Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 4**	
Title of Lab Assignment: Android program to perform CRUD operation using SQLite DB (create table students with fields roll no, name, email-Id, course, perform add, update and delete record operations).		
DOP: 06-10-2024	**DOS: 07-10-2024**	
CO Mapped:	**PO Mapped:**	**Signature:**

Practical No. 4

Aim: Android program to perform CRUD operation using SQLite DB (create table students with fields roll no, name, email-Id, course, perform add, update and delete record operations).

Theory:

SQLite is a lightweight, embedded relational database management system that is built into Android. It is widely used for local data storage in mobile applications because of its simplicity, reliability, and efficiency.

CRUD Operations

CRUD stands for Create, Read, Update, and Delete, representing the four basic operations for managing data in a database. These operations are fundamental for any database application, and SQLite provides easy methods to implement them.

1. Create: This operation involves adding new records to the database. In SQLite, this is typically done using the INSERT statement. In an Android application, the SQLiteDatabase.insert() method is used for this purpose.
2. Read: Reading or retrieving data is accomplished using the SELECT statement. In Android, this can be done with the rawQuery() method of SQLiteDatabase, allowing you to execute SQL queries and retrieve results in the form of a Cursor object.
3. Update: This operation modifies existing records in the database. In SQLite, this is achieved through the UPDATE statement, and in Android, you can use the SQLiteDatabase.update() method to perform updates.
4. Delete: The delete operation removes records from the database using the DELETE statement. In Android, you can use the SQLiteDatabase.delete() method to remove specific entries based on given criteria.

Implementation in Android

To work with SQLite in an Android application:

1. Database Helper: Create a subclass of SQLiteOpenHelper to manage database creation and version management. This class typically contains methods for CRUD operations.
2. Database Schema: Define the database schema (tables and fields) in the onCreate() method of the helper class. This includes SQL statements to create tables.

3. CRUD Methods: Implement methods for each CRUD operation, providing simple interfaces for adding, reading, updating, and deleting records.

Advantages of Using SQLite

1. Lightweight: SQLite has a small footprint and is suitable for mobile devices.
2. Self-contained: It does not require a separate server process, making it easy to integrate into applications.
3. ACID Compliance: SQLite transactions are atomic, consistent, isolated, and durable, ensuring data integrity.
4. Cross-platform: It can be used across different platforms and is widely supported.

Code:

DatabaseHelper.java

```
package com.example.mc5;
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 = "students.db";
    private static final String TABLE_NAME = "students";
    private static final String COL_1 = "rollno";
    private static final String COL_2 = "name";
    private static final String COL_3 = "emailId";
    private static final String COL_4 = "course";

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

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE_NAME + " (rollno INTEGER PRIMARY KEY,
name TEXT, emailId TEXT, course TEXT)");
    }
```

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

    public boolean insertData(int rollno, String name, String emailId, String course) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1, rollno);
        contentValues.put(COL_2, name);
        contentValues.put(COL_3, emailId);
        contentValues.put(COL_4, course);
        long result = db.insert(TABLE_NAME, null, contentValues);
        return result != -1; // returns true if data inserted
    }

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

    public boolean updateData(int rollno, String name, String emailId, String course) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1, rollno);
        contentValues.put(COL_2, name);
        contentValues.put(COL_3, emailId);
        contentValues.put(COL_4, course);
        db.update(TABLE_NAME, contentValues, "rollno = ?", new
String[]{String.valueOf(rollno)});
        return true;
    }

    public Integer deleteData(int rollno) {
```

```
        SQLiteDatabase db = this.getWritableDatabase();
        return db.delete(TABLE_NAME, "rollno = ?", new String[]{String.valueOf(rollno)});
    }
}
```

ViewDataActivity.java

```
package com.example.mc5;
import android.database.Cursor;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class ViewDataActivity extends AppCompatActivity {
    DatabaseHelper myDb;
    TextView textViewData;

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

        myDb = new DatabaseHelper(this);
        textViewData = findViewById(R.id.textViewData);

        displayData();
    }

    private void displayData() {
        Cursor res = myDb.getAllData();
        if (res.getCount() == 0) {
            textViewData.setText("No Data Found");
            return;
        }

        StringBuilder stringBuffer = new StringBuilder();
        while (res.moveToNext()) {
```

```
            stringBuffer.append("Roll No: ").append(res.getString(0)).append("\n");
            stringBuffer.append("Name: ").append(res.getString(1)).append("\n");
            stringBuffer.append("Email ID: ").append(res.getString(2)).append("\n");
            stringBuffer.append("Course: ").append(res.getString(3)).append("\n\n");
        }

        textViewData.setText(stringBuffer.toString());
        res.close(); // Always close the cursor
    }
}
```

MainActivity.java

```
package com.example.mc5;
import android.content.Intent;
import android.database.Cursor;
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 MainActivity extends AppCompatActivity {
    DatabaseHelper myDb;
    EditText editRollNo, editName, editEmailId, editCourse;
    Button btnAddData, btnViewData, btnUpdateData, btnDeleteData;

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

        myDb = new DatabaseHelper(this);
        editRollNo = findViewById(R.id.editTextRollNo);
        editName = findViewById(R.id.editTextName);
        editEmailId = findViewById(R.id.editTextEmailId);
```

```
        editCourse = findViewById(R.id.editTextCourse);
        btnAddData = findViewById(R.id.buttonAdd);
        btnViewData = findViewById(R.id.buttonView);
        btnUpdateData = findViewById(R.id.buttonUpdate);
        btnDeleteData = findViewById(R.id.buttonDelete);

        addData();
        viewData();
        updateData();
        deleteData();
    }

    public void addData() {
        btnAddData.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String rollno = editRollNo.getText().toString();
                String name = editName.getText().toString();
                String emailId = editEmailId.getText().toString();
                String course = editCourse.getText().toString();

                if (rollno.isEmpty() || name.isEmpty() || emailId.isEmpty() || course.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please fill all fields",
Toast.LENGTH_SHORT).show();
                    return;
                }

                try {
                    int rollNumberInt = Integer.parseInt(rollno);
                    boolean isInserted = myDb.insertData(rollNumberInt, name, emailId, course);
                    if (isInserted) {
                        Toast.makeText(MainActivity.this, "Data Inserted",
Toast.LENGTH_SHORT).show();
                        clearInputs();
                    } else {
```

```
                    Toast.makeText(MainActivity.this, "Data Not Inserted",
Toast.LENGTH_SHORT).show();
                }
            } catch (NumberFormatException e) {
                Toast.makeText(MainActivity.this, "Invalid Roll Number",
Toast.LENGTH_SHORT).show();
            }
        }
    });
}

public void viewData() {
    btnViewData.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, ViewDataActivity.class);
            startActivity(intent);
        }
    });
}

public void updateData() {
    btnUpdateData.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String rollno = editRollNo.getText().toString();
            String name = editName.getText().toString();
            String emailId = editEmailId.getText().toString();
            String course = editCourse.getText().toString();

            if (rollno.isEmpty() || name.isEmpty() || emailId.isEmpty() || course.isEmpty()) {
                Toast.makeText(MainActivity.this, "Please fill all fields",
Toast.LENGTH_SHORT).show();
                return;
            }
```

```
            try {
                int rollNumberInt = Integer.parseInt(rollno);
                boolean isUpdated = myDb.updateData(rollNumberInt, name, emailId, course);
                if (isUpdated) {
                    Toast.makeText(MainActivity.this, "Data Updated",
Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, "Data Not Updated",
Toast.LENGTH_SHORT).show();
                }
            } catch (NumberFormatException e) {
                Toast.makeText(MainActivity.this, "Invalid Roll Number",
Toast.LENGTH_SHORT).show();
            }
        }
    });
  }

  public void deleteData() {
    btnDeleteData.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        String rollno = editRollNo.getText().toString();

        if (rollno.isEmpty()) {
            Toast.makeText(MainActivity.this, "Please enter a Roll Number",
Toast.LENGTH_SHORT).show();
            return;
        }

        try {
            Integer deletedRows = myDb.deleteData(Integer.parseInt(rollno));
            if (deletedRows > 0) {
                Toast.makeText(MainActivity.this, "Data Deleted",
Toast.LENGTH_SHORT).show();
            } else {
```

```
                    Toast.makeText(MainActivity.this, "Data Not Deleted",
Toast.LENGTH_SHORT).show();
                }
            } catch (NumberFormatException e) {
                Toast.makeText(MainActivity.this, "Invalid Roll Number",
Toast.LENGTH_SHORT).show();
            }
        }
    });
  }

  private void clearInputs() {
    editRollNo.setText("");
    editName.setText("");
    editEmailId.setText("");
    editCourse.setText("");
  }
}
```

viewactivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp">

  <TextView
    android:id="@+id/textViewData"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="16sp" />
</LinearLayout>
```

activity_main.xml

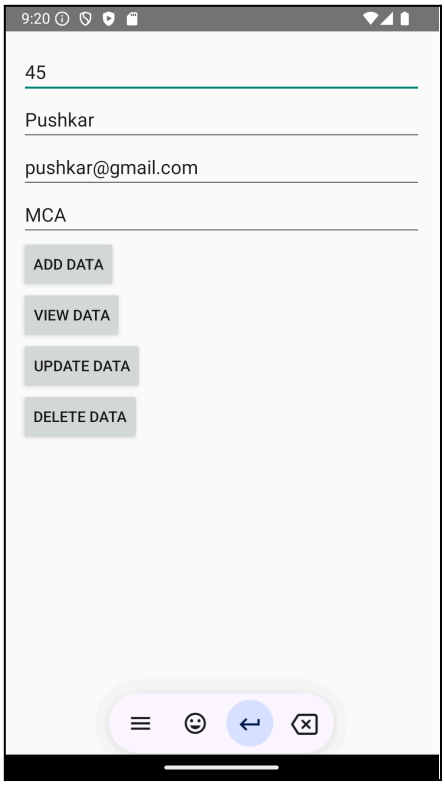
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <EditText
        android:id="@+id/editTextRollNo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Roll No" />

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

    <EditText
        android:id="@+id/editTextEmailId"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email ID" />
    <EditText
        android:id="@+id/editTextCourse"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Course" />

    <Button
        android:id="@+id/buttonAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add Data" />
```

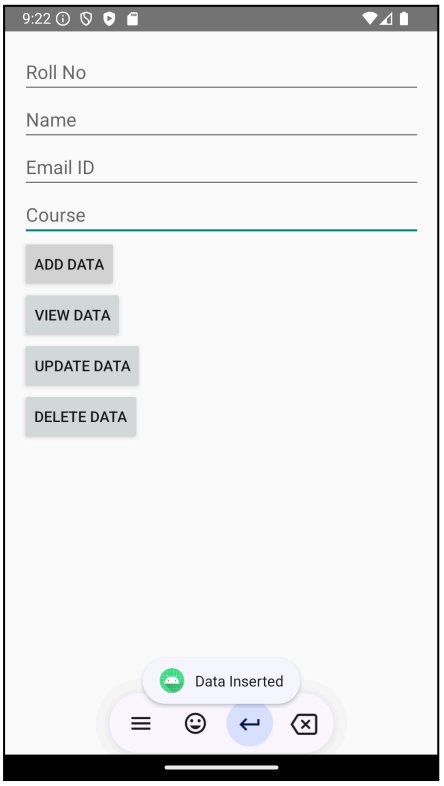
```
    <Button
        android:id="@+id/buttonView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="View Data" />

    <Button
        android:id="@+id/buttonUpdate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Update Data" />

    <Button
        android:id="@+id/buttonDelete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Delete Data" />
</LinearLayout>
```

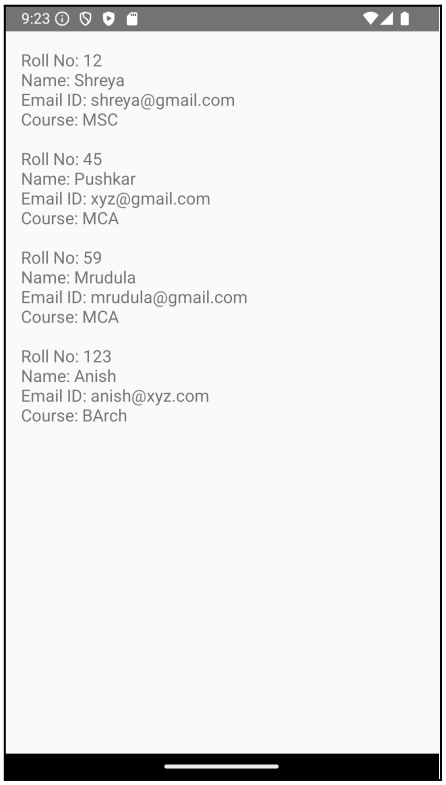
AndroidManifest.xml (Add)

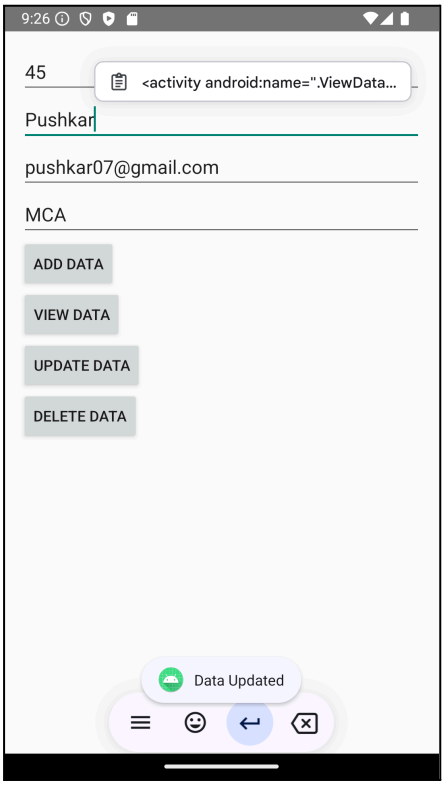
```
 <activity android:name=".ViewDataActivity" />
```

Output:


9:20
45
Pushkar
pushkar@gmail.com
MCA
ADD DATA
VIEW DATA
UPDATE DATA
DELETE DATA


9:22
Roll No
Name
Email ID
Course
ADD DATA
VIEW DATA
UPDATE DATA
DELETE DATA
Data Inserted


9:23
Roll No: 12
Name: Shreya
Email ID: shreya@gmail.com
Course: MSC
Roll No: 45
Name: Pushkar
Email ID: xyz@gmail.com
Course: MCA
Roll No: 59
Name: Mrudula
Email ID: mrudula@gmail.com
Course: MCA
Roll No: 123
Name: Anish
Email ID: anish@xyz.com
Course: BArch


9:26
45
<activity android:name=".ViewData...
Pushkar
pushkar07@gmail.com
MCA
ADD DATA
VIEW DATA
UPDATE DATA
DELETE DATA
Data Updated

Conclusion:

SQLite serves as a powerful tool for data management in Android applications, allowing developers to create robust apps that can efficiently store and retrieve data. Understanding how to implement CRUD operations is essential for building any data-driven application.