Working with SQLite Cursors

Posted by Alejandro G. Carlstein Ramos Mejia on July 27, 2011 July 27, 2011 About Programming / Android

A common mistake is the improper way to use cursors. It is very common to use a cursor and forget to close it. This produce an IllegalStateException.

A code similar to the following code example will produce an error the moment you leave the activity:

```
private void readContactData(long contactDataId) {
   Log.d(TAG, 'readContactData()');
   dbContact.open();
   Cursor cursor = dbContact.getContact(contactDataId);
   if (cursor.moveToFirst()) {
     ...
   } else {
     ...
   }
   dbContact.close();
   ...
}
```

```
The problem is that you are not closing the cursor properly.

You could try the following variable of the previous code:
```

```
private void readContactData(long contactDataId) {
   Log.d(TAG, 'readContactData()');
   dbContact.open();
   Cursor cursor = dbContact.getContact(contactDataId);
   if (cursor.moveToFirst()) {
     ...
   } else {
     ...
   }
   cursor.close();
   dbContact.close();
   ...
}
```

However, there are cases that this code produce a warning when cursor calls the method finalizer(). This seems to be a problem with Droid 2.

This method seems to work better:

```
private void readContactData(long contactDataId) {
    Log.d(TAG, 'readContactData()');
    Cursor cursor = null;
    try{
        dbContact.open();
        cursor = dbContact.getContact(contactDataId);
        if (cursor.moveToFirst()) {
            ...
        } else {
            ...
        }
        dbContact.close();
    }finally{
        if (cursor != null){
            cursor.deactivate();
            cursor.close();
        }
    }
}
```

Finally, there is always the option of using *startManagingCursor*(); however, this will be deprecated and replaced with *CursorLoader* class when using API 11.