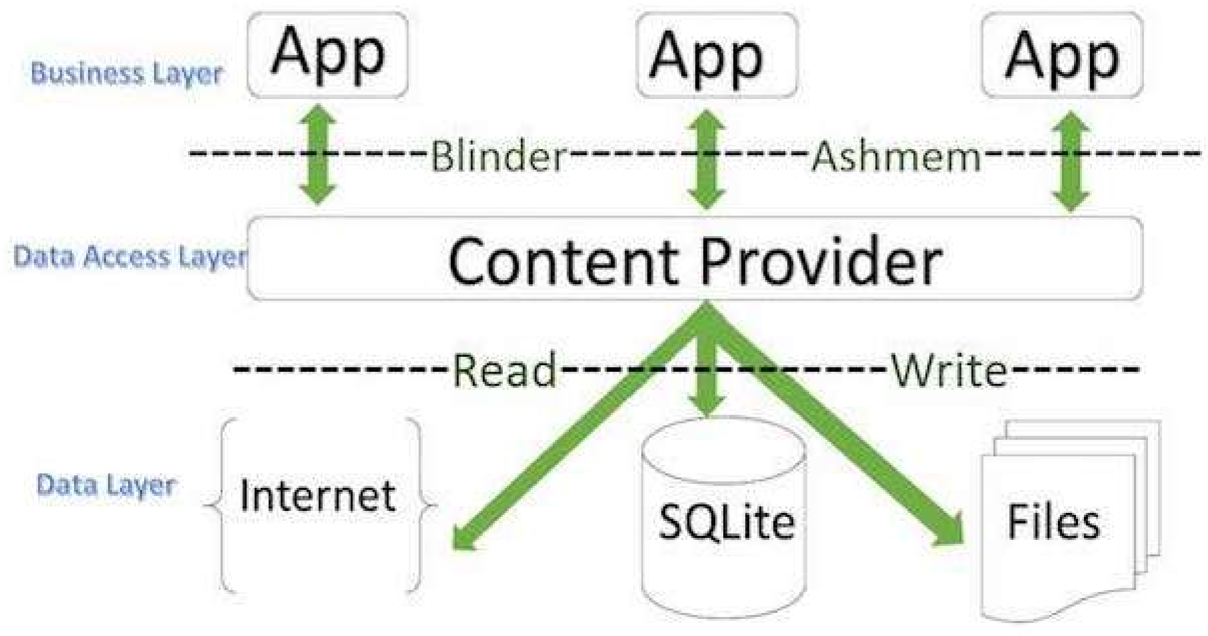
Content Provider

- A **content provider** component *supplies data from one application to others on request*. Such requests are handled by the methods of the **ContentResolver** class. A content provider can use different ways to *store its data and the data can be stored in a database, in files, or even over a network.*

Content Provider



ContentProvider

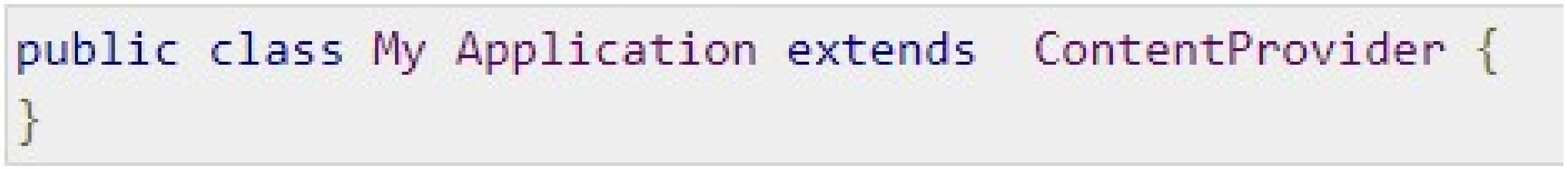
- ❑ ***content providers become very useful in sharing data across applications***

more about Content Provider…

- **Content providers** let you *centralize content in one place* and have many different applications access it as needed.
- A **content provider** *behaves very much like a database* where you can query it, edit its content, as well as add or delete content using insert(), update(), delete(), and query() methods. In most cases this data is stored in an **SQlite** database.

Content Provider

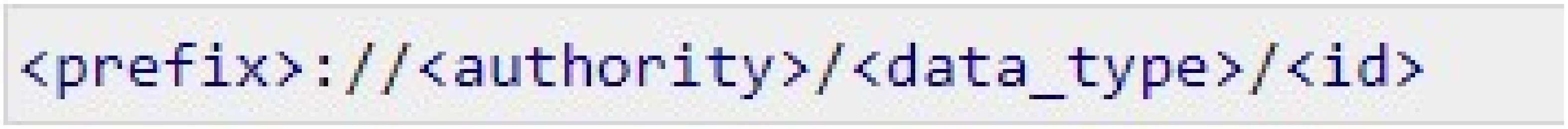
- A content provider is implemented as a subclass of **ContentProvider** class and must implement a standard set of APIs that enable other applications to perform transactions.



```
public class My Application extends  ContentProvider {
}
```

Content URIs

- To query a content provider, you specify the query string in the form of a URI which has following format –

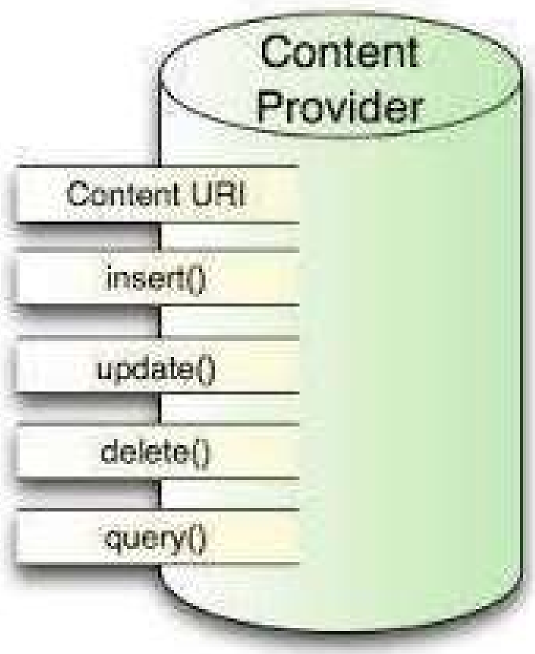


```
<prefix>://<authority>/<data_type>/<id>
```

How to create Content Provider

- First of all you need to create a Content Provider class that extends the *ContentProviderbaseclass*.
- Second, you need to define your content provider URI address which will be used to access the content.
- Next you will need to create your own database to keep the content. Usually, Android uses SQLite database and framework needs to override *onCreate()* method which will use SQLite Open Helper method to create or open the provider's database. When your application is launched, the *onCreate()* handler of each of its Content Providers is called on the main application thread.
- Next you will have to implement Content Provider queries to perform different database specific operations.
- Finally register your Content Provider in your activity file using <provider> tag.

List of methods need to override in Content Provider class to have your Content Provider working

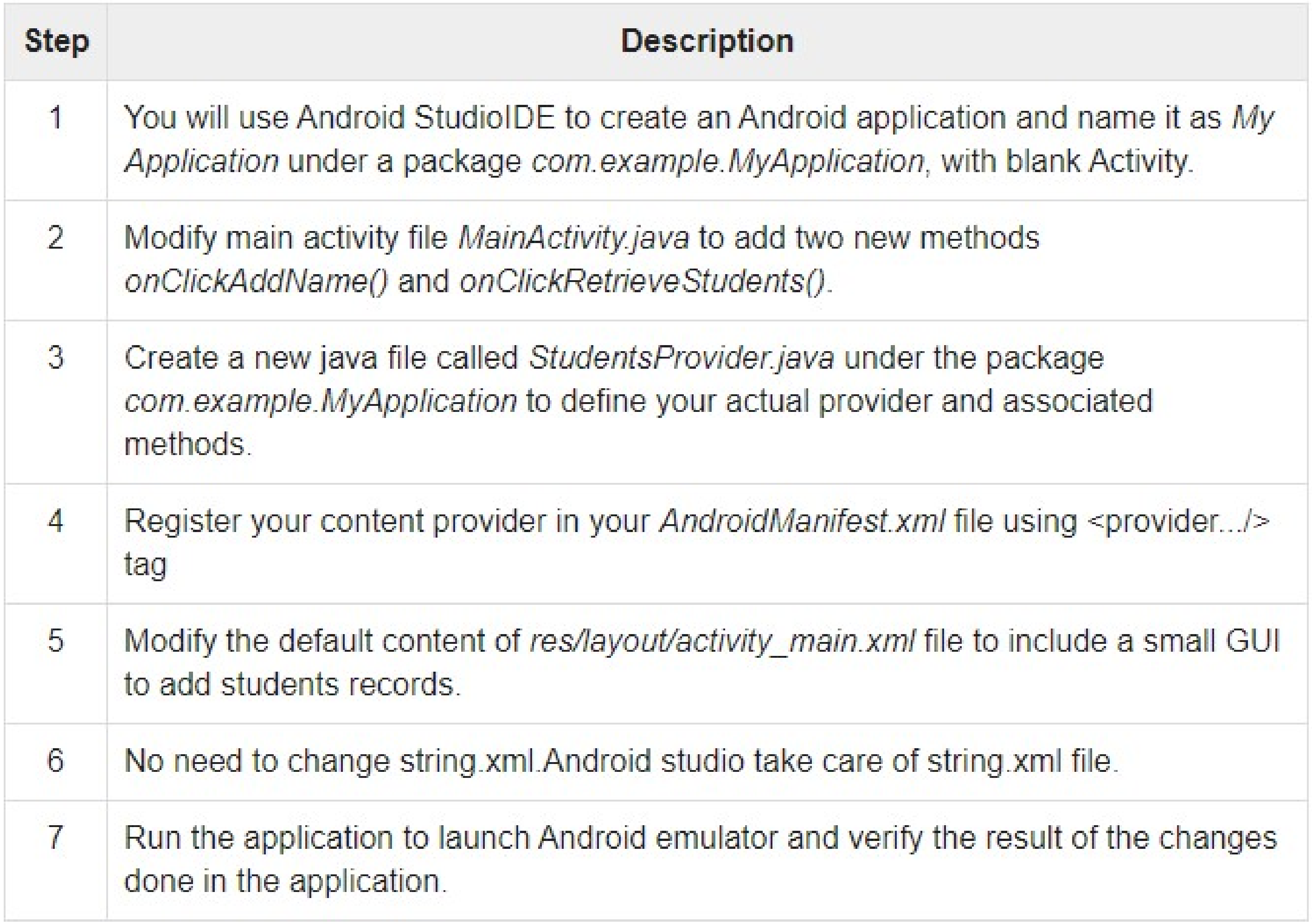


ContentProvider

List of methods need to override in Content Provider class to have your Content Provider working

- **onCreate()** This method is called when the provider is started.
- **query()** This method receives a request from a client. The result is returned as a Cursor object.
- **insert()**This method inserts a new record into the content provider.
- **delete()** This method deletes an existing record from the content provider.
- **update()** This method updates an existing record from the content provider.
- **getType()** This method returns the MIME type of the data at the given URI.

Steps to create our own ContentProvider



Step	Description
1	You will use Android StudioIDE to create an Android application and name it as *My Application* under a package *com.example.MyApplication*, with blank Activity.
2	Modify main activity file *MainActivity.java* to add two new methods *onClickAddName()* and *onClickRetrieveStudents()*.
3	Create a new java file called *StudentsProvider.java* under the package *com.example.MyApplication* to define your actual provider and associated methods.
4	Register your content provider in your *AndroidManifest.xml* file using <provider.../> tag
5	Modify the default content of *res/layout/activity_main.xml* file to include a small GUI to add students records.
6	No need to change string.xml.Android studio take care of string.xml file.
7	Run the application to launch Android emulator and verify the result of the changes done in the application.