Practical No 19

Content Provider

MainActivity.java

```
package com.example.MyApplication;
import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;
import android.content.ContentValues;
import android.content.CursorLoader;

public class MainActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }
  public void onClickAddName(View view) {
    ContentValues values = new ContentValues();
    values.put(StudentsProvider.NAME,
      ((EditText)findViewById(R.id.editText2)).getText().toString());
    values.put(StudentsProvider.GRADE,
      ((EditText)findViewById(R.id.editText3)).getText().toString());
    Uri uri = getContentResolver().insert(
      StudentsProvider.CONTENT_URI, values);
    Toast.makeText(getBaseContext(),
      uri.toString(), Toast.LENGTH_LONG).show();
  }
  public void onClickRetrieveStudents(View view) {
    String URL = "content://com.example.MyApplication.StudentsProvider";
    Uri students = Uri.parse(URL);
    Cursor c = managedQuery(students, null, null, null, "name");
    if (c.moveToFirst()) {
      do{
        Toast.makeText(this,
          c.getString(c.getColumnIndex(StudentsProvider._ID)) +
            ", " +  c.getString(c.getColumnIndex( StudentsProvider.NAME)) +
              ", " + c.getString(c.getColumnIndex( StudentsProvider.GRADE)),
        Toast.LENGTH_SHORT).show();
      } while (c.moveToNext());
    }
  }
}
```

StudentsProvider.java

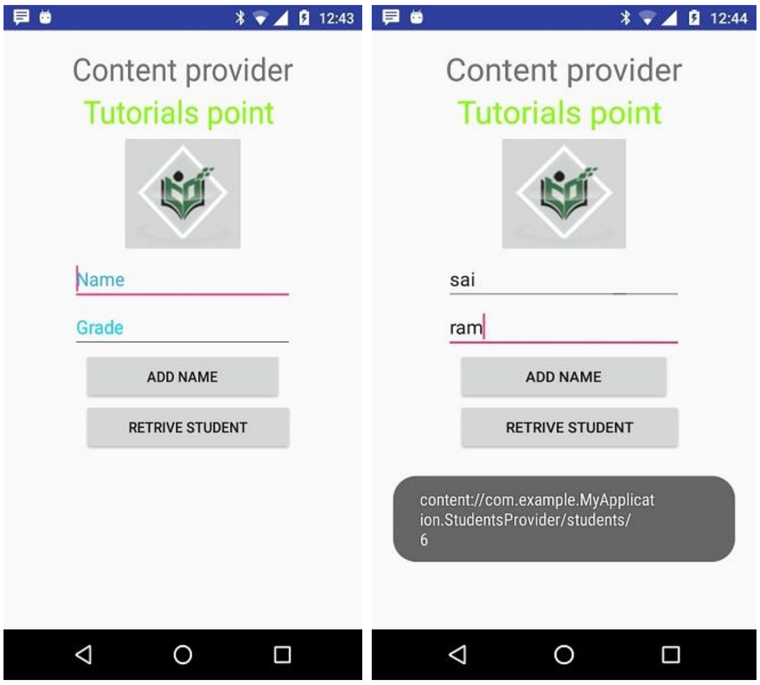
```
package com.example.MyApplication;
import java.util.HashMap;
import android.content.UriMatcher;
import android.database.SQLException;

public class StudentsProvider extends ContentProvider {
  static final String PROVIDER_NAME = "com.example.MyApplication.StudentsProvider";
  static final String URL = "content://" + PROVIDER_NAME + "/students";
  static final Uri CONTENT_URI = Uri.parse(URL);
static final String _ID = "_id";
static final String NAME = "name";
static final String GRADE = "grade";
private static HashMap<String, String> STUDENTS_PROJECTION_MAP;
static final int STUDENTS = 1;
static final int STUDENT_ID = 2;
static final UriMatcher uriMatcher;
static{
  uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
  uriMatcher.addURI(PROVIDER_NAME, "students", STUDENTS);
  uriMatcher.addURI(PROVIDER_NAME, "students/#", STUDENT_ID);
}
private SQLiteDatabase db;
static final String DATABASE_NAME = "College";
static final String STUDENTS_TABLE_NAME = "students";
static final int DATABASE_VERSION = 1;
static final String CREATE_DB_TABLE =
  " CREATE TABLE " + STUDENTS_TABLE_NAME +
    " (_id INTEGER PRIMARY KEY AUTOINCREMENT, " +
    " name TEXT NOT NULL, " +
    " grade TEXT NOT NULL);";
private static class DatabaseHelper extends SQLiteOpenHelper {
  DatabaseHelper(Context context){
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
  }
  public void onCreate(SQLiteDatabase db) {
    db.execSQL(CREATE_DB_TABLE);
  }
  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " +  STUDENTS_TABLE_NAME);
    onCreate(db);
  }
}
public boolean onCreate() {
  Context context = getContext();
  DatabaseHelper dbHelper = new DatabaseHelper(context);
  db = dbHelper.getWritableDatabase();
  return (db == null)? false:true;
}

@Override
public Uri insert(Uri uri, ContentValues values) {
  long rowID = db.insert( STUDENTS_TABLE_NAME, "", values);
  if (rowID > 0) {
    Uri _uri = ContentUris.withAppendedId(CONTENT_URI, rowID);
    getContext().getContentResolver().notifyChange(_uri, null);
    return _uri;
  }
  throw new SQLException("Failed to add a record into " + uri);
}

@Override
public Cursor query(Uri uri, String[] projection,
  String selection,String[] selectionArgs, String sortOrder) {
  SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
  qb.setTables(STUDENTS_TABLE_NAME);
  switch (uriMatcher.match(uri)) {
    case STUDENTS:
      qb.setProjectionMap(STUDENTS_PROJECTION_MAP);
    break;
    case STUDENT_ID:
      qb.appendWhere( _ID + "=" + uri.getPathSegments().get(1));
    break;
  }
  c.setNotificationUri(getContext().getContentResolver(), uri);
  return c;
}

@Override
public int delete(Uri uri, String selection, String[] selectionArgs) {
  int count = 0;
  switch (uriMatcher.match(uri)){
    case STUDENTS:
      count = db.delete(STUDENTS_TABLE_NAME, selection, selectionArgs);
    break;

    case STUDENT_ID:
      String id = uri.getPathSegments().get(1);
      count = db.delete( STUDENTS_TABLE_NAME, _ID +  " = " + id +
        (!TextUtils.isEmpty(selection) ? "
        AND (" + selection + ')' : ""), selectionArgs);
      break;
    default:
      throw new IllegalArgumentException("Unknown URI " + uri);
  }

  getContext().getContentResolver().notifyChange(uri, null);
  return count;
}

@Override
public int update(Uri uri, ContentValues values,
  String selection, String[] selectionArgs) {
  int count = 0;
  switch (uriMatcher.match(uri)) {
    case STUDENTS:
      count = db.update(STUDENTS_TABLE_NAME, values, selection, selectionArgs);
    break;

    case STUDENT_ID:
      count = db.update(STUDENTS_TABLE_NAME, values,
        _ID + " = " + uri.getPathSegments().get(1) +
        (!TextUtils.isEmpty(selection) ? "
        AND (" +selection + ')' : ""), selectionArgs);
      break;
    default:
      throw new IllegalArgumentException("Unknown URI " + uri );
  }
```

```
    getContext().getContentResolver().notifyChange(uri, null);
    return count;
  }

  @Override
  public String getType(Uri uri) {
    switch (uriMatcher.match(uri)){
      case STUDENTS:
        return "vnd.android.cursor.dir/vnd.example.students";
      case STUDENT_ID:
        return "vnd.android.cursor.item/vnd.example.students";
      default:
        throw new IllegalArgumentException("Unsupported URI: " + uri);
    }
  }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  tools:context="com.example.MyApplication.MainActivity">

  <TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Content provider"/>

  <TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tutorials point "/>

  <ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageButton"
    android:src="@drawable/abc"/>

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button2"
    android:text="Add Name"
    android:onClick="onClickAddName"/>
  <EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText" />

  <EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText2"
    android:hint="Name"/>

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Retrive student"
    android:id="@+id/button"/>
</RelativeLayout>
```

Output



```
content://com.example.MyApplicat
ion.StudentsProvider/students/
6
```