Example – SQLite

As per the pattern of the practical exam

ExampleSQLiteWithMultipleColumnListview

1. 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">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="Example - SQLite"
        android:textSize="24sp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/addbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginLeft="50dp"
        android:layout_marginTop="59dp"
        android:onClick="adduser"
        android:text="Add"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Button
        android:id="@+id/updatebutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="59dp"
        android:layout_marginEnd="50dp"
        android:layout_marginRight="50dp"
        android:onClick="update"
        android:text="Update"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Button
        android:id="@+id/deletebutton"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginLeft="50dp"
        android:layout_marginTop="31dp"
        android:onClick="delete"
        android:text="Delete"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/addbutton" />

    <Button
        android:id="@+id/viewbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="31dp"
        android:layout_marginEnd="50dp"
        android:layout_marginRight="50dp"
        android:onClick="viewdata"
        android:text="View"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/updatebutton" />

    <EditText
        android:id="@+id/nameedit"
        android:layout_width="185dp"
        android:layout_height="48dp"
        android:layout_marginStart="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="65dp"
        android:ems="10"
        android:hint="Name"
        android:inputType="textPersonName"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/deletebutton" />

    <EditText
        android:id="@+id/classedit"
        android:layout_width="185dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="30dp"
        android:ems="10"
        android:hint="Class"
        android:inputType="textPersonName"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/nameedit" />

    <Spinner
        android:id="@+id/subjectedit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="30dp"
        android:layout_marginEnd="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginBottom="40dp"
        app:layout_constraintBottom_toTopOf="@+id/emailedit"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

```
            app:layout_constraintTop_toBottomOf="@+id/classedit" />

        <EditText
            android:id="@+id/emailedit"
            android:layout_width="185dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="20dp"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="30dp"
            android:ems="10"
            android:hint="Email"
            android:inputType="textEmailAddress"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/subjectedit" />

        <EditText
            android:id="@+id/newnameedit"
            android:layout_width="185dp"
            android:layout_height="46dp"
            android:layout_marginStart="15dp"
            android:layout_marginLeft="15dp"
            android:layout_marginTop="65dp"
            android:ems="10"
            android:hint="New name"
            android:inputType="textPersonName"
            app:layout_constraintStart_toEndOf="@+id/nameedit"
            app:layout_constraintTop_toBottomOf="@+id/viewbutton" />

        <ListView
            android:id="@+id/lv"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginStart="20dp"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="300dp"
            android:layout_marginEnd="20dp"
            android:layout_marginRight="20dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

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

2. myDbAdapter.java

```
package com.example.examplesqlitewithmultiplecolumnlistview;

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

import java.util.ArrayList;

public class myDbAdapter {
    mydBHelper myHelper;
```

```
    public myDbAdapter(Context context){
        myHelper = new mydBHelper(context);
    }

    public long insertdata(String name, String classyear, String
subject, String email){
        SQLiteDatabase dbb = myHelper.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(mydBHelper.NAME,name);
        contentValues.put(mydBHelper.CLASSYEAR,classyear);
        contentValues.put(mydBHelper.SUBJECT,subject);
        contentValues.put(mydBHelper.EMAIL,email);
        long id =
dbb.insert(mydBHelper.TABLE_NAME,null,contentValues);
        return id;
    }

    public ArrayList<String> getData() {

        SQLiteDatabase db = myHelper.getWritableDatabase();
        String[] columns = {mydBHelper.UID, mydBHelper.NAME,
mydBHelper.CLASSYEAR, mydBHelper.SUBJECT, mydBHelper.EMAIL};
        Cursor cursor = db.query(mydBHelper.TABLE_NAME, columns,
null, null, null, null, null);
        ArrayList<String> theList = new ArrayList<>();
        while (cursor.moveToNext()) {
            StringBuffer strbuf=new StringBuffer();
            int cid
=cursor.getInt(cursor.getColumnIndex(mydBHelper.UID));
            String nm
=cursor.getString(cursor.getColumnIndex(mydBHelper.NAME));
            String  classyear
=cursor.getString(cursor.getColumnIndex(mydBHelper.CLASSYEAR));
            String  subject
=cursor.getString(cursor.getColumnIndex(mydBHelper.SUBJECT));
            String  email
=cursor.getString(cursor.getColumnIndex(mydBHelper.EMAIL));
            strbuf.append(cid+ "   " + nm + "   " + classyear +"  "
+ subject +" " + email +"\n");
            theList.add(strbuf.toString());
        }

        return theList;
    }


    public  int delete(String uname)
    {
        SQLiteDatabase db = myHelper.getWritableDatabase();
        String[] whereArgs ={uname};

        int count =db.delete(mydBHelper.TABLE_NAME
,mydBHelper.NAME+" = ?",whereArgs);
        return  count;
    }

    public int updateName(String oldName , String newName)
    {
```

```
        SQLiteDatabase db = myHelper.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(mydBHelper.NAME,newName);
        String[] whereArgs= {oldName};
        int count =db.update(mydBHelper.TABLE_NAME,contentValues,
mydBHelper.NAME+" = ?",whereArgs );
        return count;
    }

    String find_details(String name)
    {
        SQLiteDatabase db = myHelper.getWritableDatabase();
        String[] columns =
{mydBHelper.UID,mydBHelper.NAME,mydBHelper.CLASSYEAR,mydBHelper.SUBJ
ECT,mydBHelper.EMAIL};
        String[] whereArgs={name};
        Cursor cursor
=db.query(mydBHelper.TABLE_NAME,columns,mydBHelper.NAME+"=?",whereAr
gs,null,null,null);
        StringBuffer buffer= new StringBuffer();
        while (cursor.moveToNext())
        {
            int cid
=cursor.getInt(cursor.getColumnIndex(mydBHelper.UID));
            String nm
=cursor.getString(cursor.getColumnIndex(mydBHelper.NAME));
            String  classyear
=cursor.getString(cursor.getColumnIndex(mydBHelper.CLASSYEAR));
            String  subject
=cursor.getString(cursor.getColumnIndex(mydBHelper.SUBJECT));
            String  email
=cursor.getString(cursor.getColumnIndex(mydBHelper.EMAIL));
            buffer.append(cid+ "   " + nm + "   " + classyear +"  "
+ subject +" " + email +"\n");
        }
        return buffer.toString();
    }


    public class mydBHelper extends SQLiteOpenHelper {
        private static final String DATABASE_NAME = "myDatabase";
        private static final String TABLE_NAME = "myTable";
        private static final int DATABASE_Version = 1;
        private static final String UID = "id";
        private static final String NAME = "Name";
        private static final String CLASSYEAR = "Class";
        private static final String SUBJECT = "Subject";
        private static final String EMAIL = "Email";
        private static final String CREATE_TABLE = "CREATE TABLE " +
TABLE_NAME + " (" + UID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
NAME + " VARCHAR(255) ," + CLASSYEAR + " VARCHAR(255) ," + SUBJECT +
" VARCHAR(255) ," + EMAIL + " VARCHAR(255));";
        private static final String DROP_TABLE ="DROP TABLE IF
EXISTS "+TABLE_NAME;
        private Context context;

        public mydBHelper(Context context) {
            super(context, DATABASE_NAME, null, DATABASE_Version);
            this.context = context;
        }
```

```
        @Override
        public void onCreate(SQLiteDatabase db) {
            try {
                db.execSQL(CREATE_TABLE);
            } catch (Exception e) {
                String message = "Exception error";

Toast.makeText(context,message,Toast.LENGTH_LONG).show();

            }
        }

        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
            try {

Toast.makeText(context,"OnUpgrade",Toast.LENGTH_LONG).show();
                db.execSQL(DROP_TABLE);
                onCreate(db);
            }catch (Exception e) {
                Toast.makeText(context,"Exception
Error",Toast.LENGTH_LONG).show();
            }
        }

    }
}
```

3. MainActivity.java

```
package com.example.examplesqlitewithmultiplecolumnlistview;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.Toast;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {
    String[] subjects = {"Computer Science", "Mathematics",
"Electronics", "Physics", "Chemistry", "Botany", "Microbiology"};
    Button addbutton, viewbutton, deletebutton, updatebutton;
    EditText nameedit, classedit, emailedit, newnamedit;
    Spinner subjectedit;
    ListView lv;
```

```
    ArrayAdapter ad;
    myDbAdapter helper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addbutton = findViewById(R.id.addbutton);
        viewbutton = findViewById(R.id.viewbutton);
        updatebutton = findViewById(R.id.updatebutton);
        deletebutton = findViewById(R.id.deletebutton);
        nameedit = findViewById(R.id.nameedit);
        classedit = findViewById(R.id.classedit);
        subjectedit = findViewById(R.id.subjectedit);
        subjectedit.setOnItemSelectedListener(this);
        emailedit = findViewById(R.id.emailedit);
        newnamedit = findViewById(R.id.newnameedit);
        lv = findViewById(R.id.lv);

        ad = new ArrayAdapter(this,
android.R.layout.simple_spinner_item,subjects);

ad.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_
item);
        subjectedit.setAdapter(ad);

        helper = new myDbAdapter(this);
        nameedit.setVisibility(View.INVISIBLE);
        classedit.setVisibility(View.INVISIBLE);
        subjectedit.setVisibility(View.INVISIBLE);
        emailedit.setVisibility(View.INVISIBLE);
        newnamedit.setVisibility(View.INVISIBLE);

    }
    @Override
    public void onItemSelected(AdapterView arg0, View arg1, int
position, long id) {

    }
    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

    public void viewdata(View view) {
        nameedit.setVisibility(View.INVISIBLE);
        classedit.setVisibility(View.INVISIBLE);
        subjectedit.setVisibility(View.GONE);
        emailedit.setVisibility(View.INVISIBLE);
        newnamedit.setVisibility(View.INVISIBLE);
        ArrayList theList = helper.getData();
        ListAdapter listadpater = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, theList);
        lv.setVisibility(View.VISIBLE);
        lv.setAdapter(listadpater);
     }


    public void adduser(View view) {
```

```
        nameedit.setVisibility(View.VISIBLE);
        classedit.setVisibility(View.VISIBLE);
        subjectedit.setVisibility(View.VISIBLE);
        emailedit.setVisibility(View.VISIBLE);
        newnamedit.setVisibility(View.INVISIBLE);
        lv.setVisibility(View.INVISIBLE);
        String n = nameedit.getText().toString();
        String c = classedit.getText().toString();
        String s = subjectedit.getSelectedItem().toString();
        String e = emailedit.getText().toString();
        if (n.isEmpty() || c.isEmpty() || s.isEmpty() ||
e.isEmpty()) {
            Toast.makeText(getApplicationContext(), "Enter name,
class, subject and email", Toast.LENGTH_LONG).show();
        } else {
            long id = helper.insertdata(n, c, s, e);
            if (id <= 0) {
                Toast.makeText(getApplicationContext(), "Insertion
Unsuccessful", Toast.LENGTH_LONG).show();
                nameedit.setText("");
                classedit.setText("");
                subjectedit.setAdapter(ad);
                emailedit.setText("");
            } else {
                Toast.makeText(getApplicationContext(), "Insertion
Successful", Toast.LENGTH_LONG).show();
                nameedit.setVisibility(View.INVISIBLE);
                classedit.setVisibility(View.INVISIBLE);
                subjectedit.setVisibility(View.INVISIBLE);
                emailedit.setVisibility(View.INVISIBLE);
                nameedit.setText("");
                classedit.setText("");
                subjectedit.setAdapter(ad);
                emailedit.setText("");
            }
        }
    }

    public void update(View view) {
        nameedit.setVisibility(View.VISIBLE);
        classedit.setVisibility(View.INVISIBLE);
        subjectedit.setVisibility(View.INVISIBLE);
        emailedit.setVisibility(View.INVISIBLE);
        newnamedit.setVisibility(View.VISIBLE);
        lv.setVisibility(View.INVISIBLE);
        addbutton.setEnabled(false);
        deletebutton.setEnabled(false);
        viewbutton.setEnabled(false);

        String u1 = nameedit.getText().toString();
        String u2 = newnamedit.getText().toString();
        if (u1.isEmpty() || u2.isEmpty()) {
            Toast.makeText(getApplicationContext(), "Enter data",
Toast.LENGTH_LONG).show();
        } else {
            int a = helper.updateName(u1, u2);
            if (a <= 0) {
                Toast.makeText(getApplicationContext(), "Update
Unsuccessful", Toast.LENGTH_LONG).show();
                nameedit.setText("");
```

```
                newnamedit.setText("");
            } else {
                Toast.makeText(getApplicationContext(), "Update
Successful", Toast.LENGTH_LONG).show();
                nameedit.setText("");
                newnamedit.setText("");
            }
        }
        addbutton.setEnabled(true);
        deletebutton.setEnabled(true);
        viewbutton.setEnabled(true);
        updatebutton.setEnabled(true);
    }


    public void delete(View view) {
        nameedit.setVisibility(View.VISIBLE);
        classedit.setVisibility(View.INVISIBLE);
        subjectedit.setVisibility(View.INVISIBLE);
        emailedit.setVisibility(View.INVISIBLE);
        newnamedit.setVisibility(View.INVISIBLE);
        lv.setVisibility(View.INVISIBLE);
        addbutton.setEnabled(false);
        updatebutton.setEnabled(false);
        viewbutton.setEnabled(false);
        String nametodelete = nameedit.getText().toString();
        if (nametodelete.isEmpty()) {
            Toast.makeText(getApplicationContext(), "Enter name to
delete", Toast.LENGTH_LONG).show();
        } else {
            int a = helper.delete(nametodelete);
            if (a <= 0) {
                Toast.makeText(getApplicationContext(),
"Unsuccessful Deletion", Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(getApplicationContext(), "Successful
Deletion", Toast.LENGTH_LONG).show();
            }
            nameedit.setText("");
            nameedit.setVisibility(View.INVISIBLE);
            addbutton.setEnabled(true);
            updatebutton.setEnabled(true);
            viewbutton.setEnabled(true);
        }
    }
}
```