Westfälische
Wilhelms-Universität
Münster

Exercises for Data Integration
Winter Term 2010 — Prof. Dr. G. Vossen, WWU Münster

DBIS Group
Databases &
Information Systems

Exercise sheet no. 3

Deadline: Sunday - Febuary 06, 2011, 23:59 CET

Building Mashups

Exercise 5 *(30 Points)*

In this exercise you will apply the basic knowledge about mashups you have acquired in class to a practical idea. While working on mashup problems, you might encounter typical data integration problems, such as inconsistent data, inconsistent data types and other data extraction problems. In this exercise, you might encounter such problems but on a very small scale. Then you will need to use different APIs to get most of the needed information.[1] In order to solve the problem, you are going to write a small program that integrates different sources and the information from different APIs. Your program should also be able to output a "mashup" generated from these sources. The program should be written in the PHP (5.0 and above) programming language.

Scenario - WikiLeaks releases the names of the best 864 restaurants in the United States

WikiLeaks is a non-profit media organization dedicated to bringing important news and information to the public. WikiLeaks publishes material of ethical, political and historical significance while keeping the identity of the sources anonymous, thus providing a universal way for the revealing of suppressed and censored injustices.[2] In recent times, WikiLeaks has published numerous confidential documents concerning the US Government.

Yesterday, WikiLeaks has released the names of 864 restaurants in the United States. These restaurants are said to be the best restaurants in the US and are owned by top Government officials of the United States. The food in these restaurants is excellent, creative and ultra fresh. With a focus on organic components, each menu is a perfect fusion of the cuisines of the Mediterranean region. Visitors are thrilled with these menus and their diversity. The services at these restaurants are wonderful. Every visitor always recommended these restaurants: "these are places one must visit".

Data Sources

The names of these restaurants can be found in the file "*restaurant.csv*". After going through the list, you might notice that the source does not really contain the names of 864 different restaurants; there are 112 duplicates in the list. The Duplicate Detection (DuDe) toolkit was used to search for tuples that represent the same real-world object in this data source. The output of the duplicate detection can be found in the file "*restaurant_dude.csv*". Import both datasources into a **MySQL** database for usage.

The source Restaurant contains information about the `name`, `address`, `city`, `phone-number` and `type` (category) of a restaurant. This information is very important for the mashup application.

[1] See http://www.programmableweb.com/apis/directory for information about the different APIs. Note: The usage of most APIs require registration.

[2] Source: http://213.251.145.96/ last called on the 10th of January 2011.

Your Task

In this exercise, you are supposed to firstly construct a mashup showing these restaurants on top of a map (**Bing**, **Google Maps** etc.) If there is more than one restaurant in the same place - dependening on the zoom factor - you should show these restaurants as clusters using the amount of restaurants in this area to categorize them (see Figure 1).

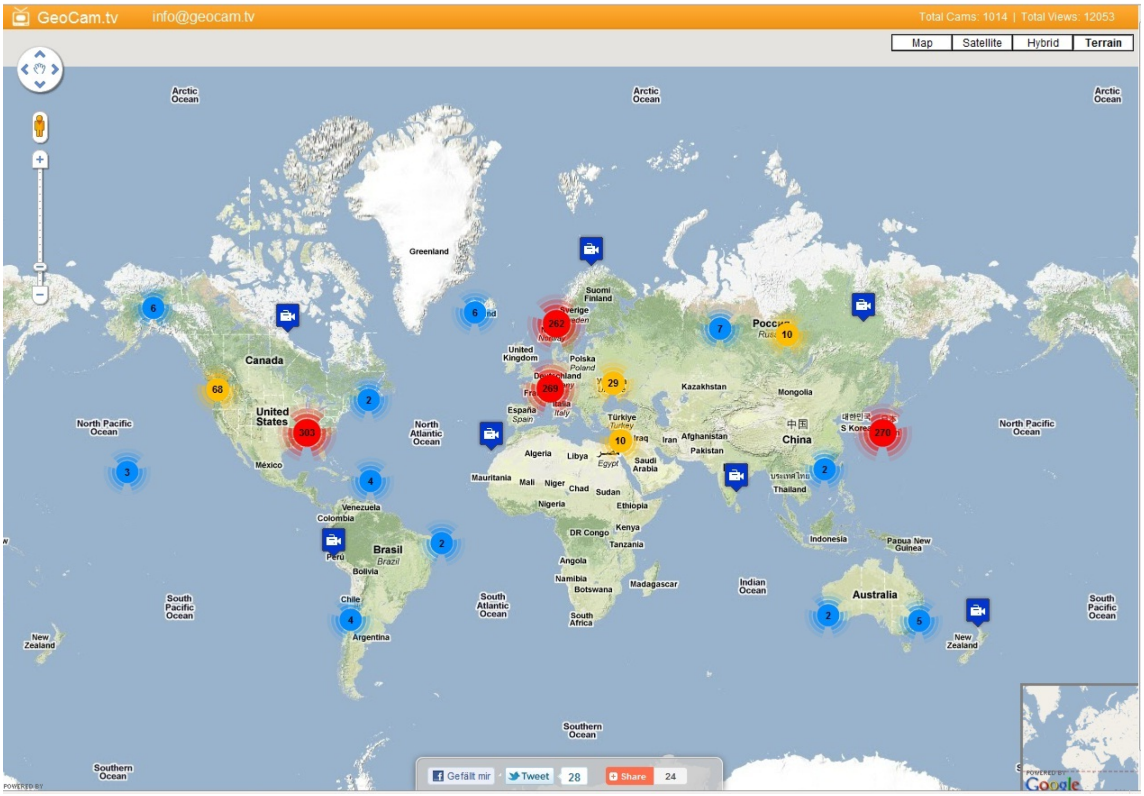

Figure 1: GeoCam as a sample application[3]

Secondly, it should be possible to get additional information concerning a particular restaurant. If you click on a bubble depicting a particular restaurant, it should give other relevant information to this restaurant. For example something like a Google Map picture of the restaurant should be shown on the top left corner of the screen, the address on top middle side. The actual weather situation of the city in which this restaurant is residing and some visitors recommendations should be shown on the top right corner of the screen. To get information about the weather use weather services like **Google Weather**, **Yahoo** etc. Visitor recommendations could be gotten using **QYPE** or any other recommendations API you know. There should also be an option for visitors to see the menus from the menu card of this restaurant. To get menus for a given restaurant use the **BigOven Recipe API** or any other recipes API using the `type` information of the restaurant as attribute. Limit this list of the menu card to 30 randomly choosen items. The **Fast Secret API** should also be used to get health information of the different menu card items. Additionally, you should also use the **Pizza Rat Restaurant Health API** to get information from the US Health Agency concerning the health inspection scores of the particular restaurant.

On the bottom left side of the screen, something like a slide or diashow should be implemented using pictures of the particular city from **Flickr** or any other APIs you know. See this like a sight seeing advertisement of this particular city. On the other part of the screen (bottom right), you should list and link every other restaurant in this particular city if there is any.

[3] Source: http://www.geocam.tv/

Parameter	Source 4
Username	`studentWS10`
Password	`1994Inmon`
Server	`localhost or 127.0.0.1`
Port	`3306`
Database	`dataint_source_4`

Table 1: Connection parameters for the relational data source.

Your Submissions

You should submit the following:

a) **List of used APIs as Word- or PDF-document** and further information concerning these APIs.

b) **PHP frontend which generates the mashup**.

c) **SQL dump of your database.**[4] Use the parameters given in Table 1.

Please submit your final project in a ZIP archive by email to Alvin Ikenna Obih. Use ”[Data Integration] Solution to Exercise 3 - Group *n*” as the subject where *n* is the group number as indicated in the announcements. The body of the email should state clearly which group is submitting and also include the first and last names as well as the matriculation numbers of all group members. **Remember the deadline: All solutions that are not received by our mail server by Sunday - Febuary 06, 2011, 23:59 CET will not be graded.** As usual: If you have any questions, feel free to discuss them in the LearnWeb forum.

Scoring

This exercise is worth 30 points. The points are awarded and reduced according to the following non-exhaustive list of criteria:

- The output format has to be adhered to very strictly. If the output format is not valid according to the task description, a maximum of two thirds of the points can be achieved.
- The program must be easy to run. If the program needs manual modification[5] in order to run, up to one third of the points will be deducted.
- Good coding style will be honored by bonus points. This specifically includes well-commented and easy-to-read programs.
- I will try to fix obvious blocking bugs in the program in order to see whether the remainder of the code is any good. However, I will not try excessively to understand tricky parts of your program in order to fix bugs.[6]

[4] Submit only the statements for creating the required tables.

[5] In a case where some modifications might be needed, please remember to also submit a readme text alongside your solutions.

[6] The chance of me fixing bugs is of course increased greatly by good commenting and coding style.