IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2019-2020

MSc in Artifical Intelligence
MSc in Computing Science (Specialist)
MSc in Advanced Computing
MEng Honours Degrees in Computing Part IV
MEng Honours Degree in Mathematics and Computer Science Part IV
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER C410

SCALABLE SYSTEMS AND DATA

Wednesday 11th December 2019, 14:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

1 Write queries for graph databases and document stores in the following two parts.

a A simple Neo4J graph database stores information about orders, employees handling them, products involved, suppliers producing them and so on. Each order involves a product (which has suppliers and categories), a customer who ordered it, a shipper who ships it and employees who handle an order (which are assigned to a territory which is part of a region). The data model is shown in Figure 1.

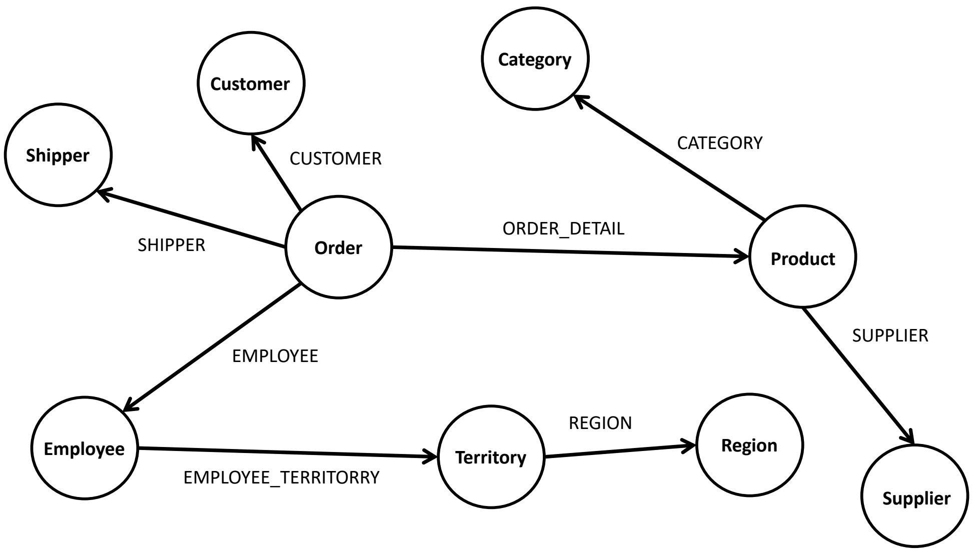


Fig. 1: Data model for orders.

Given this description, write Cypher queries for the following:

i) Assuming Employee has a property called 'name', count all orders handled by "Joe Ghosh".

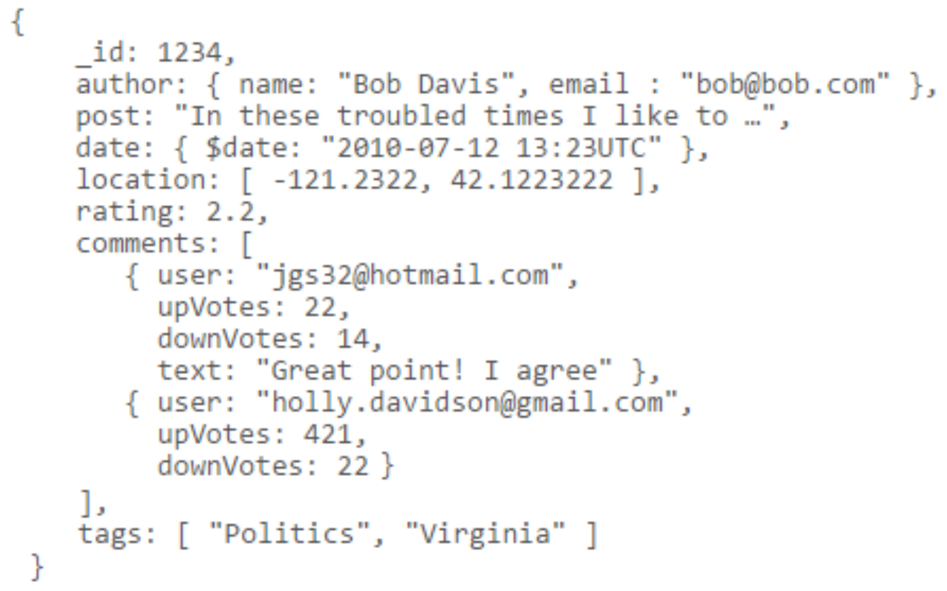
ii) Assuming the Region has a property 'name' and the Order has a property 'ID', return a list of all the regions associated with the product with the ID "123".

iii) Assuming Customer as well as Shipper have a property 'name' and Order has a property 'ID', write the most compact query returning the names of Shipper's and Customer's involved with the Order with ID "123".

iv) Assuming Shipper and Supplier have a property 'name', find how many products each Shipper has shipped and list all name of suppliers for these products.

v) Assuming Customer, Employee and Supplier have a property ’name’, return the names of all Customers who work with an Employee named ’Mark’ but who have never ordered product from a Supplier by the name of ’Jeff’.

b You are given the structure of the *author* collection, containing information about authors, their posts as well as comments, ratings etc., as shown in Figure 2.



```
{
    _id: 1234,
    author: { name: "Bob Davis", email : "bob@bob.com" },
    post: "In these troubled times I like to …",
    date: { $date: "2010-07-12 13:23UTC" },
    location: [ -121.2322, 42.1223222 ],
    rating: 2.2,
    comments: [
       { user: "jgs32@hotmail.com",
         upVotes: 22,
         downVotes: 14,
         text: "Great point! I agree" },
       { user: "holly.davidson@gmail.com",
         upVotes: 421,
         downVotes: 22 }
    ],
    tags: [ "Politics", "Virginia" ]
 }
```

Fig. 2: Example of the authors collection in MongoDB.

i) Write a query to display all the authors in the collection *author*.

ii) Write a query to find all authors with a rating of 2.0.

iii) Write a query to find all authors living within a distance of 100 meters of the point with coordinates 45, 47.

iv) Write a query to find all authors of whom a post has more than 50 upvotes and less than 20 downvotes.

v) Find all authors of whom a post has a comment which contains the word 'point' and which has 'History' as one of the tags.

The two parts carry equal marks.

2 a Describe the methods to optimize data structures/index for disk. Equally describe optimizations for data structures in main memory.

b Describe the different methods as to how joins can be carried out with MongoDB.

c Name the advantages of DNA storage compared to other forms of storage. Highlight its advantages over glass storage.

d What is the purpose of the flash translation layer? What functions does it provide?

The four parts carry equal marks.

Section B (Use a separate answer book for this section.)

3 a Briefly explain each of the following concepts.

i) CAP theorem

ii) Resource disaggregation

iii) Hinted handoff

iv) Hybrid cloud

b A data-parallel dataflow model, as provided by *Spark*, is a popular way to process large datasets on a cluster of machines. The Spark API includes the following calls for data processing:

collect() : RDD[T] ⇒ Seq[T] *count*() : RDD[T] ⇒ Long
filter(f : T ⇒ Bool) : RDD[T] ⇒ RDD[T]
flatMap(f : T ⇒ Seq[U]) : RDD[T] ⇒ RDD[U]
groupByKey() : RDD[(K, V)] ⇒ RDD[(K, Seq[V])]
map(f : T ⇒ U) : RDD[T] ⇒ RDD[U]
reduce(f : (T,T) ⇒ T) : RDD[T] ⇒ T
reduceByKey(f : (V,V) ⇒ V) : RDD[(K, V)] ⇒ RDD[(K, V)]
save(path : String) : RDD[T] ⇒ ()

i) Briefly describe the **two types** of API calls that are supported by Spark and justify why both types are necessary.

ii) Classify the above APIs calls according to the two types of calls, which you identified under i), and briefly justify your classification.

iii) Using the Spark API calls above, write a job that takes a set of 100 words as input and then counts how many times each word occurs in a 1 TB dataset.

(The details of the Scala syntax used to write the job are not important. Use any consistent programming language syntax to describe the Spark job.)

iv) Describe how you would extend the design of Spark in order to obtain the words to be counted from a micro-service with a key/value store interface. The words stored by the key/value service are dynamically updated.

Ensure that your design (i) does not limit *scalability*, (ii) remains *fault-tolerant* and (iii) achieves *consistency* of the produced results.

The two parts carry, respectively, 40% and 60% of the marks.

4 You work as a software architect for ROADZ, a start-up company that wants to offer a service to push dynamically generated, geographically-local traffic alert about accidents and traffic jams to mobile devices. As soon as new information about a traffic alert becomes available from external sources, it should be sent to potentially millions of mobile devices.

The ROADZ system should allow mobile devices to provide *subscriptions* that define the *alert messages* that devices want to receive. Subscriptions include filter expressions to decide if a particular altert is relevant for a given device, e.g., by defining a geographic area. Subscriptions change over time, e.g. as devices move between locations. Alert messages contain attributes such as the type of alert and the specification of geographic relevance that can be matched against the filters in subscriptions.

The design of the ROADZ system should be (i) incrementally-scalable, (ii) fault-tolerant and (iii) maintain low latency (on the order of seconds) when pushing alert messages to mobile devices.

(You should make justified decisions about any other requirements that are left unspecified.)

a Draw a high-level diagram of the *system design*, clearly labelling all distributed components. Explain the operation of each component, and justify its function.

b Explain how your design is *scalable* in that it can handle an increasing number of concurrent subscriptions and content messages.

c Explain how your design is *fault-tolerant* in that it can handle the failure of individual machines in the data centre.

The three parts carry, respectively, 35% 35%, and 30% of the marks.