CHAPTER 10

Big Data

Solutions for the Practice Exercises of Chapter 10

Practice Exercises

10.1

Answer:
The key-value store, since the distributed file system is designed to store a moderate number of large files. With each file block being multiple megabytes, kilobyte-sized files would result in a lot of wasted space in each block and poor storage performance.

10.2

Answer:
We would store the student data as a JSON object, with the takes tuples for the student stored as a JSON array of objects, each object corresponding to a single takes tuple. Give example ...

10.3

Answer:
Create a key by concatenating the customer ID and date (with date represented in the form year/month/date, e.g., 2018/02/28) and store the records indexed on this key. Now the required records can be retrieved by a range query.

10.4

Answer:
With the `map` function, output records from both the input relations, using the join attribute value as the `reduce` key. The `reduce` function gets records from both relations with matching join attribute values and outputs all matching pairs.

10.5

Answer:
The problem with the code is that the collect() function gathers the RDD data at a single node, and the map and reduce functions are then executed on that single node, not in parallel as intended.

10.6

Answer:

a. RDDs are stored partitioned across multiple nodes. Each of the transformation operations on an RDD are executed in parallel on multiple nodes.

b. Transformations are not executed immediately but postponed until the result is required for functions such as collect( ) or saveAsTextFile( ).

c. The operations are organized into a tree, and query optimization can be applied to the tree to speed up computation. Also, answers can be pipelined from one operation to another, without being written to disk, to reduce time overheads of disk storage.

10.7

Answer:
FILL IN ANSWER (available with SS)

10.8

Answer:
Divide by 3600, and take floor, group by that. To output the timestamp of the window end, add 1 to hour and multiply by 3600

10.9

Answer:
Each relation corresponding to an entity (student, instructor, course, and section) would be modeled as a node. *Takes* and *teaches* would be modeled as edges. There is a further edge between *course* and *section*, which has been merged into the *section* relation and cannot be captured with the above schema. It can be modeled if we create a separate relation that links sections to courses.