

# Project Details

## Database Systems Course (2014-2015) – Final Project

YAGO<sup>1</sup> is a huge semantic knowledge base, derived from Wikipedia, WordNet, and GeoNames. YAGO contains almost 10 million entities (e.g. persons, organizations, cities), and 120 million facts about them.

Your goal is developing an *interesting* application based on the data in YAGO. Following this year's theme, your application should contain data about geographical locations, countries, cities, languages etc. Your application can be anything, but must include some searching for data, manual update of data, and a "full update" process to retrieve information from YAGO files.

More specifically:

- Design and create an application, with an underlying database schema for storing / querying / updating information from YAGO.
- Populate your database with at least 150K records (table rows). You should support massive import from YAGO by processing files from <http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/>  
Note: do not be afraid on importing (much) more records if the information is available, but on the other hand do not import the entire YAGO into your database.
- The application must support some manual updates of the tables that contain data from YAGO, including editing, deleting and/or adding data. Then, you need to think what to do with these updates when an import is performed.

### **Bureaucracy:**

- Work should be done in groups of 4 students. All the groups should be approved by email, teams smaller than 4 will only be approved if no students are left without a team.
- Send the TA an email by **May 5th** with the details (students names and ids) of your group (one mail per group)

### **The program**

- Your programs will be in **Java**
- For the UI, we overview SWT in class, but other packages such as Swing, AWT, Qt, etc. can be used.
- DO NOT use ORM packages such as hibernate.
- Other than that, you can use any existing Java package as long as it is freely available, and does not require special registrations.
- The program must be *portable*: installation steps should be (i) copy, (ii) paste, (iii) create the DB schema using an SQL script or connect to the school MySQL server, (iv) edit configuration and (v) run!
- Test your program on different environments, at least Linux (as in the school labs) and Windows.

---

<sup>1</sup> Fabian M. Suchanek, Gjergji Kasneci and Gerhard Weikum, **YAGO - A Core of Semantic Knowledge**, WWW '07

## **DB design**

According to the principles taught in class, and in particular:

- Give meaningful names to tables, fields, indexes, keys, etc.
- Use keys and foreign keys. They must be integers in MySQL.
- Use indexes where needed to optimize your queries.
- Avoid redundant data as much as possible (e.g., by the decomposition principles of lesson 9).

## **The code**

- Make sure to handle errors and avoid crashes
- Make sure to close resources (ResultSets, Statements...)
- Your code should contain 4 low-level packages:
  - o *appname.parsing* – including all the code that parses the YAGO files
  - o *appname.db* – including all the DB calls, connection pooling, etc.
  - o *appname.core* – including all the application core logic, thread pooling, etc.
  - o *appname.ui* – including the UI code.
- Add sub-packages as needed (e.g., *appname.core.exceptions*).
- Allow defining properties through the UI, or through a configuration file (e.g., the DB connection)
- The code is also examined, so keep it readable...

## **Documentation**

Very important! If the documentation is good, a lot of the grader's work will be based on it, which is in your favor...

- User manual:
  - o What does your application do? – an overview
  - o Installation guidelines, including how to adjust the swt.jar to the environment if needed
  - o Administrator username and password (if exist)
  - o The screens of the applications, how to get to them and what are their features
  - o How to run the YAGO import, and how to provide the update files
- Software documentation
  - o DB structure
  - o Code structure
  - o Which data from YAGO you used (and other sources, if you used any)
  - o External packages that you used
  - o General flow of the application
- Optional: javadocs, wiki, a presentation...

## **Submission and Grade**

In the course website, you can find the project grading guide and examination form.

Submission date is the **on the course website**.

The submission should include:

- Your DB schema fully populated on TAU's MySQL server
- "Executable" Directory containing:
  - o Your final application – JAR file
  - o Any other required JAR files (for example JDBC, SWT...)
  - o "runme.bat" for Windows and "runme" for Unix, containing the command for executing your program (for example, "java -jar myapp.jar myparameter")
  - o User manual

- “Source” Directory containing:
  - o All your java sources
  - o An SQL script for creating your DB (without the data)
  - o Software documentation
- “Members” Directory containing:
  - o A file with all the group member FULL NAMES & IDS
  - o Your MySQL user & password at TAU’s server

Please submit the project HARDCOPY to Yael’s mailbox – **no. 376, on Schreiber’s floor no. 2:**

- Print the “User manual”, Software documentation and the files in the “Members” directory.  
**You DO NOT need to print the source files.**
- Burn all the files to a CD (write the team number and application name on it)
- Submit them together in a folder, an envelope etc.

As a backup, submit a zip file with the user manual, software documentation and your code to moodle.

#### **Bonus**

- Originality in design, features, “thinking out of the box”
- Interesting algorithms implemented
- Exceptionally convenient and aesthetic GUI

**Good luck!**