**Unstructured and Streaming Data Engineering 2020-21 22.06.2021**

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

Exercise 1. Graph DB (8)

A dedicated online review and social networking system tracks the activities of customers of hotels, including their stays, their searches and their online interactions, friendships, and activities. Hotels are described by their services, rooms, owner, and employees. Customers can write reviews about each of these aspects. Suppose you want to create a graph database for this.

1) Motivate if and how a graph DB could help the organization (2)

2) Represent a sketch of the data model that can be used. Show it by sketching a model including examples of all the relevant concepts and relations. (2)

3) Write a command that adds a new review to the database, and connects it to the respective hotel and author. (2)

4) Write a Cypher query that extracts the reviews of hotels whose owner is the same as the author of the review. (2)

Exercise 2. Documental DB (6 points)

Suppose you want to re-design the problem of exercise 1 as a documental database implementation.

1) How would you structure the document(s)? (2)

2) Which solution is more effective between the document DB and the graph DB, in case we want to explore the connections between customers, reviews, and hotels? Why? (2)

3) Write a graph query using the MongoDB API that extracts hotels that include a swimming pool among their services (2)

4) Write a command that adds a new hotel in the database. (2)

Exercise 3. Crowd and Quality (4 points)

With respect to the problem in exercise 1:

1. propose some strategies to increase the probability of customers to write reviews. (2)
2. describe some possible problems and risks that the collected data might have. (2)

Exercise 4. Streaming (10 points)

Suppose you want to monitor with a stream processing engine a fleet of autonomous vehicles that pick and drop people and goods in a city.

Each vehicle sends events reporting origin, destination and its status: available to pick people, pinking people, moving, dropping people, and recharging. A sensor in each vehicle reports the number of people in it. If the people in the vehicle are three or less, the vehicle can pick up more people (up to 4)

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 autonomous vehicles (1)
* A continuous query that emits the sum of the number of people the fleet is currently moving. (1)
* A continuous query that emits the average number of people per vehicle grouped by destination (2)
* A continuous query that returns every vehicles that: (4)
  + in less than 5 minutes
  + picked enough people to fill up all the 4 spaces, and
  + dropped them all.
* A continuous query that monitors the results of the previous one and counts how many vehicles were detected in the stream over a tumbling window of 5 min (2).

Explain your choices in detail.

Exercise 5. EPL (4 points)

Suppose you receive the following stream of events: A1@0,C1@1,B1@2,B2@3,A2@4,A3@5,B3@6,A4@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 continuous query:

select a.n, b.n

from pattern [

every a=A -> (b=B where timer:within(2 sec) )

];

Which are the events that trigger the matching? (0.5) Why? (1.5)

Which are the events that do not trigger the matching? (0.5) Why? (1.5)