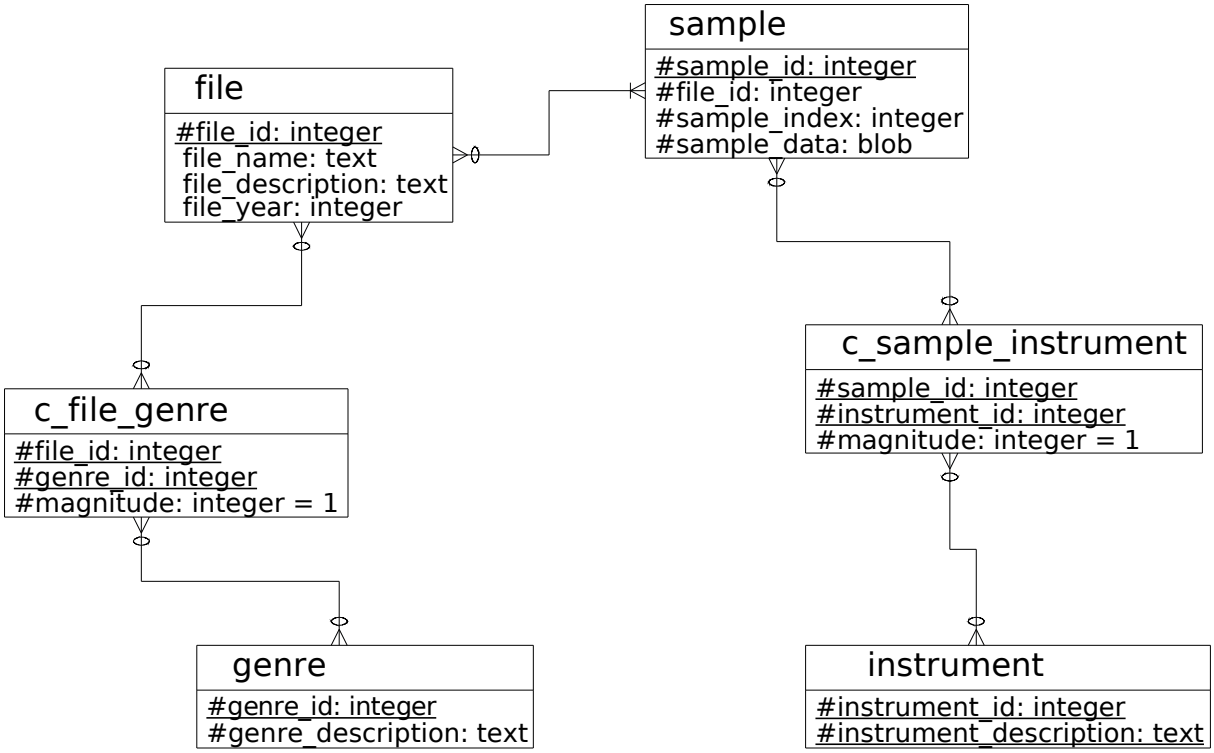
Here's what I know this far into the project:

- What I've figured out how to do:
  - Set up a single-file, network- and operating-system-independent relational database system using `SQLite` v3.3.8.
  - Access to said database from within `Scilab`. Figuring out how to split output from the query into took the better part of and hour. Go figure.
  - Extract samples from uncompressed PCM wav files from within `Scilab`. Result is a vector of **double**s.
- Problems into which I've run:
  - `MATLAB` is **not** free software and does not run on Linux. This would make it extremely hard for me to develop a program in it, as my computer, to which I have unlimited access, has only Linux on it. This noted, I've chosen to use `Scilab` v4.0 instead. I chose it for a variety of reasons:
    - Despite not having found a `Scilab` library backend for access to `SQLite` databases, I have found a library function to execute a `Perl` script (`perl()`). I have hence built my database access around `Perl`'s `SQLite` module (`DBD::SQLite`).
    - As I mentioned before, `Scilab` is completely free and cross-platform, as is `Perl` and `SQLite`. This is important to me, for many times I am stuck using Windows to write code. `Scilab` for Windows even has a folder into which one may copy the Windows `Perl` distribution, hence giving it seamless use of the `perl()` library function.
- What I still have to do
  - Develop an easy interface for listening to and cataloging the sample data.
  - **Listen to and catalog each half-second clip and determine instrument lists and emotional context. This will take a very long time and will be very tedious.**
    - Choose a sample size that will:
      1. Be small enough to determine whether the sample lies directly on the beat (said data being perhaps represented in a new column in sample)
      2. Not be too tedious to look at, listen to and catalog individually.
    - I might have to hire some help with this one.
      - Never mind. I don't trust anyone to do this but myself.
  - Find a large set of mathematic functions by whose use I might generate some sense of similarity of samples and/or sample sets.
    - Weed out functions with no correlation to statistical determination.
      - Figure out what functions give viable correlations to what statistics.
    - Weed out rogue samples (with either a `Scilab` function or by eye) the output of said functions.

- The database:
  - The diagram of the database. This is far from concrete. Right now it's just to hold the samples as I categorize them.




  - Data structure:
    - Songs are first down-mixed into mono. This makes the stream of bytes easier to deal with.
    - I plan to add all music data to the database, then begin fleshing out the associated data. This means having uncategorized samples floating around somehow associated with genres. I will have to do some sort of join to weed them out of the returned data.
  - Shortcomings of the current database setup:
    - Current structure is just out of reach of being able to group samples into “texture” samples (for use in cataloging beat structure, for instance).
    - Current structure does not allow for emotional data.
      - Music is art, and it may be incredibly hard to determine the state of mind in which the artist wrote the song.
    - Songs that change genre mid-file are not supported.
    - I have no idea how I'm going to use `Scilab`'s `wavread()` function, which returns **double**s, with `SQLite`'s “blob” (binary) data type.
      - I may have to truncate all samples not of the file's original blended genres and/or put them into an additional file entry.
    - Database was designed by a moron.
- General collected data:
  - Interviews:
    - James Roy (music professor; piano player)
      - instruments do not necessarily map to genre

        - genre is a combination of rhythmical and theoretical aspects of music
        - classical vs. pop piano
            - rhythm-based
            - different chords used
            - pop is more syncopated ← more variation in rhythm; loudest notes vary between patterns of on/off-beat occurrence.
            - classical is very regular
            - classical has chords, standard ← 4 notes apart
            - pop has more complex chord structure
        - jazz & blues use 1-4-5 chords [whatever that means]
    - Craig Hara (music professor; liver and breather of jazz)
        - jazz has different types of overlaid rhythms
            - insight: more than 3 layers of rhythm == jazz?
            - Example:
                - snare drum
                    - beats 2&4
                - cymbal
                    - beats 1,2,3&4
                - kick drum
                    - between beats 4&1
                - piano played as a beat
                    - high notes on beats 1&2
                    - low notes on beats 3&4
            - usually 4 beats per measure
        - jazz uses bass guitar... a lot.
        - gave me CD of **Benny Carter** for study
- Data set:
    - Since most of the music I like defies classification into traditional genres, I visited the library to collect music I hate. Of the CDs I picked up are the following:
        - Jazz: Duke Ellington, Charlie Mingus, Max Roach – **Money Jungle** [1962]
        - Classical: Beethoven, Ludwig Van – **Missa Solemnis** [1818]
        - Calypso: **Steel Band Music of the Caribbean** [traditional]
        - Blues & Acid Rock: Carlos Santana – **A Profile of** [60's]
        - Big Band: Michael Davis – **Absolute Trombone** [1997]
        - Big Band: Michael Davis – **Brass Nation** [2000]
    - Of the easily-categorized music from my personal collection I have selected:
        - Industrial & Metal: Nine Inch Nails – **The Downward Spiral** [1994]
        - Punk & Ska: Operation Ivy – **Energy** [1990]
        - Hard Rock: The Midnight Evils – **Straight Til Morning** [2002]
        - Metal: Fear Factory – **Obsolete** [1999]
        - Techno & Pop: Lamb – **Self-titled** [1996]
        - Rock & Pop: Marcy Playground – **Self-titled** [1997]
        - Pop: Motion City Soundtrack – **I Am the Movie** [2002]