**/\* 1. write a program to create your own content provider to insert and access data in android application**

**\*/**

*<?***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"**

**android:orientation="vertical"**

**tools:context=".MainActivity"**>

<**TextView**

**android:id="@+id/textView1"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Content provider"**

**android:layout\_alignParentTop="true"**

**android:layout\_centerHorizontal="true"**

**android:textSize="30dp"** />

<**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"**

**android:textColorHint="@android:color/holo\_blue\_light"** />

<**EditText**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:id="@+id/editText3"**

**android:hint="Grade"**

**android:textColorHint="@android:color/holo\_blue\_bright"** />

<**Button**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:id="@+id/button2"**

**android:text="Add Name"**

**android:onClick="onClickAddName"**/>

<**Button**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content" android:text="Retrive student"**

**android:id="@+id/button"**

**android:onClick="onClickRetrieveStudents"**/>

</**LinearLayout**>

Manifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*

<**manifest xmlns:android="http://schemas.android.com/apk/res/android"**

**package="com.example.myapplicationcontentprovider"**>

<**application**

**android:allowBackup="true"**

**android:icon="@mipmap/ic\_launcher"**

**android:label="@string/app\_name"**

**android:roundIcon="@mipmap/ic\_launcher\_round"**

**android:supportsRtl="true"**

**android:theme="@style/AppTheme"**>

<**activity android:name=".MainActivity"**>

<**intent-filter**>

<**action android:name="android.intent.action.MAIN"** />

<**category android:name="android.intent.category.LAUNCHER"** />

</**intent-filter**>

</**activity**>

<**provider android:name="StudentsProvider"**

**android:authorities="com.example.MyApplication.StudentsProvider"**/>

</**application**>

</**manifest**>

StudentsProvider.java

**package** com.example.myapplicationcontentprovider;

**import** android.content.ContentProvider;

**import** android.content.ContentUris;

**import** android.content.ContentValues;

**import** android.content.Context;

**import** android.content.UriMatcher;

**import** android.database.Cursor;

**import** android.database.SQLException;**import** android.database.sqlite.SQLiteDatabase;

**import** android.database.sqlite.SQLiteOpenHelper;

**import** android.database.sqlite.SQLiteQueryBuilder;

**import** android.net.Uri;

**import** android.text.TextUtils;

**import** java.util.HashMap;

**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***);

}

*/\*\**

*\* Database specific constant declarations*

*\*/*

**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);"**;

*/\*\**

*\* Helper class that actually creates and manages*

*\* the provider's underlying data repository.*

*\*/*

**private static class** DatabaseHelper **extends** SQLiteOpenHelper {

DatabaseHelper(Context context){ **super**(context, ***DATABASE\_NAME***, **null**, ***DATABASE\_VERSION***);

}

@Override

**public void** onCreate(SQLiteDatabase db) {

db.execSQL(***CREATE\_DB\_TABLE***);

}

@Override

**public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {

db.execSQL(**"DROP TABLE IF EXISTS "** + ***STUDENTS\_TABLE\_NAME***);

onCreate(db);

}

}

@Override

**public boolean** onCreate() {

Context context = getContext();

DatabaseHelper dbHelper = **new** DatabaseHelper(context);

*/\*\**

*\* Create a write able database which will trigger its*

*\* creation if it doesn't already exist.*

*\*/*

**db** = dbHelper.getWritableDatabase();

**return** (**db** == **null**)? **false**:**true**;

}

@Override

**public** Uri insert(Uri uri, ContentValues values) {

*/\*\**

*\* Add a new student record*

*\*/*

**long** rowID = **db**.insert( ***STUDENTS\_TABLE\_NAME***, **""**, values);

*/\*\**

*\* If record is added successfully*

*\*/*

**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**;

**default**:

}

**if** (sortOrder == **null** || sortOrder == **""**){

*/\*\**

*\* By default sort on student names*

*\*/*

sortOrder = ***NAME***;

}

Cursor c = qb.query(**db**, projection, selection,

selectionArgs,**null**, **null**, sortOrder);

*/\*\**

*\* register to watch a content URI for changes*

*\*/*

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)){

*/\*\**

*\* Get all student records*

*\*/*

**case *STUDENTS***:

**return "vnd.android.cursor.dir/vnd.example.students"**;

*/\*\**

*\* Get a particular student*

*\*/*

**case *STUDENT\_ID***:

**return "vnd.android.cursor.item/vnd.example.students"**;

**default**:

**throw new** IllegalArgumentException(**"Unsupported URI: "** + uri);

}

}

}

MainActivity.java

**package** com.example.myapplicationcontentprovider;

**import** androidx.appcompat.app.AppCompatActivity;**import** android.content.ContentValues;

**import** android.database.Cursor;

**import** android.net.Uri;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.EditText;

**import** android.widget.Toast;

**public class** MainActivity **extends** AppCompatActivity {

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

}

**public void** onClickAddName(View view) {

*// Add a new student record*

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) {

*// Retrieve student records*

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());

}

}

}