



MIT Global Startup Labs

<http://aiti.mit.edu>

Indonesia Summer 2013
Meetup 06 – Database Management



Today's Meetup

- Database overview
- Setting up the Database
- Manipulating the Database
- Today's Assignment

Database Overview

How do we store data?

- Shared Preferences
 - Store private primitive data in key-value pairs.
- Internal Storage
 - Store private data on the device memory.
- External Storage
 - Store public data on the shared external storage.
- SQLite Databases
 - Store structured data in a private database.
- Cloud
 - Store data on the web with your own network server.

The SQLite Database

- The Android OS provides a built-in database management system called SQLite(a DB system specialized for embedded devices)
- Each Android application can have its own SQLite database, but may not access the database of any other application (for security)
- Advantages of SQLite:
 - Uses standard SQL syntax
 - Open-source, zero-configuration (no effort required by developer to set up the DB before using it)
 - Each SQLite database exists in its own, single file (very secure)

	unique Tweet ID	created_at	txt	user
Tweet 1:	ID#1	Date/Time	"What up?"	Markus
Tweet 2:				
.				
.				
.				

Setting up the Database

SQLiteOpenHelper

- Helper Class to manage database creation and version management
 - Connection to database
- To manipulate databases use
 - onCreate(): create SQL schema
 - onUpgrade(): what happens when DB changes
 - Optionally onOpen()

DbHelper

Create a subclass of
SQLiteOpenHelper

Implement Constructor

onCreate() *

onUpgrade()

```
public class DbHelper1 extends SQLiteOpenHelper {
    //define database name, version, other constant
    static final String DB_NAME = "timeline.db";
    static final int DB_VERSION = 1;
    Context context;

    // Constructor
    public DbHelper1(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
        this.context = context;
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        //create SQL schema (use SQL code)
        //run db.execSQL() on created schema
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        //define what happens when db is altered
    }
}
```

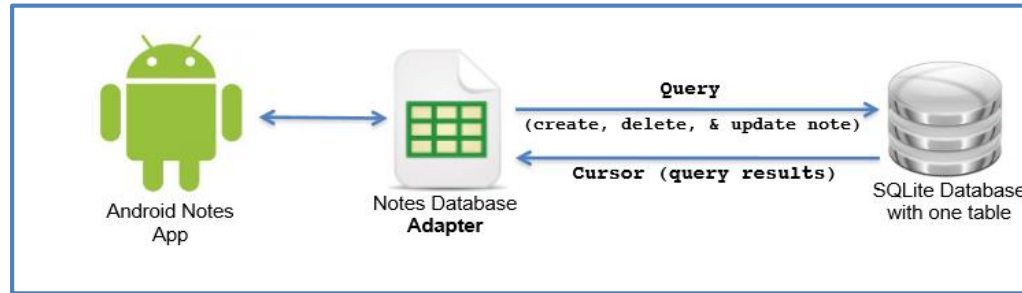
* Example to set up db:

- Save SQL code in String
- Execute SQL code

```
String sql = "create table " + TABLE + " (" + C_ID + " int primary key, "
    + C_CREATED_AT + " int, " + C_USER + " text, " + C_TEXT + " text)";
db.execSQL(sql);
```


Database Operations

Four major operations



- `insert()`
 - Inserts one or more rows into the database
- `query()`
 - Requests rows matching the criteria you specify
- `update()`
 - Replaces ones or more rows that match the criteria you specify
- `delete()`
 - Deletes rows matching the criteria you specify

Write to database

Open database for writing

Get name-value pairs
online

Insert ContentValues() in
database *

Close database

```
//open database for writing
db = dbHelper.getWritableDatabase();

//create new name-value pair data structure
ContentValues values = new ContentValues();

for (Twitter.Status status : timeline) {

    //empty variable type ContentValues()
    values.clear();

    //put name-value pair in variable type ContentValues()
    values.put(DbHelper.C_ID, status.id);
    values.put(DbHelper.C_CREATED_AT, status.createdAt.getTime());
    values.put(DbHelper.C_SOURCE, status.source);
    values.put(DbHelper.C_TEXT, status.text);
    values.put(DbHelper.C_USER, status.user.name);

    //put values in database
    db.insertOrThrow(DbHelper.TABLE, null, values);
}

//close database
db.close();
```

* [insertOrThrow\(String table, String nullColumnHack, ContentValues values\)](#)

Read database

Open database for reading

Get data from database *

Print data in output

Close database

```
//open database to read
db = dbHelper.getReadableDatabase();

// Get the data from the database
cursor = db.query(DbHelper.TABLE, null, null, null, null, null,
    DbHelper.C_CREATED_AT + " DESC");

// Iterate over all the data and print it out
while (cursor.moveToNext()) {
    user = cursor.getString(cursor.getColumnIndex(DbHelper.C_USER));
    text = cursor.getString(cursor.getColumnIndex(DbHelper.C_TEXT));
    output = String.format("%s: %s\n", user, text);
}

//close database
db.close();
```

* [query](#)(**String** table, [String\[\]](#) columns, [String](#) selection, [String\[\]](#) selectionArgs, [String](#) groupBy, [String](#) having, **String** orderBy, [String](#) limit)

Today's Assignment

Today's Assignment

Finish off services and carry on working on databases.

All of you should have a working timeline by the end of Friday (i.e. worked through chapter 10)

I recommend working through Chapters 11 & 12,
Broadcast Receivers and Content Providers

Documentation: Gargenta – Learning Android: Chapters
9 and 10