INFS3200/7907 Advanced Database System – Practical 2

Learning objectives:

- Date integration from different sources in different formats.
- Understand Edit Distance for textual similarity measurement.

Five source data sets:

1. Athlete.txt - it contains the detail of all the athletes, such as their name, date of Birth, country they are serving, etc.
2. Country.html - it contains *code* and *name* of all countries in the world that ever participated the Olympic Games in history.
3. Results.xlxs - it provides athletes' results in Olympic Games, including information such as Medal or score.
4. Sports.accdb – it contains information of sports and events. One sport may have many events. For example, sport "Swimming" contains events such as "Men's 50m Freestyle", "Women's 50m Freestyle".
5. City.csv - it provides information (such as name and country) of the city which hosted Olympic Games in history.

Analyze each individual data source and identify a list of issues, both in schema level and data level, which need to be resolved for data integration.

Schema Level

The schema level issues in data integration are common in practice. One example is that relations from different data sources may use different names for the same attribute. In the given five source data sets, country.html and city.csv both have an attribute for showing country names of records; in the country.html, the attribute is named as "Country Name" and in the city.csv, the corresponding attribute is named as "Region". Another example is that the ages of athletes may be represented as *data of birth* or *age* at a given year. For data integration purpose, you need identify these attributes and use consistent attribute names after integration.

Task 1: Design an integrated schema in Oracle XE and systematically import the respective data sources resolving schema level issues

Data Level

The data level issues also exist in data integration. It is common that the name of a person/city/country is not presented identically in different data sources due to reasons such as typos and/or applying different input customs. For example, the first name of athlete "Eunk-Yunk Chang"

may be presented as "Eunk-Junk". To resolve this problem, *edit distance* (ED) is widely used for measuring textual similarity.

> *Edit Distance - Given two character strings S1 and S2, the edit distance between them is the minimum number of edit operations required to transform S1 into S2. Most commonly, the edit operations allowed for this purpose are: (i) insert a character into a string; (ii) delete a character from a string and (iii) replace a character of a string by another character; for these operations, edit distance is sometimes known as Levenshtein distance. For example, the edit distance between "Eunk-Yunk" and "Eunk-Junk" is 1.*

Generally, if two non-identical names have small edit distance, it indicates these two names are similar, and it implies they are likely to refer to the same person/city/country. A python program (ed.py) for computing edit distance is provided. You are asked to use it to do the following:

Task 2: For a given name "Eunk-Junk", find all athletes from file athlete.txt whose first names are similar and report your findings in a .txt file. If the edit distance is less than a threshold, they are similar. The default threshold is 2.

Optional: Set different thresholds in ed.py to test different names and observe the difference in the similarity findings.

(Note: If you use laptop rather than the computers in the computer Lab, you may need download and install IDLE 2.7.3 from http://www.python.org/download/releases/2.7.3/ on your system in order to run ed.py)