

# COMP3005 Assignment 2

## Problem 1 - Music Database

The thought process is vaguely outlined below.

CD
Title
Artist
Release Date
Label
Producer
Songs(collection)

**Note:** "Various Artists" is used for CD's that are a compilation

### Thoughts:

- Label or producers could be a key, but besides that there's nothing. Titles, artists, release dates, and songs can all share names. This is why I added an ID field

Song
Title
Author
Publishing Company
Lyrics
Tracks(collection)

### Thoughts:

- The lyrics of the song can't be a key. Covers of this song may have the exact same lyrics, so I can't go by that.
- The title, author, and publishing company can't be the key for a song.

Track
Instrument
Name(could be the key)

## Database Grammar

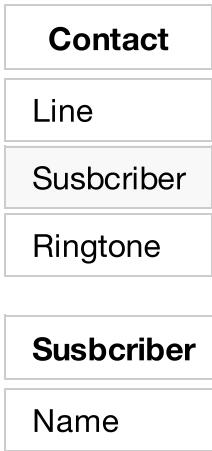
- One CD must contain zero or more songs
- One song may be on one or more CDs
- One song may consist of one or more tracks
- One track must be in one or more songs

Please refer to the accompanying E-R Diagram

## Problem 2 - Custom Ringtones

The thought process is vaguely outlined below.

In order to make a "contact" which will have a ringtone assigned to it, it needs to have 3 things:



**Note:** This would be a weak identity, since a subscriber needs a line in order to exist. Name could be a partial key

## Ringtone

- Title

## Database Grammar

- One subscriber must have one line
- One line must have one subscriber
- One subscriber may have many lines as contacts
- A line may exist for one susbciber as a contact
- A line may have one ringtone for a contact
- A ringtone may have one line in a contact

Please refer to the accompanying E-R Diagram

## Problem 3 - Search by Chord Sequences

The database would need to store the positions of the chords within the song, as well as which bar they're in. That must mean that a chord object could be like:

<b>Chord</b>
<b>Chord</b>
Value(the actual chord, perhaps a string)
bar_position
chord_position

### Thoughts:

- The chord position would store the index of the chord relative to the song
- The bar position would be the same for a number of chords in a song, since there are more than one chord in a bar

Of course, to be able to store books and users we'll need other additional objects

<b>Book</b>
title
page_offset

### Thoughts:

- The title of the book can't be the key, since other books could have the same title. I may make a separate ID for this

<b>User</b>
username
password

### Thoughts:

- The username can be the key

<b>Song</b>
title
chords
page_num

## Search

In order to be able to select chords by sequence, you'd have to make a complicated select statement. For example, to select one of the more common sequences: E, B, C#m, A; we'd have to do the following select statement:

```

select * from chords where chord="E" and position=(  

    select position from chords where chord="B" and position=(  

        select position from chords where chord="C#m" and position=(  

            select position from chords where chord="A"  

                )-1  

            )-2  

        )-3;

```

The above SQL should return the first row where the E, B, C#m and A chords appear in sequence.

Please refer to the accompanying E-R Diagram

## Problem 4 - Food by nutritional value and cost

---

### Preface

In the summer of 2014 I was going through a period where I was trying to gain weight. I realized the importance of nutrition at this time and I found it difficult to sustain my diet with a student budget as well as my limited dining options. It's a busy life, and I didn't always have time to cook food that I could accurately count. This idea was simply something that I came up with during this time, and I'm not sure of all the fine details, but I have a general idea of how I'd like the final product to look.

### Description

The database I'm proposing would be used for people that are looking for very specific nutritional information at a restaurant with respect to money. There exist a lot of apps in the market that can track nutritional information, and contain a wide variety of different foods and their stats, but don't contain any prices, and don't allow you to search for the biggest "bang for your buck".

Here are a couple of example use-cases for this application:

1. A user wants to know what item on the menu simply contains the most calories per dollar.
2. A user has few calories remaining for their day, the calorie spread being 19% fat, 44% protein, and 37% carbohydrates. They want to know what they can purchase at the restaurant that's going to bring their total macronutrients to that day to the ideal 20% fat, 30% protein, and 50% carbohydrates.

### Example Data

At this point I'm not entirely sure of all the fields that I would store in the database, but here's an example of what kind of things I'd like to see. Since this is a simple example, it doesn't show relational elements (like keys).

#### Restaurants

Obviously to be able to store information about different menu items, they would need to be separated into their different host restaurants.

name
------

McDonald's
Wendy's
Taco Bell

## Users

I want users to need to sign up in order to use the app, for more accurate results. For example, 20pc Chicken Nuggets at McDonald's has the highest protein content of all their foods, so search for protein-heavy food items might yield that in the results. If I hate Chicken Nuggets, I can tell the app that and Chicken Nuggets will no longer be shown in my results.

username	email	password
mcdguy123	JohnStevenson@aol.com	912ec803b2ce49e4a541068d495ab570
bestwestern	ACarver@gmail.com	6a204bd89f3c8348afd5c77c717a097a
ihatefries97	chickenburger185@mysite.com	289dff07669d7a23de0ef88d2f7129e7

## Menu Items

Obviously the app is geared towards filtering menu items at different restaurants, so we'll need a table somewhere containing menu items.

name	calories	carbohydrates	fat	protein
20 White Meat Chicken McNuggets	920	53	58	48
Double Quarter Pounder with Cheese	740	42	43	47
Angus Third Pounder with Bacon and Cheese	760	53	42	47

There are many things that I didn't include so far, like the storage for a user's "disliked" foods or perhaps their "favorites", but ideally these are data that would be present in the final version of the database.

## Legal Issues

Such a project would be obviously intimidating because I'd be copying and hosting the nutritional information from various high-profile fast-food chains, and generally the larger corporations have the resources to bully people hosting this data into removing it.

Despite this, my belief is that this project idea is perfectly legal.

### 1. There are many other nutrition databases

Various online databases exist that store nutritional information for a ton of foods. The most popular app for calorie counting and storing/retrieving nutritional information must be MyFitnessPal(<http://www.myfitnesspal.com/>). Copying the individual values for each of the macronutrients may not be of issue, because McDonald's can't own a certain number of calories.

## **2. Names can't be copyrighted**

Instead, a name can only be trademarked. As long as I use the appropriate symbol I should be allowed to use their names.

## **3. McDonald's cares about nutrition**

Since Supersize Me, McDonald's has been fighting to maintain its clientelle. There are many foods at McDonald's that are alarmingly unhealthy, but they even go as far as to provide tools to track it now on their website.

The FDA now requires that large chains post their nutritional data, and it's increasingly beneficial for them to be transparent about what's going in their food anyways.

---

Considering the above points, I think that it would be unlikely to get into trouble just by cataloging their menu and the nutritional information from it.

Notice that I didn't mention the prices — this is because the prices vary between restaurants and countries they can be found in. I haven't decided whether I'll be including base prices for each or whether I'll have the user enter in the price information on their own.

Also notice that I've given McDonald's as an example for the above points, but they all apply to other fast-food joints as well.

## **Conclusion**

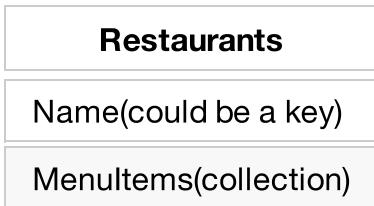
I've described the inception, application, and domain for this database throughout the document, and I'm fairly certain everyone is at least familiar with the *concept* of nutrition even if they don't know how to properly nourish themselves. It's a straightforward idea that's complex enough to be able to implement it as a database project for COMP3005.

Any questions regarding this may be directed to me, and I'd be more than happy to provide you with further information on this idea.

## **ER Diagram**

I've included an ER diagram that describes this project database.

Here's an overview of why I did what I did.



### **Thoughts:**

- I think it's reasonable to assume there won't be two restaurants with the same name, so the name can be the key for this table.

MenuItems
Name
Calories
Fat
Carbohydrates
Protein

### Thoughts:

- The values for fat, carbs and protein would be in grams. Each menu item won't contain its price because of regional differences. Instead, price is something that the user will enter himself(which could be tedious)

Users
username(key candidate)
password
email(key candidate)

### Thoughts:

- Both the username and the email could be the key, so we'll make them the combined key.
- Users also should have the ability to say which menu items they like or dislike, so they show up more often or show up less.
- In the future, this database could be augmented to store purchase history with dates and times. I'm not sure about that feature yet, so I'll think about it.

I didn't include a database grammar for this database, I don't believe it will be necessary since I've communicated my thoughts here.

## Sources

---

Trademark use: <http://www.tmweb.com/dodofuse.asp> Trademark symbols:

<http://www.uspto.gov/trademarks/basic/register.jsp> Requirement to post nutritional information:

<http://www.fda.gov/Food/IngredientsPackagingLabeling/LabelingNutrition/ucm248732.htm#menu>