MediaStore介绍：

MediaStore类是 android系统提供的一个多媒体数据库，android手机中多媒体信息都可以从这里提取,包括音频、视频和图像，android把所有的多媒体数据库接口进行了封装，所有的数据库不用自己进行创建，直接利用ContentResolver调用封装好的接口就可以进行数据库的操作了。

MediaStore.Files: 共享的文件,包括多媒体和非多媒体信息

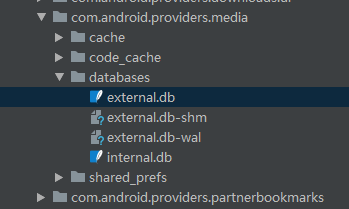
MediaStore.Audio: 存放音频信息

MediaStore.Image: 存放图片信息

MediaStore.Vedio: 存放视频信息

多媒体表

/data/data/com.android.providers.media/databases/下。



SQLite3的命令行：

1. Sqlite3 数据库名
2. Sqlite3 表名 #可以列出多媒体数据库的表
3. Sqlite3 Scheme 表名 #可以查询表中的所有列名

多媒体库操作

查询

Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder);

ContentResolver的query方法接受几个参数，参数意义如下:

Uri：这个Uri代表要查询的数据库名称加上表的名称。这个Uri一般都直接从MediaStore里取得。例如，查询所有歌的信息，要用这个Uri来查询MediaStore.Audio.Media. EXTERNAL \_CONTENT\_URI。

具体如下表所示：

|  |  |
| --- | --- |
| 名称 | URI |
| 查询所有文件 | 只能用getContentUri()或者直接使用Uri。  MediaStore.Files.Media.getContentUri("external");或者  Uri.parse("content://media/external/files"); |
| 查询图片文件 | MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI |
| 查询音频文件 | MediaStore.Audio.Media.EXTERNAL\_CONTENT\_URI |
| 查询视频文件 | MediaStore.Video.Media.EXTERNAL\_CONTENT\_URI |

projection：这个参数代表要从表中选择的列，用一个String数组来表示。

selection：相当于SQL语句中的where子句，就是代表你的查询条件。

selectionArgs：这个参数是说你的Selections里有？这个符号是，这里可以以实际值代替这个问号。如果Selections这个没有？的话，那么这个String数组可以为null。

sortOrder：说明查询结果按什么来排序。例如：MediaStore.Images.Media.\_ID+" DESC"按照id倒序查询。

上面就是各个参数的意义，它返回的查询结果一个Cursor，这个Cursor就相当于数据库查询的中Result，用法和它差不多。

说明：

URI一般有三种写法

Uri uri1 = MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI;

Uri uri2 = MediaStore.Images.Media.getContentUri("external");

Uri uri3 = Uri.parse("content://media/external/images/media");

增加

核心代码：

|  |
| --- |
| ContentValues values = new ContentValues;  values.put(MediaStore.Audio.Playlists.Members.PLAY\_ORDER,0);  resolver.insert(uri, values); |

这个insert传递的参数只有两个，一个是Uri(同查询那个Uri)，另一个是ContentValues。这个ContentValuses对应于数据库的一行数据，只要用put方法把每个列的设置好之后，直接利用insert方法去插入就好了。

例子：

ContentResolver resolver = ctx.getContentResolver;

Uri uri = MediaStore.Audio.Media.EXTERNAL\_CONTENT\_URI;

ContentValues values = new ContentValues;

values.put(MediaStore.Audio.Media.DATE\_MODIFIED, sid);

resolver.update(MediaStore.Audio.Playlists.EXTERNAL\_CONTENT\_URI,values,where, lectionArgs);

上面update方法和查询还有增加里的参数都很类似。

删除

ContentResolver resolver = ctx.getContentResolver;

resolver.delete(MediaStore.Audio.Playlists.EXTERNAL\_CONTENT\_URI,where, selectionArgs);

delete和更新的方法很类似