**Unstructured and Streaming Data Engineering 2020-21 15.01.2021**

|  |  |
| --- | --- |
| **Surname:** | **I** |
| **Name:** |
| **Codice persona:** |

Exercise 1. Design (10 points)

Suppose you need to design the data model that supports the structure and organization of a university campus. The campus comprises several buildings. Each building has an address, a map, some photos, an opening schedule. Every building is either: a Department (office building for faculty), a service building (i.e., Dean’s office, administration, general services, and so on), a teaching building (including classrooms). A building has several floors that, in turn, have several rooms. Each building, each floor, and each room has a unique code, and a type (classroom, restroom, hall, ... ). Each room includes a set of facilities, a set of furniture items, the number of available seats, and the access level. The possible access levels are: public, restricted to all employees, and restricted only to one or more specific employees. In the latter case, the room is associated with the respective employees (one or more). Courses have a title, code, description, and associated main instructor and teaching team, and a list of subscribed students. Courses have also associated websites, materials, and exams. Access to buildings and offices are registered electronically. Moreover, a building automation solution monitors the buildings by observing the temperature, the electricity consumption, and the presence of people.

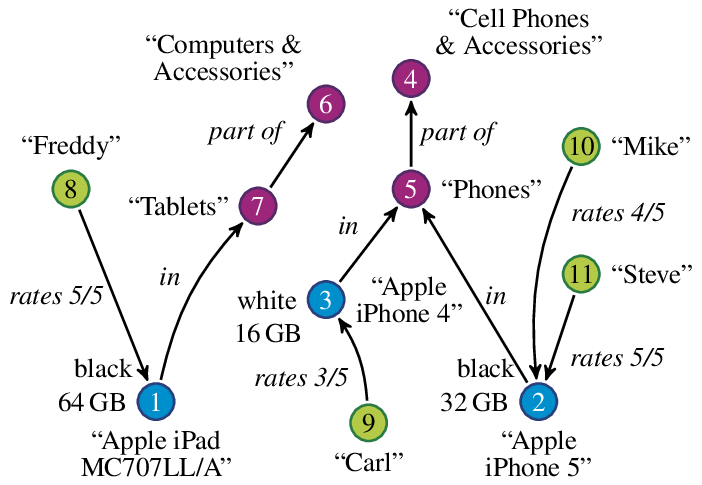
Describe a high-level model of the data focusing on which parts of the data you would implement using relational, non-relational, and streaming solutions. **Motivate your choice**, and for each piece of the solution, represent a schema (if relational) or exemplify the data model (if not relational).

Exercise 2. Graph DB (5 points)

Consider the following **excerpt** of a graph model, possibly implemented in Neo4J, describing reviews of products written by users. Colours of nodes represent node labels as follows:

* Green = User (with attribute: UserName)
* Blue = Product (with attributes: Model, Color, Memory)
* Purple = Category (with attribute: CategoryName)

A product is “in” a category; and a category may be “part of”a supercategory, **with maximum 3 levels** of categorization. Users review products assigning a rate.



Write the following queries:

1. Find the top category of products with more than 32GB of RAM.
2. Find the list of lowest-level categories of products reviewed by users that always rate products only with rate equal to 5.

Exercise 3. Documental DB (5 points)

Suppose you want to record in a document-based storage solution a list of scientific articles. Each article is described by a title, a list of authors with respective affiliation and email, the number of pages, an abstract and the actual content. Content includes paragraphs, in turn containing text, table and images. Paragraphs can also recursively contain subparagraphs, with the same structure. An article includes also references to other articles (bibliography). As an example:

{\_id: doc1,

Title: "X",

Authors: [{name="A", affiliation="Polimi"},

{name="B", affiliation="KTH"},

…],

Pages:45,

Abstract: "…",

Content: [{section: "S1", title: "T1", text: "…"},

{section: "S2", title: "T2", text: "…", imgs: […] },

…],

Bibliography: [{paper: doc2},{paper: doc3},…]

}

Write the following queries:

1. Find the documents written by at least one author from “PoliMi” that contains at least 20 pages and includes a section titled “Big Data” at any of the 3 top-most section levels
2. Find the documents that include a reference to another article written by at least one author from “UniMi”

Exercise 4. Streaming (8 points)

Suppose you want to monitor with a stream processing engine an Industry 4.0 aluminum extrusion plant in which aluminum billets are:

1. preheated before extrusion in an oven,
2. transferred to the extrusion press by a robotic arm, and
3. extruded by a press.

All the pieces of equipment (the oven, the robotic arm, and the press) send events reporting their status. The oven’s statuses are: empty, loaded, and billet preheated. The robotic harm’s statuses are: ready, billet grasped, and billet loaded in the press. The press’s statuses are: ready and extruding. Moreover, all the pieces of equipment are instrumented with IoT sensors that measure the temperature of the billet every second.

In a continuous query language of your choice among EPL, flux, KSQL, and Spark Structured Streaming propose:

* how to model the streaming data generated by the pieces of equipment and by the IoT sensors, and
* a continuous query that raises an alarm if 1) the robotic harm takes more than 20 seconds to transfer a preheated billet from the oven to the press (notably, the harm cannot load the billet in the press, unless the press is ready), and 2) the difference in temperature between the billet in the oven and in the press is larger than 5 C°.

Exercise 5. EPL (3 points)

Suppose you receive the following stream of events: A1@0,C1@1,B1@2,B2@3,A2@4,B3@5,A3@6,B4@10.

Note that A3@6 denotes an event of type A identified by the number 3 that is received at time 6.

Given the patter: every A -> (B and not C where timer:within(3 sec))

When does such a pattern match? Which are the events that trigger the matching? Why?