ASSIGNMENT -1

Android App Development -2

REVIEW REPORT ON SQLite AND CONTENT PROVIDER

Submitted by,

Kiran prasannan

Mvoc MPAD

DDUKK

contents

<u>INTRODUCTION</u>

A review for identifying about the SQLite and ContentProvider in Android.A ContentProvider provides data from one application to another, when requested. It manages access to a structured set of data. SQLite is a Structure query base database, open source, light weight, no network access and standalone database.It contains Advantages and Disadvantages of SQLite and ContentProvider.Usescase diagram for identifying the more about the topics

CONTENTPROVIDER

In Android, Content Providers are a very important component that serves the purpose of a relational database to store the data of applications. The role of the content provider in the android system is like a central repository in which data of the applications are stored, and it facilitates other applications to securely access and modifies that data based on the user requirements.

Android system allows the content provider to store the application data in several ways. Users can manage to store the application data like images, audio, videos, and personal contact information by storing them in SQLite Database, in files, or even on a network. In order to share the data, content providers have certain permissions that are used to grant or restrict the rights to other applications to interfere with the data.
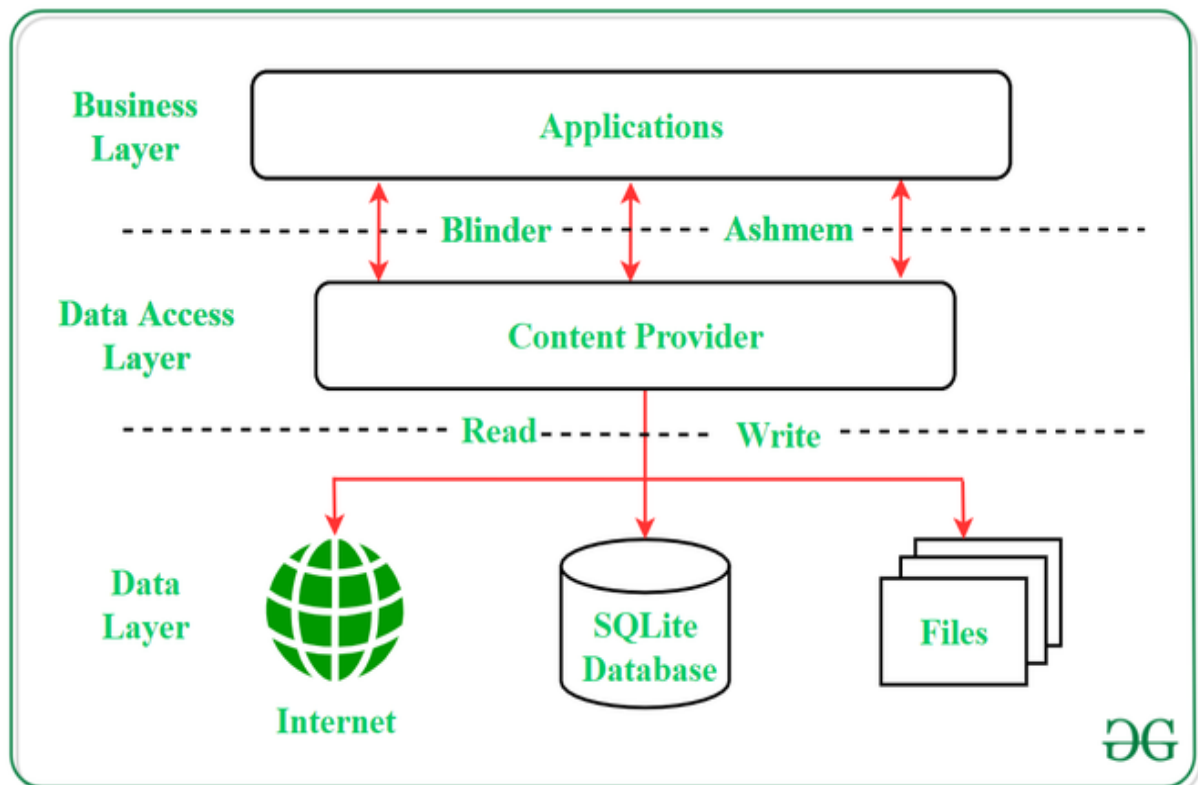
Content URI

Content URI (Uniform Resource Identifier) is the key concept of Content providers. To access the data from a content provider, URI is used as a query string.

Different parts of Content URI

- ❖ **content://** – Mandatory part of the URI as it represents that the given URI is a Content URI.
- ❖ **authority** – Signifies the name of the content provider like contacts, browser, etc. This part must be unique for every content provider.
- ❖ **optionalPath** – Specifies the type of data provided by the content provider. It is essential as this part helps content providers to support different types of data that are not related to each other like audio and video files.
- ❖ **optionalID** – It is a numeric value that is used when there is a need to access a particular record

Structure of a Content URI: *content://authority/optionalPath/optionalID*

Operations in Content Provider

- ❖ **Create:** Operation to create data in a content provider.
- ❖ **Read:** Used to fetch data from a content provider.
- ❖ **Update:** To modify existing data.
- ❖ **Delete:** To remove existing data from the storage.

Working of Content Provider

UI components of android applications like Activity and Fragments use an object **CursorLoader** to send query requests to **ContentResolver.** The Content Resolver object sends requests (like create, read, update, and delete) to the **Content Provider** as a client. After receiving a request, Content Provider process it and returns the desired result. Below is a diagram to represent these processes in pictorial form
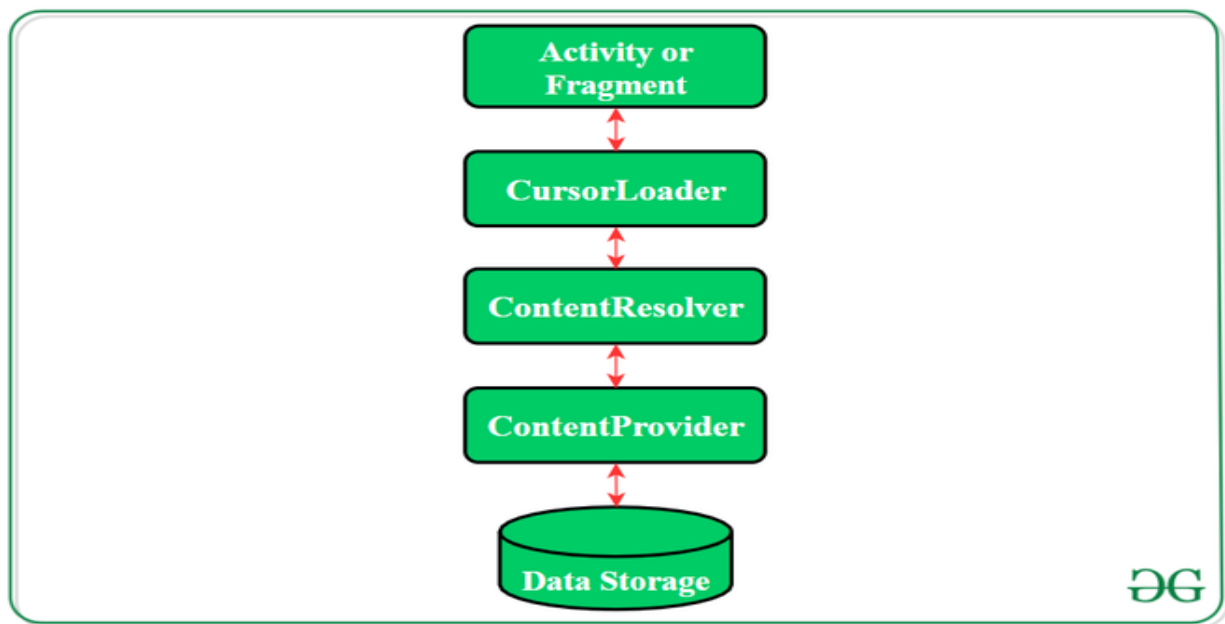
Creating a Content Provider

- ❖ Create a class in the same directory where the that **Main Activity** file resides and this class must extend the Content Provider base class.
- ❖ To access the content, define a content provider URI address.
- ❖ Create a database to store the application data.
- ❖ Implement the **six abstract methods** of Content Provider class.
- ❖ Register the content provider in **AndroidManifest.xml** file using **<provider> tag**

| Method	Description
Query ()	A method that accepts arguments and fetches the data from the desired table. Data is retired as a cursor object
Insert ()	To insert a new row in the database of the content provider. It returns the content URI of the inserted row.
Update ()	This method is used to update the fields of an existing row. It returns the number of rows updated.
Delete ()	This method is used to delete the existing rows. It returns the number of rows deleted.
getType ()	This method returns the Multipurpose Internet Mail Extension(MIME) type of data to the given Content URI.
onCreate ()	As the content provider is created, the android system calls this method immediately to initialise the provider

ADVANTAGES OF CONTENT PROVIDER

Content providers offer granular control over the permissions for accessing data. You can choose to restrict access to a content provider from solely within your application, grant blanket permission to access data from other applications, or configure different permissions for reading and writing data. For more information on using content providers securely, see Security tips for storing data, as well as Content provider permissions.

You can use a content provider to abstract away the details for accessing different data sources in your application. For example, your application might store structured records in a SQLite database, as well as video and audio files. You can use a content provider to access all of this data, if you implement this development pattern in your application

DISADVANTAGES OF CONTENT PROVIDERS

Although using content providers is a very good idea for standardizing data sharing, it involves some drawbacks. In their current state, content providers can only use SQLite as a storage mechanism. This may change in the future, though, because there are no hard-coded SQLite constraints.The arguments selection, selection arguments, and sort order are typical to relational databases. More complicated queries, with joins and distinct, are not possible

USECASE DIAGRAM
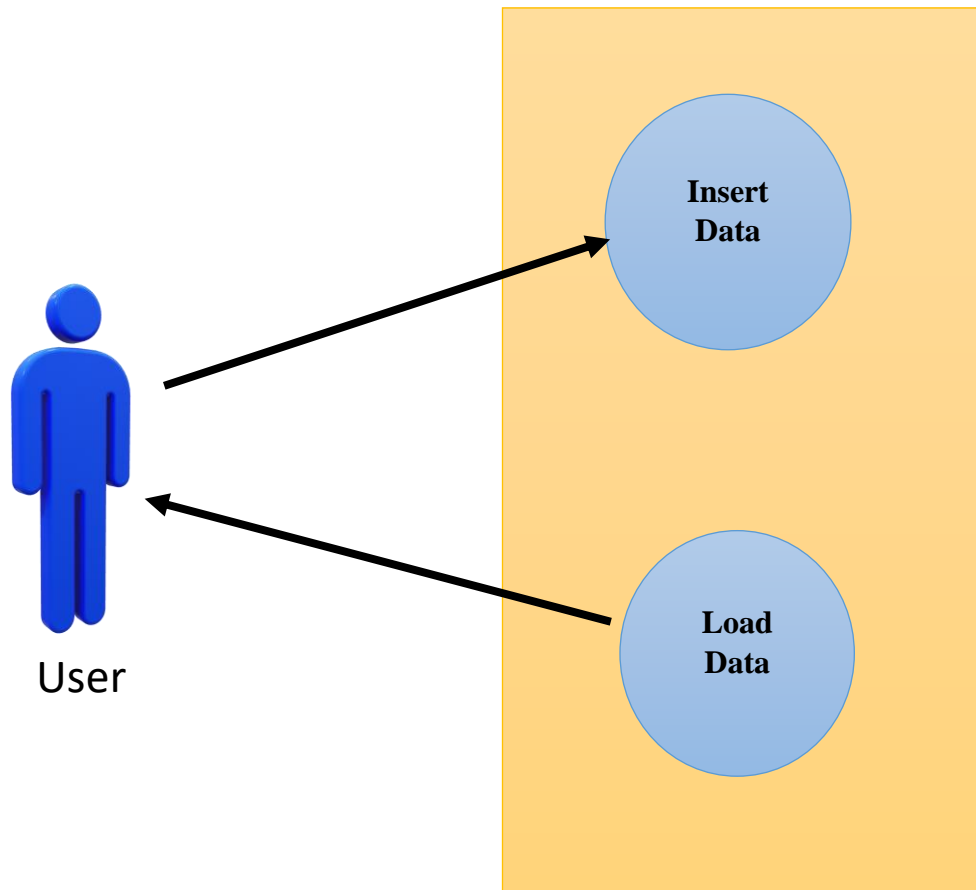
SQLITE

SQLite is a Structure query base database, open source, light weight, no network access and standalone database. It support embedded relational database features.SQLite Database is an open-source database provided in Android which is used to store data inside the user's device in the form of a Text file. We can perform so many operations on this data such as adding new data, updating, reading, and deleting this data. SQLite is an offline database that is locally stored in the user's device and we do not have to create any connection to connect to this

Important Methods in SQLite Database

Method	Description
getColumnNames()	This method is used to get the Array of column names of our SQLite table.
getCount()	This method will return the number of rows in the cursor
isClosed()	This method returns a Boolean value when our cursor is closed
getColumnCount()	This method returns the total number of columns present in our table
getColumnName(int columnIndex)	This method will return the name of the column when we passed the index of our column in it
getColumnIndex(String columnName)	This method will return the index of our column from the name of the column.
getPosition()	This method will return the current position of our cursor in our table

ADVANTAGES OF SQLITE

1) Lightweight

- o SQLite is a very light weighted database so, it is easy to use it as an embedded software with devices like televisions, Mobile phones, cameras, home electronic devices, etc.

2) Better Performance

- o Reading and writing operations are very fast for SQLite database. It is almost 35% faster than File system.
- o It only loads the data which is needed, rather than reading the entire file and hold it in memory.
- o If you edit small parts, it only overwrite the parts of the file which was changed.

3) No Installation Needed

- o SQLite is very easy to learn. You don?t need to install and configure it. Just download SQLite libraries in your computer and it is ready for creating the database.

4) Reliable

- o It updates your content continuously so, little or no work is lost in a case of power failure or crash.
- o SQLite is less bugs prone rather than custom written file I/O codes.
- o SQLite queries are smaller than equivalent procedural codes so, chances of bugs are minimal.

5) Portable

- o SQLite is portable across all 32-bit and 64-bit operating systems and big- and little-endian architectures.
- o Multiple processes can be attached with same application file and can read and write without interfering each other.

SQLITE DISADVANTAGES

- SQLite is used to handle low to medium traffic HTTP requests.
- Database size is restricted to 2GB in most cases.

USECASE DIAGRAM