Android SQLite Java Classes

SQLite is, as previously mentioned, written in the C programming language whilst Android applications are primarily developed using Java. To bridge this “language gap”, the Android SDK includes a set of classes that provide a Java layer on top of the SQLite database management system. The remainder of this chapter will provide a basic overview of each of the major classes within this category. More details on each class can be found in the online Android documentation.

**Cursor**

A class provided specifically to provide access to the results of a database query. For example, a SQL SELECT operation performed on a database will potentially return multiple matching rows from the database. A Cursor instance can be used to step through these results, which may then be accessed from within the application code using a variety of methods. Some key methods of this class are as follows:

* **close()** – Releases all resources used by the cursor and closes it.
* **getCount()** – Returns the number of rows contained within the result set.
* **moveToFirst()** – Moves to the first row within the result set.
* **moveToLast()** – Moves to the last row in the result set.
* **moveToNext()** – Moves to the next row in the result set.
* **move()** – Moves by a specified offset from the current position in the result set.
* **get<type>()** – Returns the value of the specified <type> contained at the specified column index of the row at the current cursor position (variations consist of getString(), getInt(), getShort(), getFloat() and getDouble()).

**SQLiteDatabase**

This class provides the primary interface between the application code and underlying SQLite databases including the ability to create, delete and perform SQL based operations on databases. Some key methods of this class are as follows:

* **insert()** – Inserts a new row into a database table.
* **delete()** – Deletes rows from a database table.
* **query()** – Performs a specified database query and returns matching results via a Cursor object.
* **execSQL()** – Executes a single SQL statement that does not return result data.
* **rawQuery()** – Executes an [SQL query](http://www.techotopia.com/index.php/An_Overview_of_Android_SQLite_Databases) statement and returns matching results in the form of a Cursor object.

**SQLiteOpenHelper**

A helper class designed to make it easier to create and update databases. This class must be subclassed within the code of the application seeking database access and the following callback methods implemented within that subclass:

* **onCreate()** – Called when the database is created for the first time. This method is passed as an argument the SQLiteDatabase object for the newly created database. This is the ideal location to initialize the database in terms of creating a table and inserting any initial data rows.
* **onUpgrade()** – Called in the event that the application code contains a more recent database version number reference. This is typically used when an application is updated on the device and requires that the database schema also be updated to handle storage of additional data.

In addition to the above mandatory callback methods, the onOpen() method, called when the database is opened, may also be implemented within the subclass.

The constructor for the subclass must also be implemented to call the super class, passing through the application context, the name of the database and the database version. Notable methods of the SQLiteOpenHelper class include:

* **getWritableDatabase()** – Opens or creates a database for reading and writing. Returns a reference to the database in the form of a SQLiteDatabase object.
* **getReadableDatabase()** – Creates or opens a database for reading only. Returns a reference to the database in the form of a SQLiteDatabase object.
* **close()** – Closes the database.

**ContentValues**

ContentValues is a convenience class that allows key/value pairs to be declared consisting of table column identifiers and the values to be stored in each column. This class is of particular use when inserting or updating entries in a database table.