Mobile Applications Development 2

Android Platform

Database Access

Musfiq Rahman, Ph.D.

Assistant Professor, Computing Science Dept.
Thompson Rivers University

THOMPSON RIVERS UNIVERSITY

Outline

- *Database Support in Android*
- *Lightweight Databases*
- *About SQLite*
- *Accessing Databases from Android*
- *SQLiteDatabases & Cursors*
- *SQLiteOpenHelper*

THOMPSON RIVERS UNIVERSITY

Why Database?

- Android persists data into an internal file system.
- It provides two functional layers on top of the file system:
  - Shared Preferences
  - SQLite Database
- Shared Preferences are good way to store some (small amount of data) settings permanently.
  - Used for permitive datatypes (key, value pair)
- How to use shared preferences? (refer to the tutorial)
- Databases are another form of permanent storage in Android. They are often used to store more complex data.

Outline

- ***Database Support in Android***
- *Lightweight Databases*
- *About SQLite*
- *Accessing Databases from Android*
- *SQLiteDatabases & Cursors*
- *SQLiteOpenHelper*

THOMPSON RIVERS UNIVERSITY

Database Support in Android

- Android has built-in support for SQLite
- SQLite is an open-source database that:
  - Uses a single file for its databases
  - Uses memory for many operations
  - Is designed for simplicity
  - Has a very small footprint (< 400kb)
  - Requires zero configuration
  - Is used by Android, iOS, WP8, BB, ...
- Other databases can be installed By user or bundled (if its license permits)

THOMPSON RIVERS UNIVERSITY

Outline

- *Database Support in Android*
- ***Lightweight Databases***
- *About SQLite*
- *Accessing Databases from Android*
- *SQLiteDatabases & Cursors*
- *SQLiteOpenHelper*

Lightweight Databases

- *Databases come in two main categories:*
  - *Enterprise databases*
    - *Deployed on the server*
    - *Handle requests centrally*
    - *Often costly (e.g. Oracle, SQL Server, DB2)*
    - *Requirements: Scalability & Security & Reliability*
  - *Client-deployed databases*
    - *Installed along with your software on a client's machine*
    - *Often open source (SQLite, MySQL)*
    - *Does not add to the cost of your software*
    - *Requirements: Small footprint (memory, disk) & Low cost & Reliability*

THOMPSON RIVERS UNIVERSITY

Outline

- *Database Support in Android*
- *Lightweight Databases*
- ***About SQLite***
- *Accessing Databases from Android*
- *SQLiteDatabases & Cursors*
- *SQLiteOpenHelper*

THOMPSON RIVERS UNIVERSITY

SQLite Is Serverless

- *Most SQL database engines are implemented as a separate server process. Programs that want to access the database communicate with the server using some kind of interprocess communication (typically TCP/IP) to send requests to the server and to receive back results.*

- *SQLite does not work this way. With SQLite, the process that wants to access the database reads and writes directly from the database files on disk. There is no intermediary server process.*

Thompson Rivers University

Outline

- *Database Support in Android*
- *Lightweight Databases*
- *About SQLite*
- ***Accessing Databases from Android***
- *SQLiteDatabases & Cursors*
- *SQLiteOpenHelper*

THOMPSON RIVERS UNIVERSITY

Accessing Databases from Android

- **Android has two ways to access an SQLite database**
  - **Use Java Database Connectivity (JDBC) to access the database in a traditional way (outdated)**
  - **Use the Android-specific database access library**

$$\begin{tikzcd}[column sep=small] & \text{ } \arrow[dl] \arrow[dr] & \\ \textbf{2 level DB Design} & & \textbf{3 level DB Design} \end{tikzcd}$$

Outline

- *Database Support in Android*
- *Lightweight Databases*
- *About SQLite*
- *Accessing Databases from Android*
- ***SQLiteDatabases & Cursors***
- *SQLiteOpenHelper*

THOMPSON RIVERS UNIVERSITY

SQLiteDatabase & Cursors

- ***SQLiteDatabase*** *class in Android* exposes methods to manage a SQLite database.

- ***SQLiteDatabase*** has methods to create, delete, execute SQL commands, and perform other common database management tasks.

- Database names must be unique within an application, not across all applications.

Ref: http://developer.android.com/training/notepad/index.html

Ref: http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html

THOMPSON RIVERS UNIVERSITY

SQLiteDatabase & Cursors

SQLiteDatabase **Important methods:**

- **insert**(String table, String nullColumnHack, ContentValues values)
- **query**(String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)
- **rawQuery**(String sql, String[] selectionArgs)
- **update**(String table, ContentValues values, String whereClause, String[] whereArgs)
- **delete**(String table, String whereClause, String[] whereArgs)
- **execSQL**(String sql, Object[] bindArgs)

THOMPSON RIVERS UNIVERSITY

SQLiteDatabase & Cursors

- **rawQuery() and query() return a cursor**
  - Cursors represent the result set returned by the database
  - Important methods:

getCount()	# of rows
getColumnCount()	# of columns
moveToFirst(),moveToLast()	
moveToNext(),moveToPrevious	
moveToPosition(5)	move to row 5 (0-based)
getString(3)	get column 3 (0-based)
getInt(4)	get column 4 (0-based)
...	

SQLiteDatabase & Cursors

- **insert()** and **update()** use **ContentValues** to store data
  - ***ContentValues*** is like a list of column name/value pairs
  - Important methods:
    - put("columnName", "some value")

THOMPSON RIVERS UNIVERSITY

Outline

- *Database Support in Android*
- *Lightweight Databases*
- *About SQLite*
- *Accessing Databases from Android*
- *SQLiteDatabases & Cursors*
- ***SQLiteOpenHelper***

THOMPSON RIVERS UNIVERSITY

SQLiteOpenHelper

- SQLiteOpenHelper class acts as a lifecycle manager for databases
  - It helps to ensure that you follow best practices for client-deployed databases
- Lifecycle events exist for:
  - Create
  - Upgrade
  - Downgrade (not often used)
  - Open (not often used)

SQLiteOpenHelper

- To use SQLiteOpenHelper, you create a subclass of it
  - Your database will have a version number
  - Implement the **onCreate()** method
  - Implement the **onUpgrade()** method

SQLiteOpenHelper

- To use SQLiteOpenHelper, you create a subclass of it
  - **Your database will have a version number**
    - If you change your database structure, you should increase the version number
    - This does not have to correspond to your application version number
      - If you database does not change from version 2.0 to 2.1, do not modify the database version
  - Implement the onCreate() method
  - Implement the onUpgrade() method

SQLiteOpenHelper

- To use SQLiteOpenHelper, you create a subclass of it
  - Your database will have a version number
  - **Implement the onCreate() method**
    - Execute SQL to create your database tables
  - Implement the onUpgrade() method

SQLiteOpenHelper

- To use SQLiteOpenHelper, you create a subclass of it
  - Your database will have a version number
  - Implement the onCreate() method
  - **Implement the onUpgrade() method**
    - Execute SQL to modify the structure of your database
    - This is far more difficult, if you have many versions
      - It takes two extra arguments:
        - The old version number
        - The new version number
      - A good approach is to upgrade step-wise
        - e.g. Upgrade from 2.0 to 2.2
          - 2.0 ‣ 2.1
          - 2.1 ‣ 2.2

SQLiteOpenHelper

- Many Android developers also use this subclass as the main data access object for their data
  - Normally, you'd create a separate subclass for each table available
  - In this subclass, you can create methods for creation, reading, updating, and deletion (CRUD)
  - The idea is to keep the CRUD objects simple, by handling only a single table

Outline

- *Database Support in Android*
- *Lightweight Databases*
- *About SQLite*
- *Accessing Databases from Android*
- *SQLiteDatabases & Cursors*
- *SQLiteOpenHelper*

THOMPSON RIVERS UNIVERSITY

ANY QUESTIONS?