Information Integration – Exercise 2 – Gabriel Glaser

Task 1: Heterogeneity

a)
- *Syntactic* (Hardware, Software, Interface): No, only one source.
- *Structural* (Data model, schematic): No, only one source
- *Semantic*:
  - Name conflict: synonym labels “year” and “YEAR”.
  - Identity: Some of the first results likely the same (, but couldn’t be differentiated, because different and incomplete representations)
  - Value conflict: Track titles: *You know you’re right* vs. *You Know You’re Right* among first Nirvana results.

b) Extensions to scenario to satisfy missing heterogeneity types:
- *Syntactic*: Consider a CD data source from America which likely has a longer response time. (different hardware)
- *Structural*: Consider a CD data source which returns an XML file. (different data model)

Task 3: Distributed DBMS

a) Distribution:
- *Physical*: Servers are located in different physical/geographical locations. Therefore, they don’t share hardware components (except network).
- *Logical*: Result of application requirements, e.g., store data on different servers to handle network failure or implement caching for more speed.

b) Contained autonomy types:
- *Design*: Choose to store DB in 3NF (3^{rd} normal form) or as a “BigTable” with less normalization.
- *Communication*: Decision to communicate with SQL and web form (*decision on specific query languages*). Also, they decide to allow write access to older table but not on newer table (*decision on what query capabilities to support*).

c) Execution autonomy is the last type of autonomy. For instance, a system could block queries from some countries.

d) Syntactic heterogeneity:

- *Hardware*: This type of heterogeneity can be noticed when some datasources process a query faster than others (different CPU/bandwidth).
- *Software*: Different datasources are stored using different operation systems, e.g., need to use different file separators ("/" for Linux, "\" for Windows).
- *Interface*: For instance, access via URL containing various parameters vs. access via SQL query given to an URL as string.

e) a) Value vs. relation heterogeneity:

Employee(ID, Firstname, Lastname, isManager, department, gender)

vs.

MaleEmployee(ID, Firstname, Lastname, isManager, department)
FemaleEmployee(ID, Firstname, Lastname, isManager, department)

b) Value vs. attribute heterogeneity:

Employee(ID, Firstname, Lastname, isManager, department, gender)

vs.

Employee(ID, Firstname, Lastname, isManager, department, isFemale, isMale)

c) Different labels of attributes (in comparison with the original):

Employee(ID, Firstname, Lastname, isManager, *workArea*)

d) Normalized vs non-normalized schema (original is normalized):

Employee(ID, Firstname, Lastname, department)
Managers(MID, ID → Employee)

Task 4: Integrating publication data

a)

b)

c)