**Fall 2014, 336 PROJECT**

Your objective will be to pitch the database with web interface to investors (*the Imielinski VC group*). You can pick one of the three topics below. It is a competitive project but you may get an A without having to win it or even be in the finals. If you are a finalist you probably get an A, but this is not guaranteed either.

You will have to make convincing business sense (OK, no balance sheets and business projections – we are not in Business School!).

What I mean by business sense is that you have to identify your audience and benefits which it will have from your product.

Finalists will be asked to deliver presentations and the winner will be selected on the basis of the presentation and demo of the product.

Pick one of the following schemes:

1. **Food**

Imagine that you have data about all restaurants in the area (your city, state, country?) in addition to what these restaurants have on the menu, on different days. We also know who ate at each restaurant and what they ordered. This is the CORE scheme which you can extend anyway you want to as long as it is realistic and interesting

Such data would have some value…..and probably raise some serious privacy issues!

You are pitching a new company to investors. You have unique data (!) which will make a good business case to restaurant owners or to patrons. Ratings? Annual sales? Ingredients? Names of different menu items (you can either find some on the web or may be generate yourself, you can show some humor here ☺

Once you settle of the db scheme, you will have to populate it with realistic tuples. By realistic, I mean names of bars, drinker names, dollar figures etc when appropriate. No a1, b1, c1! No patron X and restaurant Y! Generate and load your db with the large number of synthetic tuples, may be 10,000? May be more. It is your choice.

But your instances should not be completely random. On the contrary – you should think of a few patterns which will have business significance. For example, French cuisine restaurants are usually more expensive than Italian cuisine restaurants. Frog and a Peach restaurant has more loyal patrons (say, come more than once a month and rarely go to other resturants) than any other restaurant in New Brunswick (yes, location can be a nice piece of data).

Embed a number of such patterns in your data and construct SQL queries which would prove that indeed your db satisfies such pattern. Your patterns should be intuitive and have actionable business value either for a bar o for a drinker.

Notice, you are not going to load your database manually (just in case you wondered). You will write a program (that one is not in SQL) which will load the database for you. Data will be “almost” random, except the pattern(s). The logic of the pattern will be part of your program.

Thus, there are following tasks:

1) Scheme definition

2) Realistic db instance generation and db loading + Pattern embedding in the instance as well as validation of the pattern using SQL query

3) Simple web interface to best demonstrate features that you implemented.

The project will be structured as competition. The winner will win the title of ....we will find a nice name and nice AWARDS.

**B) Music**

Imagine that you have data about all music venues which play live music. You know who is performing in each venue. Additionally, you know the names of fans who attended each concert.

This is the CORE scheme which you can extend anyway you want to as long as it is realistic and interesting.

Such data would have some value…..and probably raise some serious privacy issues!

Imagine you are pitching a new company to investors. You have unique data (!) which will make a good business case to music venue owners or to music fans. .Ratings? Ticket sales. Names of bands and venues are up to you (you can either find some on the web or may be generate yourself, you can show some humor here ☺

[[the rest of the description is the same as for scheme A]]

**C) Games**

Imagine you have data about computer games. You have game categories, operating system, price, year of creation, you also know every player who ever played each of these games and how often, when was the last game etc. This is the CORE scheme which you can extend anyway you want to as long as it is realistic and interesting.

Such data would have some value…..and probably raise some serious privacy issues!  
  
Imagine you are pitching a new company to investors. You have unique data (!) which will make a good business case to game developers and gamers.  Ratings? Game suggestions? Game categorization? Most probable audience guesstimate? Different kinds of games and different game names (you can either find some on the web or may be generate yourself, you can show some humor here.

Once you settle of the db scheme, you will have to populate it with realistic tuples. By realistic, I mean names of games, gamers,  developer companies and operating systems, etc when appropriate. No a1, b1, c1!   No game X and gamer  Y!  Generate and load your db with the large number of synthetic tuples, may be 10,000?  May be more. It is your choice.

  But your instances should not be completely random. On the contrary – you should think of a few patterns which will have business significance.

For example, ages of people playing educational games must be less than those playing an RPG or a strategy.  Indie games would have less players than 3D shooters. Linux OS supports less games than any other.

Embed a number of such patterns in your data and construct SQL queries which would prove that indeed your db satisfies such pattern.  Your patterns should be  intuitive and have actionable business value either for a gamer or a game developer."

**MILESTONES**

1. Submit your scheme definition described in English, ER diagram and Relational Tables through Sakai (further instructions will be sent shortly) by end of the day, **October 16th**
2. Populate your database (VM) with instances, embedded patterns and SQL queries which verify that pattern is satisfied by the data by **November 6th**
3. Create GUI in java (or PhP) which will best feature your data. Create power point presentation which would be a “pitch” for your solution to potential “investors” (that is us, TAs and me ☺– **by November 20th**
4. Final submission – by **December 1**

***Frequently Asked Questions***

1. Can I implement all of this on my own machine?
2. How many patterns should I embed in my data? Answer: At least one, but good one. If you have more, great.
3. How many tuples should be in my database instance? Answer: at least 1000, but make it a parameter so you can load more or less.
4. How many tables and/or attributes should I have? Answer: What will matter is how interesting, natural is your data and whether you designed your tables properly. So no need to fudge the numbers
5. How do I get an “A”? Answer: You do not need to win to get an A. But to have your project a part of Hall of Fame you need to be in the finals. What will count is how interesting and practical is your data, whether it works and how does it look. The interface appeal will matter
6. I have never implemented web interface , how do I learn this? Answer: we will offer recitations which will help you to set up such interfaces.