().toString(), rollno.getText().toString(), name.getText().toString());  
 if (updated)  
 Toast.*makeText*(this, "Data Updated Successfully !!!", Toast.*LENGTH\_SHORT*).show();  
 else  
 Toast.*makeText*(this, "Unable to Update Data...", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 private void deleteData() {  
 myDb.deleteData();  
 Toast.*makeText*(this, "Data Deleted Successfully !!!", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 private void insertdata() {  
 boolean inserted = myDb.insertData(id.getText().toString(), rollno.getText().toString(), name.getText().toString());  
 if (inserted)  
 Toast.*makeText*(this, "Data Inserted Successfully !!!", Toast.*LENGTH\_SHORT*).show();  
 else  
 Toast.*makeText*(this, "Unable to Insert Data...", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 private void getdata() {  
 Cursor resultSet = myDb.getData();  
 if (resultSet.getCount() == 0) {  
 showMessage("ERROR", "No Data Found...");  
 return;  
 }  
 StringBuilder buffer = new StringBuilder();  
 while (resultSet.moveToNext()) {  
 buffer.append("ID: ").append(resultSet.getInt(0)).append("\n");  
 buffer.append("Roll No: ").append(resultSet.getString(1)).append("\n");  
 buffer.append("Name: ").append(resultSet.getString(2)).append("\n");  
 }  
 showMessage("DATA", buffer.toString());  
 }  
  
 private void showMessage(String title, String message) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setCancelable(true);  
 builder.setTitle(title);  
 builder.setMessage(message);  
 builder.show();  
 }  
}

CRUD.java

package com.example.mad\_playground;  
  
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 CRUD extends SQLiteOpenHelper {  
 public static final String *DATABASE\_NAME* = "Student.db";  
 public static final String *TABLE\_NAME* = "Student\_info";  
 public static final String *COL\_1* = "ID";  
 public static final String *COL\_2* = "Roll\_No";  
 public static final String *COL\_3* = "Name";  
  
 public CRUD(@Nullable Context context) {  
 super(context, *DATABASE\_NAME*, null, 1);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase sqLiteDatabase) {  
 sqLiteDatabase.execSQL("CREATE TABLE " + *TABLE\_NAME* + " (ID INTEGER PRIMARY KEY AUTOINCREMENT, ROLL\_NO TEXT, NAME TEXT)");  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {  
 sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME*);  
 onCreate(sqLiteDatabase);  
 }  
  
 public boolean insertData(String roll\_no, String name, String s) {  
 SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();  
 ContentValues contentValues = new ContentValues();  
 contentValues.put(*COL\_2*, roll\_no);  
 contentValues.put(*COL\_3*, name);  
 long result = sqLiteDatabase.insert(*TABLE\_NAME*, null, contentValues);  
 return result != -1;  
 }  
  
 public Cursor getData() {  
 SQLiteDatabase sqLiteDatabase = this.getReadableDatabase();  
 return sqLiteDatabase.rawQuery("SELECT \* FROM " + *TABLE\_NAME*, null);  
 }  
  
 public void deleteData() {  
 SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();  
 sqLiteDatabase.execSQL("DELETE FROM " + *TABLE\_NAME*);  
 }  
  
 public boolean updateData(String id, String roll\_no, String name) {  
 SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();  
 ContentValues contentValues = new ContentValues();  
 contentValues.put(*COL\_2*, roll\_no);  
 contentValues.put(*COL\_3*, name);  
 int rowsAffected = sqLiteDatabase.update(*TABLE\_NAME*, contentValues, *COL\_1* + " = ?", new String[]{id});  
 return rowsAffected > 0;  
 }  
}

OUTPUT:

 

 