Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Big Data
Fall Semester 2017
Lecturer(s): Ghislain Fourny
Assistant(s): Alexandr Nigay, Damien Desfontaines, Ingo Müller, Konstantin Taranov, Marco Ancona, Renato Marroquín

ETH Zurich
Date: January 25, 2018
Last update: January 19, 2018

Exam

Questions

Rules (please read carefully)

- You have 150 minutes for the exam.
- Write the answers on the stack of papers with the title "answer sheets".
- Write your name and Legi number on the cover page of the answer sheets.
- Please write your Legi number on all other pages of the answer sheets.
- Use blue or black ink, DO NOT USE red ink. DO NOT USE pencils.
- Write as clearly as possible and cross out everything that you do not consider to be part of your solution. Answers can be given in either English or German.
- Hand in your answer sheet. Do not hand in the sheets with the questions. You may keep the question sheets.

Remarks

- The multiple choice questions only have a single correct answer. If you cross more than one box of a question, you will get zero points for that question.
- There are no negative points.
- In a multiple choice question, if you picked an answer and then would like to change it, **make it very clear**. For example, you could cross out the old answer, tick the new answer, and draw a circle around your new answer.

A Block storages (19 points)

(1) (4 points) Which of the following statements about HDFS are true?

a. HDFS was designed for storing very large files.
b. HDFS was designed to be deployed on commodity hardware.
c. HDFS ensures high performance at storing lots of small files.
d. HDFS exposes a flat storage file system.
e. HDFS is optimized for concurrent file modification.
f. HDFS is optimized for random file accesses.
g. HDFS was designed for low-latency data access.
h. HDFS is optimized for batch processing.

(2) (3½ points) Which of the following statements about HDFS fault tolerance are correct? On the answer sheet, mark the correct answers with a checkmark (✓) .

a. Checksum metadata is used to detect data corruption because of disk faults.
b. Generation stamp metadata prevents reading stale data.
c. The checksum metadata is calculated per file rather than per each block.
d. Each block's metadata contains generation stamp.
e. The checksums are sent to the client when it reads files.
f. The checksums are only checked on the server when a client attempts to read data.
g. During normal operation DataNodes periodically send heartbeats to the NameNode to confirm thier liveliness.

(3) (2 points) Which of the following statements about writes to HDFS are correct? On the answer sheet, mark the correct answers with a checkmark (✓) .

a. Client chooses to which data node to write and then signals the namenode about its choice.
b. Client writes packets to the first datanode which in turn forwards them to the second datanode in the pipeline fashion.
c. If a failure of the first data node happens, all partial blocks are discarded and the client writes data from scratch again.
d. If a client wrote all packets successfully, but failed before signaling the namenode to close the file, then the file will be closed after predefined timeout and the written data will not be erased.

(4) (2½ points) Which of the following statements about reads from HDFS are correct? On the answer sheet, mark the correct answers with a checkmark (✓) .

a. To read a file, a client sends a request to the namenode before accessing datanodes.
b. Upon a client's read request, the namenode returns the addresses of the datanodes that have a copy of each block of the requested file sorted according to their proximity to the client.
c. Clients have exclusive access on reading a file.
d. A client has to signal the namenode after completing reading a file.
e. A client verifies checksums for the data transferred to it from the datanode.

(5) (2 points) For each of the following HDFS components, say if the information is stored persistently on the NameNode or on a DataNode. On the answer sheet, mark the correct answers with a checkmark (✓) .

a. The block's metadata. A. NameNode B. DataNode

b. The list of blocks belonging to each file. A. NameNode B. DataNode

c. The files containing the actual blocks of data. A. NameNode B. DataNode

d. File namespace. A. NameNode B. DataNode

(6) (5 points) What are benefits of storing objects as blocks against storing them as single files? Mention at least 3 benefits.

B Object storage and key-value stores (17 points)

(1) (6 points) Mark each of the following statements about Azure Blob Storage as either true or false. On the answer sheet, mark your answers with a checkmark (✓).

a. A page blob can have a variable length.

b. Multimedia data files are usually stored as append blobs to access them sequentially.

c. Page blobs are usually used for non-sequential read and write accesses.

d. Block blobs are used when sequential access is required.

e. All block types expose block IDs to the user.

f. Each block in an append blob has to be of the same fixed size.

g. A user must specify a write offset when adding a block to an append blob.

h. A page blob is composed of pages of a fixed size.

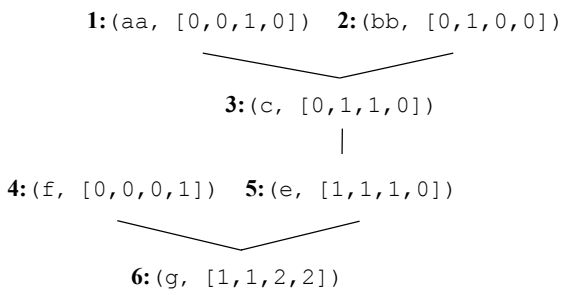
i. An object ID is comprised of the bucket ID and the object name.

j. An object can be stored using blocks or pages.

k. A storage stamp uses two different types of replication: Intra-stamp and inter-stamp.

l. Vector clocks are used to define the latest available version of an object.

(2) (3 points) For each of the following Amazon Dynamo vector clocks, say whether they have been computed correctly (true) from their parents or not (false).



a. Vector clock 1

b. Vector clock 2

c. Vector clock 3

d. Vector clock 4

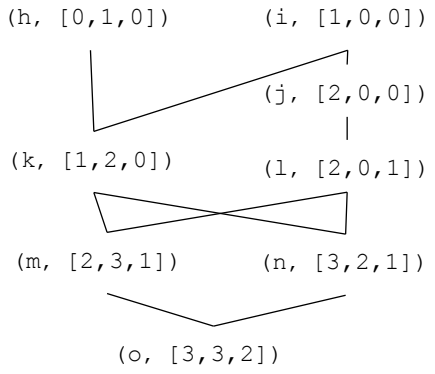
e. Vector clock 5

f. Vector clock 6

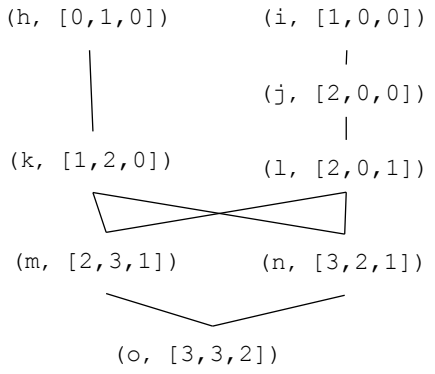
(3) (4 points) Given the following sequence of vector clocks:

- (h, [0,1,0])
- (i, [1,0,0])
- (j, [2,0,0])
- (k, [1,2,0])
- (l, [2,0,1])
- (m, [2,3,1])
- (n, [3,2,1])
- (o, [3,3,2])

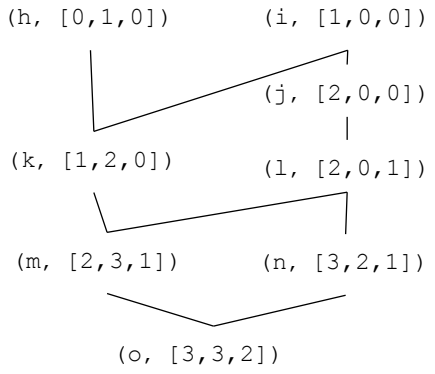
Choose the correct DAG representation, and pick the chosen letter on the answer sheet.



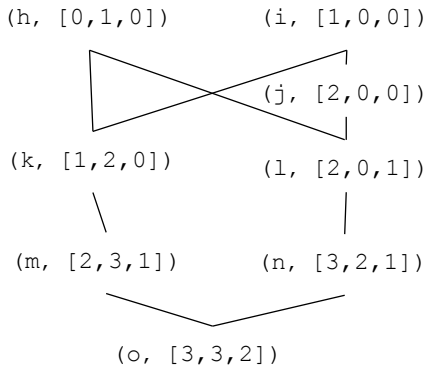
(A)



(B)



(C)



(D)

(4) (2 points) Having six servers (i_1 to i_6) with the following number of virtual nodes/tokens assigned to each: 2, 4, 32, 64, 16, and 1. How much memory each physical server allocated for virtual nodes if each virtual node has 16GB of main memory assigned?

A. 52, 32, 512, 1024, 16, 16
B. 32, 64, 512, 1024, 256, 16
C. 32, 64, 1024, 512, 256, 96
D. 42, 84, 572, 256, 1024, 16
E. 22, 94, 512, 1044, 256, 16

(5) (2 points) Mark each of the following statements about Merkle trees as either true or false. On the answer sheet, mark your answers with a checkmark (✓).

a. Merkle trees are binary trees in which not all nodes are labelled.
b. Merkle trees are binary trees in which every non-leaf node is labelled with cryptographic hash of the labels of its child nodes.
c. Key-value stores use Merkle trees only for replication.
d. For detecting inconsistencies in data, the Merkle tree root hash is exchanged first between nodes, and then, every child hash is exchanged as well.

C HBase (13 points)

(1) (1 point) Choose the most precise ending to the phrase: "HBase is a . . . "
   A. object store
   B. wide column store
   C. relational database
   D. document store

(2) (1 point) Which data shape suits HBase's data model the most?
   A. tree
   B. graph
   C. cube
   D. table
   E. text

(3) (1 point) If HFiles in HBase are written once and never edited after that, then how is deleting a cell even possible in HBase?
   A. It is not possible; cells written into HBase stay there forever, and the user application has to deal with this.
   B. Deletion is possible only on compaction, when multiple HFiles are combined into a single one.
   C. Deletion is possible through overwriting of the existing cell value with a special "NULL" value.
   D. Deletion is possible through insertion of a special marker, which marks the cell as deleted.

(4) (1 point) Which of the following is the best-fitting use case for HBase?
   A. Nothing is known about the schema in advance, and the data will not fit on a single machine.
   B. The schema is fixed on a high level, but some details may change on the fly; the individual data items exhibit highly irregular relations between each other.
   C. The data is tabular; it will not fit on a single machine; expected access patterns consist mostly of full table scans and lookups by key.
   D. The data will not fit on a single machine; no schema is defined at all, so the data items will have to be stored as-is.

(5) (9 points) On the page after this introductory text you will see a diagram which presents a simplified representation of the internal state of an HBase RegionServer (a similar diagram was used in the exercise sheet). In this simplified representation, each HBase key-value pair consists of the following fields, in order: **row key, column qualifier, timestamp, value**. Column family here is omitted, because we assume that there is only one column family in that table. This HBase table is called **phrases**. These simplifications are the same as were used in the exercise sheet.

   What you have to do: on the answer sheet you will see multiple queries in HBase shell syntax: `get <table name>, <row key>`. In provided spaces, write down the data that will be returned by each query. **Assume that the system is configured to return only the latest version of each cell.**

   On the next page you can find an example of the solution.

Note the following about the solution format:

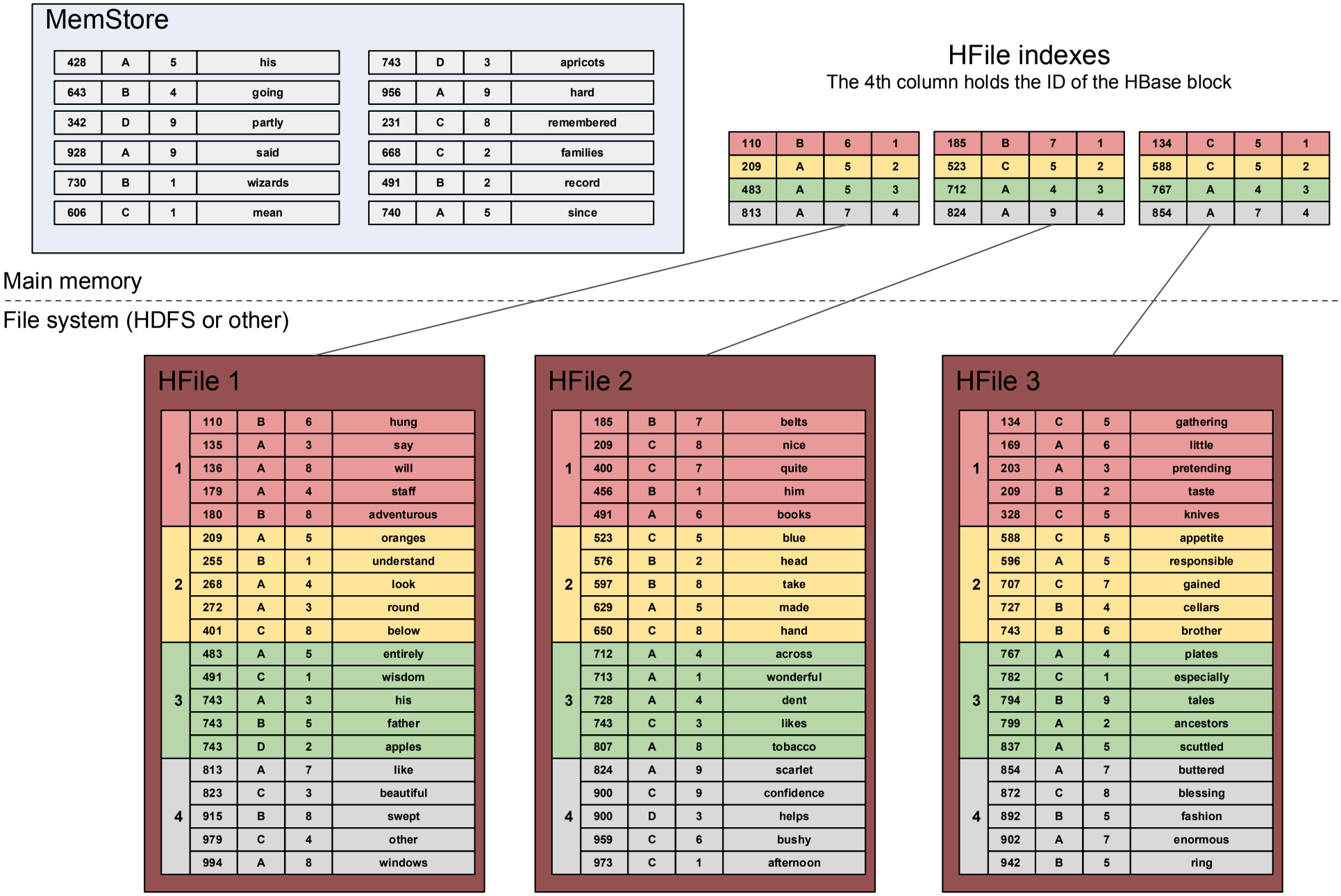
- Each answer table has 5 lines in it. It may happen that you need fewer lines than that, but if you find yourself needing more than 5, you are doing something wrong.
- The order in which you write lines into the answer table does *not* matter.
- If you make a mistake, you can just cross out the wrong line in the answer table.

Query: `get 'phrases', '209'`			
Row key	**Column**	**Timestamp**	**Value**
209	A	5	oranges
209	B	2	taste
209	C	8	nice

On the answer sheet, you need to fill in similar tables for the following queries (they are repeated there):

- `get 'phrases', '491'`
- `get 'phrases', '900'`
- `get 'phrases', '743'`

The diagram is on the next page.


MemStore
428 A 5 his
643 B 4 going
342 D 9 partly
928 A 9 said
730 B 1 wizards
606 C 1 mean
743 D 3 apricots
956 A 9 hard
231 C 8 remembered
668 C 2 families
491 B 2 record
740 A 5 since
HFile indexes
The 4th column holds the ID of the HBase block
110 B 6 1
209 A 5 2
483 A 5 3
813 A 7 4
185 B 7 1
523 C 5 2
712 A 4 3
824 A 9 4
134 C 5 1
588 C 5 2
767 A 4 3
854 A 7 4
Main memory
File system (HDFS or other)
HFile 1
1 110 B 6 hung
135 A 3 say
136 A 8 will
179 A 4 staff
180 B 8 adventurous
2 209 A 5 oranges
255 B 1 understand
268 A 4 look
272 A 3 round
401 C 8 below
3 483 A 5 entirely
491 C 1 wisdom
743 A 3 his
743 B 5 father
743 D 2 apples
4 813 A 7 like
823 C 3 beautiful
915 B 8 swept
979 C 4 other
994 A 8 windows
HFile 2
1 185 B 7 belts
209 C 8 nice
400 C 7 quite
456 B 1 him
491 A 6 books
2 523 C 5 blue
576 B 2 head
597 B 8 take
629 A 5 made
650 C 8 hand
3 712 A 4 across
713 A 1 wonderful
728 A 4 dent
743 C 3 likes
807 A 8 tobacco
4 824 A 9 scarlet
900 C 9 confidence
900 D 3 helps
959 C 6 bushy
973 C 1 afternoon
HFile 3
1 134 C 5 gathering
169 A 6 little
203 A 3 pretending
209 B 2 taste
328 C 5 knives
2 588 C 5 appetite
596 A 5 responsible
707 C 7 gained
727 B 4 cellars
743 B 6 brother
3 767 A 4 plates
782 C 1 especially
794 B 9 tales
799 A 2 ancestors
837 A 5 scuttled
4 854 A 7 buttered
872 C 8 blessing
892 B 5 fashion
902 A 7 enormous
942 B 5 ring

D Apache Spark and YARN (11 points)

(1) (2 points) Which of the following statements about Spark are correct? On the answer sheet, mark the correct answers with a checkmark (✓) .

a. Spark achieves fault tolerance by storing for each partition the dependency information needed to recalculate the partition.

b. Spark maintains a log of updates to each RDD to recover the lost data.

c. Spark works only with RDDs of key-value pairs.

d. Actions always return an RDD.

(2) (3 points) For each of the following computation units in Spark, say how many of units A in units B: at most one, may be many, or none? On the answer sheet, mark the correct answers with a checkmark (✓).

a. How many tasks in a stage?	A. at most one	B. may be many	C. none
b. How many actions in a job?	A. at most one	B. may be many	C. none
c. How many transformations in a job?	A. at most one	B. may be many	C. none
d. How many stages in a job?	A. at most one	B. may be many	C. none
e. How many actions in a task?	A. at most one	B. may be many	C. none
f. How many transformations in a task?	A. at most one	B. may be many	C. none

(3) (2 points) Which of the following statements about Spark transformations are correct? On the answer sheet, mark the correct answers with a checkmark (✓) .

a. Transformations are functions that return another RDD.

b. GroupBy transformation always induces a shuffle.

c. FilterByKey transformation can be always computed without a shuffle.

d. Transformations with shuffles are more network heavy than ones without them.

(4) (2 points) What aims was YARN designed to target in comparison with MapReduce v1. On the answer sheet, mark the correct answers with a checkmark (✓) .

a. Scale to 10,000 nodes.	A. Yes	B. No
b. Ability to share cluster with non-MapReduce jobs.	A. Yes	B. No
c. Better resource utilization.	A. Yes	B. No
d. Ability to maintain MapReduce frameworks of different versions.	A. Yes	B. No

(5) a. (½ point) In a typical configuration, how many ResourceManagers are there in a YARN cluster?

A. One per cluster

B. One per node

C. Many per node

D. Many per cluster, but not necessarily on every node

b. (½ point) In a typical configuration, how many ApplicationMasters are there in a YARN cluster?

A. One per cluster

B. One per node

C. Many per node

D. Many per cluster, but not necessarily on every node

c. (½ point) In a typical configuration, how many NodeManagers are there in a YARN cluster?
   A. One per cluster
   B. One per node
   C. Many per node
   D. Many per cluster, but not necessarily on every node

d. (½ point) In a typical configuration, how many Containers are there in a YARN cluster?
   A. One per cluster
   B. One per node
   C. Many per node
   D. Many per cluster, but not necessarily on every node

E Document stores (13 points)

(1) (3 points) Which of the following statements about Document stores and MongoDB are correct? On the answer sheet, mark the correct answers with a checkmark (✓).

a. In MongoDB, operations are atomic at the document level only.
b. Document stores expose only a key-value interface.
c. MongoDB does not support schema validation.
d. MongoDB stores documents in the XML format.
e. MongoDB performance degrades when the number of documents increases.
f. Document stores are column stores with flexible schema.

(2) (2 points) Consider the following collection of JSON documents:

```
{
  _id: "Susan",
  name: "Susan Bookreader",
  age : 23,
  grades : [ "A", "A"]
}
```

```
{
  _id: "James",
  name: "James Smith",
  age : 35,
  grades : [ "B", "C"]
}
```

```
{
  _id: "John",
  name: "John Johnson",
  age : 26,
  grades : [ "A", "B", "C"]
}
```

```
{
  _id: "Robert",
  name: "Robert Williams",
  age : 19,
  grades : [ "B", "A", "C"]
}
```

```
{
  _id: "Michael",
  name: "Michael Brown",
  age : 37,
  grades : [ "A" ]
}
```

```
{
  _id: "Mary",
  name: "Mary Miller",
  age : 29,
  grades : [ "C" ]
}
```

What does the following JSONiq query output? (Write what it would print in a console after execution.)

```
for $person in collection("people")
  where not exists(
    for $grade in $person.grades[]
    where $grade eq "A"
    return $grade
  )
return $person._id
```

(3) (2 points) Consider the dataset from the previous question. What does the following MongoDB query output? (Write what it would print in a console after execution.)

```
people.find({ "grades" : "A" }, {"age" : 1} ).sort( { "age" : 1 } )
```

(4) (2 points) Consider a collection of JSON documents of the following form:

```
{
  _id: "joe",
  name: "Joe Bookreader",
  addresses: [
              {
                street: "123 Fake Street",
                city: "Faketon"
              },
              {
                street: "1 Some Other Street",
                city: "Boston"
              }
            ]
}
```

Which queries will use the following index: `.createIndex({ addresses.street : 1 })`

a. `find({"addresses.city":"Boston"})` A. yes B. no

b. `find({"addresses.street":"Fake Street"})` A. yes B. no

c. `find({"addresses.street":"Street"},{"addresses.city":1})` A. yes B. no

d. `find({"addresses.city":"Boston"},{"addresses.street":1})` A. yes B. no

e. `find({"addresses.name":"Joe"})` A. yes B. no

(5) (4 points) Consider the collection of JSON documents of the following form:

```
{
  _id: "joe",
  name: "Joe Bookreader",
  age : 35,
  zip : 12345
}
```

For which queries the following index is **useless**: `.createIndex({ age : 1, zip : -1 })`

a. `.find({"age":12 })` A. yes B. no

b. `.find({"zip":23456 })` A. yes B. no

c. `.find({"age":{$gt:12}})` A. yes B. no

d. `.find({"age":{$gt:12}, "zip":23456})` A. yes B. no

e. `.find().sort({age:1, zip:1})` A. yes B. no

f. `.find().sort({age:-1, zip:-1})` A. yes B. no

g. `.find().sort({age:-1, zip:1})` A. yes B. no

h. `.find().sort({age:1, zip:-1})` A. yes B. no

i. `.find({"age":12 }).sort({age:1, zip:1})` A. yes B. no

F Well-formedness (12 points)

(1) (4 points) Consider the following document:

```
1. {
2.   'widget': {
3.     "debug": False,
4.     "window": {
5.       "title": `Sample "Konfabulator" Widget`,
6.       "size@width": 42.17,
7.       "size.height": 5.6e48392903754859382932,
8.     },
9.     image: {
10.      "src": "Images/Sun.png",
11.      "name": "sun1",
12.       " alignment": ["vertical": center, "horizontal": right],
13.      "src": "images/sun.png"
14.     },
15.    "text": {
16.      "data": [[["Click Here"]]],
17.      "onMouseUp": "box.things = \_()_/ + box.things;"
18.    }
19.  }
20. }
```

Which of the following lines prevent the document from being well-formed JSON? On the answer sheet, mark with a checkmark (✓) "true" for each line that has an error, and "false" for each line who does not have an error.

a. Line 2
b. Line 3
c. Line 5
d. Line 6
e. Line 7
f. Line 9
g. Line 10
h. Line 11
i. Line 12
j. Line 13
k. Line 16
l. Line 17

(2) (4 points) Consider the following document:

```
1. <?xml version="1.0"?>
2. <Never>
3.   <gonna>
4.     <give you='up'/>
5.     <7et_you_down_>
6.       <br>
```

```
 7.     </7et_you_down_>
 8.     <run-around and="">
 9.       &apos;desert you&quote;
10.     </run-around>
11.     Make you cry! -- Say good-bye! Tell a lie &amp; hurt you
12.     <giiiiive/>
13.   </gonna>
14.   \_()<_/>
15.   <someWeirdNamespace xmlns:xdc=" hmmm weird " xmlwat="http://foo.bar"/>
16.   <XML_is somewhat="\"interesting\", isn't it?" _quotes="\\\\"/>
17. </never>
18. <!-- insert verse here --!>
19. <never>etc., etc.</never>
```

Which of the following lines prevent the document from being namespace-well-formed XML or are otherwise illegal? On the answer sheet, mark with a checkmark "true" for each line that has an error and "false" for each line that does not have an error.
If two lines are indicated, mark "true" if one of the two lines has an error, "false" otherwise.

a. Line 2 & 17
b. Line 4
c. Line 5 & 7
d. Line 6
e. Line 8 & 10
f. Line 9
g. Line 11
h. Line 14
i. Line 15
j. Line 16
k. Line 18
l. Line 19

(3) (4 points) Mark each of the following statements as either true or false. On the answer sheet, mark your answers with a checkmark (✓).

a. JSON documents are in UTF-8, unless specified otherwise.
b. XML documents are in UTF-8, unless specified otherwise.
c. HTML files can also be well-formed XML files.
d. HTML files can also be well-formed XHTML files.

G XML validation (11 points)

(1) (1 point) What is the purpose of XML validation?

A. To make sure that a given XML document adheres to the rules of XML syntax.

B. To make sure that a given XML parser implements the XML standard validly.

C. To make sure that a given XML document fulfills structural and type restrictions.

D. To make sure that a given user application implements the parsing of the given XML document correctly.

(2) (1 point) We can say that a certain XML document is well-formed or not well-formed. We can also say that a certain XML document is valid or not valid against a certain schema. Which of the four possible combinations of these two states is *impossible*?

A. not well-formed **and** not valid

B. not well-formed **and** valid

C. well-formed **and** not valid

D. well-formed **and** valid

(3) (1 point) Imagine that you have to validate 1 billion XML documents against a certain schema. Which of the following ways of accelerating this process is most likely to provide the best results?

A. Writing a faster XML validator.

B. Parallelising the validation across multiple machines.

C. Improving the hardware of the computer which performs the validation.

D. Combining the documents into a single one and running the validation on that single document.

(4) (2 points) Which of the following is the best-fitting use case for XML validation?

A. Ensuring that two distinct applications that exchange complex information in XML format can understand each other.

B. Ensuring that an application that generates certain XML files does so as fast as possible.

C. Ensuring the confidentiality of XML documents transmitted over the Internet between two applications.

D. Ensuring that an application that generates certain XML files is stable against hardware failures.

(5) (2 points) Mark each of the following statements about differences between DTD validation and XML Schema validation as either true or false. On the answer sheet, mark your answers with a checkmark (✓).

a. XML Schemas can specify the order in which child elements must appear, while DTD cannot.

b. XML Schemas support XML Namespaces, while DTD does not.

c. XML Schemas can specify a default attribute value, while DTD cannot.

d. DTD can specify that an element must be empty, while XML Schemas cannot.

(6) (2 points) Mark each of the following statements about XML Schemas as either true or false. On the answer sheet, mark your answers with a checkmark (✓).

a. An XML Schema cannot apply to elements that belong to no namespace.

b. An XML Schema cannot refer to elements from namespaces other than its *target* namespace.

c. An XML Schema that specifies a target namespace and an XML document that is *valid* against that schema must both use the same namespace prefix.

d. There exists an XML Schema for XML Schemas.

(7) (2 points) Consider the following XML Schema instance. For each of the XML documents given below, state whether it will successfully be validated against this schema or not.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="restaurant">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="visitedOn" type="xs:date"
                                     minOccurs="0"
                                     maxOccurs="unbounded"/>
        <xs:element name="borough" type="xs:string" minOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

a.
```
<restaurant>
  <name>Vella</name>
  <visitedOn>true</visitedOn>
  <borough>Manhattan</borough>
</restaurant>
```
A. valid B. not valid

b.
```
<restaurant>
  <name>Vella</name>
  <borough>Manhattan</borough>
  <visitedOn>2017-10-26</visitedOn>
  <visitedOn>2017-12-14</visitedOn>
  <visitedOn>2017-12-23</visitedOn>
</restaurant>
```
A. valid B. not valid

c.
```
<restaurant>
  <name>Vella</name>
  <borough>Manhattan</borough>
</restaurant>
```
A. valid B. not valid

d.
```
<restaurant>
  <borough>Manhattan</borough>
  <name>Vella</name>
</restaurant>
```
A. valid B. not valid

H XQuery (12 points)

(1) (1 point) Choose the most precise ending to the phrase: "XQuery operates on ..."

A. DAG-structured data
B. Graph-structured data
C. Tree-structured data
D. Table-structured data

(2) (2 points) Mark each of the following statements as either true or false. On the answer sheet, mark your answers with a checkmark (✓).

a. Clauses in a FLWOR expression have to appear in this exact order and only at most once: For-Let-Where-Order-Return.
b. XQuery supports XML Namespaces.
c. There exist XQuery expressions whose output is serializable to well-formed XML.
d. A FLWOR expression is called "not well-formed" if it generates XML output which is not well-formed.

(The exam continues on the next page.)

The following questions will be about specific queries and will use the following XML document (it is well-formed):

```
<bookstore>
    <book category="CHILDREN">
        <title>If Animals Kissed Good Night</title>
        <author>Ann Whitford Paul</author>
        <author>David Walker</author>
        <year>2014</year>
        <price>5.86</price>
    </book>
    <book category="CHILDREN">
        <title>Dragons Love Tacos</title>
        <author>Adam Rubin</author>
        <author>Daniel Salmieri</author>
        <year>2012</year>
        <price>10.79</price>
    </book>
    <book category="COOKING">
        <title>Cooking Under Pressure</title>
        <author>Lorna J. Sass</author>
        <year>2009</year>
        <price>16.31</price>
    </book>
    <book category="COOKING">
        <title>Culinary Reactions</title>
        <author>Simon Quellen Field</author>
        <year>2011</year>
        <price>11.74</price>
    </book>
</bookstore>
```

(The questions start on the next page)

For each of the following XQuery 3.0 expressions, indicate what it will return when run on the document presented on the previous page.

(3) (1 point)

```
for $book in /bookstore/book
return $book/title
```

A.

```
<title>If Animals Kissed Good Night</title>
<title>Dragons Love Tacos</title>
<title>Cooking Under Pressure</title>
<title>Culinary Reactions</title>
```

B.

```
<book>
  <title>If Animals Kissed Good Night</title>
</book>
<book>
  <title>Dragons Love Tacos</title>
</book>
<book>
  <title>Cooking Under Pressure</title>
</book>
<book>
  <title>Culinary Reactions</title>
</book>
```

C.

```
If Animals Kissed Good Night
Dragons Love Tacos
Cooking Under Pressure
Culinary Reactions
```

D.

```
<title/>
<title/>
<title/>
<title/>
```

(4) (1 point)

```
(for $book in /bookstore/book
order by xs:integer($book/year) descending
return $book/title)[1]
```

A.
```
<title>If Animals Kissed Good Night</title>
<title>Dragons Love Tacos</title>
<title>Culinary Reactions</title>
<title>Cooking Under Pressure</title>
```

B.
```
If Animals Kissed Good Night
Dragons Love Tacos
Culinary Reactions
Cooking Under Pressure
```

C.
```
If Animals Kissed Good Night
```

D.
```
<title>If Animals Kissed Good Night</title>
```

(5) (1½ points)

```
for $book in /bookstore/book
group by $foo := $book/@category
where $book/year = "2014"
return $book/year
```

A.
```
<year>2014</year>
```

B.
```
<year>2014</year>
<year>2012</year>
```

C.
```
Error
```

D.
```
2014
```

(6) (1½ points)

```
for $book in /bookstore/book
group by $foo := $book/@category
where $book/year eq "2014"
return $book/year
```

A.
```
<year>2014</year>
```

B.
```
<year>2014</year>
<year>2012</year>
```

C.
```
Error
```

D.
```
2014
```

(7) (2 points)

```
for $book in /bookstore/book
group by $foo := $book/@category
let $bar := (
    for $toe in $book
    order by xs:float($toe/price) ascending
    return $toe
)
return $bar[1]/title
```

A.
```
<bar foo="CHILDREN">
  <title>If Animals Kissed Good Night</title>
  <title>Dragons Love Tacos</title>
</bar>
<bar foo="COOKING">
  <title>Culinary Reactions</title>
  <title>Cooking Under Pressure</title>
</bar>
```

B.
```
<title>If Animals Kissed Good Night</title>
<title>Dragons Love Tacos</title>
<title>Culinary Reactions</title>
<title>Cooking Under Pressure</title>
```

C.
```
Empty sequence
```

D.
```
<title>If Animals Kissed Good Night</title>
<title>Culinary Reactions</title>
```

(8) (2 points)

```
for $book in /bookstore/book
where xs:integer($book/year) lt 2013
group by $foo := $book/@category
where count($book/author) eq 2
return $book/author
```

A.
```
Syntax error:  cannot have 2 "where" clauses
```

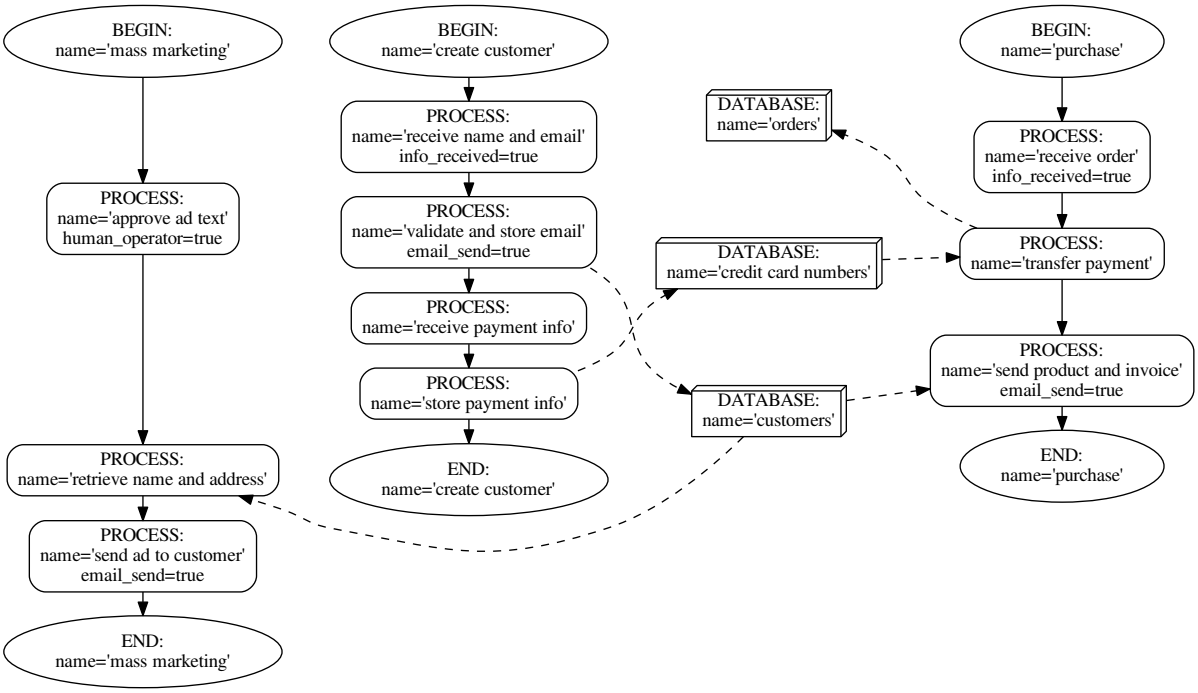
B.
```
<book>
  <author>Adam Rubin</author>
  <author>Daniel Salmieri</author>
</book>
```

C.
```
<author>Adam Rubin</author>
<author>Daniel Salmieri</author>
```

D.
```
<author>Lorna J. Sass</author>
<author>Simon Quellen Field</author>
<author>Adam Rubin</author>
<author>Daniel Salmieri</author>
```

I Graph databases (12 points)

Consider the following graph database, which represents the "Business Processes" that happen at a company.



The uppercase label determines the node type, lowercase key/value are the node properties. Full edges have type `PROCESSFLOW`, dashed edges have type `DATAFLOW`. Edges have no properties.

In all the following questions, you will find *syntactically* correct Cypher queries. On the answer sheet, mark with a checkmark (✓) which of the following statements are correct, or select the correct answer.

(1) (1 point) The query:

```
MATCH (s:PROCESS)
WHERE s.email_send = TRUE
RETURN s
```

will return exactly two nodes.

(2) (1 point) The query:

```
MATCH (b:BEGIN)-[*]->(db:DATABASE)
WHERE db.name='customers'
RETURN b.name
```

will return exactly one row containing `"create customer"`.

(3) (1 point) The query:

```
MATCH (p:PROCESS)-[:DATAFLOW]-(:DATABASE)
RETURN p
```

will return exactly six nodes: all of those who have a direct data flow from or to a database.

(4) (2 points) The query:

```
MATCH (b:BEGIN {name:"create customer"}),(e),p=shortestPath((b)-[*]->(e))
WHERE b <> e AND length(p) >= 6
RETURN e.name
```

will return (assuming order is irrelevant):

A. `"mass marketing", "orders", "transfer payment"`
B. `"mass marketing", "orders", "transfer payment", "send product and invoice", "purchase"`
C. An error
D. None of the above

(5) (2 points) The query:

```
MATCH (b:BEGIN {name:"create customer"}),(e),p=shortestPath((b)-[*]-(e))
WHERE b <> e
WITH e, p
WHERE length(p) >= 6
RETURN e.name
```

will return (assuming order is irrelevant):

A. `"mass marketing", "orders", "transfer payment"`
B. `"mass marketing", "orders", "transfer payment", "send product and invoice", "purchase"`
C. An error
D. None of the above

(6) (1 point) Data models based on trees and graphs were invented before the relational data model.

(7) (1 point) Is it possible for a graph database to support a large number of concurrent reads (when there are too many for one single machine)?

A. Yes, for example, Neo4j can do this.
B. Yes, but you have to use a cluster framework like GraphX (on Spark).
C. No, this is fundamentally incompatible with graph databases.

(8) (1 point) Is it possible for a graph database to support a large number of concurrent writes (when there are too many for one single machine)?

A. Yes, for example, Neo4j can do this.
B. Yes, but you have to use a cluster framework like GraphX (on Spark).
C. No, this is fundamentally incompatible with graph databases.

(9) (1 point) Under the hood, Neo4j and Cypher use as a data model:

A. A labeled property graph

B. A RDF triple store

C. A mix of both

D. None of the above

(10) (1 point) In general, RDF Triple Stores are more suited than triple graphs to very large sets of data.

J Relational databases (19 points)

(1) (1 point) Who is considered to be the inventor of relational algebra?
   A. Jeffrey Dean
   B. Sanjay Ghemawat
   C. Edgar Codd
   D. Matei Zaharia

(2) (1 point) In relational algebra, which of the following is *not* a unary operator with respect to input relations?
   A. Natural join
   B. Projection
   C. Selection
   D. Rename

(3) (1 point) Which of the following pairs is made of elements that *cannot* be used as synonyms?
   A. Document - Record
   B. Column - Attribute
   C. Primary key - Id
   D. Row - Field

(4) (1 point) Which one of these concepts is not commonly used as a synonym for *record*?
   A. Tuple
   B. Attribute
   C. Row
   D. Document

(5) (1 point) Which one of these relational algebra operations is usually avoided in NoSQL technologies?
   A. Selection
   B. Grouping
   C. Join
   D. Projection

(6) Which of the following statements about *database normalization* are correct? On the answer sheet, mark the correct answers with a checkmark (✓).
   a. (½ point) It is a process of organizing attributes and relations of a relational database to reduce data redundancy.
   b. (½ point) It guarantees data consistency.
   c. (½ point) It was first proposed by Jeffrey Dean.
   d. (½ point) It was proposed together with the relational model for databases.
   e. (½ point) A relational database relation is often described as "normalized" if it meets second normal form (2NF).
   f. (½ point) It often helps avoiding insertion anomalies.
   g. (½ point) It often helps avoiding update anomalies.
   h. (½ point) It often helps avoiding deletion anomalies.
   i. (½ point) It often helps reducing the storage required for the data.

j. (½ point) It usually increases the number of joins required to retrieve data.
k. (½ point) It primarily aims at reducing the time necessary to run queries.
l. (½ point) It is a common practice for NoSQL databases as well as for SQL ones.

(7) (1 point) Which one of these queries performs a projection?
   A. SELECT id, email, username FROM users
   B. SELECT id FROM users WHERE email = 'jhon.doo@foo.com'
   C. Both of them
   D. None of them

(8) (1 point) Which one of these queries performs a selection?
   A. SELECT id, email, username FROM users
   B. SELECT id FROM users WHERE email = 'jhon.doo@foo.com'
   C. Both of them
   D. None of them

(9) (1 point) When a query is executed, which of the following clauses is the last logically performed?
   A. WHERE
   B. ORDER BY
   C. GROUP BY
   D. FROM

(10) (1 point) Which of the following statements is *not* true when talking about atomicity?
   A. It is the 'A' in ACID.
   B. It is the 'A' in the CAP theorem.
   C. It means that any updates that a transaction might affect on a system are completed in their entirety.
   D. HBase provides row-level atomicity.

(11) (1 point) Which of the following is a necessary condition for a relation to be in 1NF?
   A. The candidate key is unique.
   B. There are no null values in the primary key fields.
   C. All non-prime attributes are atomic.
   D. None of the options above.

(12) (1 point) Which of the following is a necessary condition for a relation to be in 2NF?
   A. The candidate key is unique.
   B. There are no functional dependencies.
   C. All non-prime attributes are atomic.
   D. None of the options above.

(13) (1 point) Which of the following is a necessary condition for a relation to be in 3NF?
   A. The candidate key is unique.
   B. There are no functional dependencies.
   C. The primary key is made of a single column.
   D. None of the options above.

(14) (1 point) Should an SQL database be considered OLAP or OLTP?

A. OLAP
B. OLTP
C. It depends on the size of the database.
D. It depends on the design of the database.

K MapReduce (15 points)

(1) (1 point) MapReduce is...
   A. a software written in Java.
   B. an Hadoop framework first developed by Google.
   C. a programming model.
   D. None of the above.

(2) (1 point) When was MapReduce first proposed?
   A. In the early 1980s.
   B. In the early 1990s.
   C. In the early 2000s.
   D. In the early 2010s.

(3) Which of the following statements about MapReduce are correct? On the answer sheet, mark the correct answers with a checkmark (✓).
   a. (½ point) MapReduce is designed to work on specialized hardware.
   b. (½ point) MapReduce is designed to work on hundreds or thousands of machines, taking into account the possibility of hardware failure.
   c. (½ point) Two different reducers might process pairs with the same key if the key distribution for a job is unbalanced.
   d. (½ point) The mapper produces exactly one key-value pair for each key-value pair provided as input.
   e. (½ point) If mapper and reducer both implement the identity function, the output file of the overall MapReduce job will always have the same content as the input file.
   f. (½ point) Apache Hadoop provides an open-source implementation of MapReduce.
   g. (½ point) In Hadoop MapReduce v1, the JobTracker is responsible for scheduling mappers and reducers and make sure all nodes are correctly running.
   h. (½ point) In Hadoop MapReduce v1, it is best practice that the TaskTracker runs on NameNodes.
   i. (½ point) The TaskTracker is replaced by Node Manager in MapReduce v2.
   j. (½ point) TaskTracker failure is not considered fatal: when a TaskTracker becomes unresponsive, JobTracker will wait for it to recover.
   k. (½ point) The Partitioner is responsible for ensuring that each reducer receives the same number of key-value pairs.
   l. (½ point) In Hadoop, by default, the output of the mappers is temporally stored in HDFS from where it will be read by reducers.

(4) (1 point) Which of the following statements is true about key-value pairs in Hadoop MapReduce?
   A. Both key and value must implement Writable.
   B. Both key and value must implement WritableComparable.
   C. All of the above.
   D. None of the above.

(5) (1 point) Which of the following statements is true about MapReduce splits?

A. They are the physical blocks of data used as input to a MapReduce job.
B. The number of map and reduce tasks matches the number of splits.
C. The content of a single split might be stored on different physical nodes.
D. None of the above.

(6) (1 point) Consider the following map function,

```
function map(key, value)
   emit(1, key);
```

can we say that it leads to a low degree of parallelization?

A. No, because all values are discarded .
B. No, because the degree of parallelization of a job does not depend on the choice of mapper and reduce functions.
C. Yes, because one single Mapper will have to process the all input data.
D. Yes, because one single Reducer will have to process the intermediate data.

(7) (4 points) Consider the following map and reduce functions, where the inputs to the mapper are integers for both key and value,

```
function map(key, value)
   emit(key, value);


function reduce(key, values[])
   n, sum, sum_sq = 0
   for value in values:
      sum += value
      sum_sq += value*value
      n += 1
   tmp = (sum_sq / n) - (sum / n)^2
   emit(key, tmp)
```

a. (1 point) What can you say about the output of this MapReduce job?
   A. The output consists of as many key-value pairs as distinct keys are in the input to the mapper.
   B. The output consists of a key-value pair for each key-value pair in input to the mapper.
   C. The output consists of a single key-value pair.
   D. There is no output because this is not a valid MapReduce job.

b. (1 point) Which of the following statements is correct about the reduce function?
   A. It is associative.
   B. It is commutative.
   C. It is neither associative nor commutative.
   D. Is it both associative and commutative.

c. (1 point) Can the reduce function be used for the combiner as well?
   A. Yes, and this will make job significantly faster.

B. Yes, but in this case we would not observe any significant improvement in running time.

C. No, the result would be surely wrong (ie. different than the one not using the combiner).

D. No, the result might be wrong.

d. (1 point) Re-write the map function such that, using the same reducer, the output of the job would be the variance of *all* input values. Report your answer in the answer sheet.

L Data Warehousing (12 points)

Consider the following fact table:

`CarSales(Model, Color, Date, Quantity, Value)`,

which stores the number of cars of a particular model and color sold on a particular date as well as their combined value.

(1) (1 point) For each of the attributes in the table, say whether they are a dimension or a measure in a typical analysis.

a. Model: A. dimension B. measure
b. Color: A. dimension B. measure
c. Date: A. dimension B. measure
d. Quantity: A. dimension B. measure
e. Value: A. dimension B. measure

(2) (1 point) In a ROLAP query written in SQL, which of the following clauses must be changed always if we change the slicing:

a. `SELECT`: A. yes B. no
b. `FROM`: A. yes B. no
c. `JOIN`: A. yes B. no
d. `WHERE`: A. yes B. no
e. `GROUP BY`: A. yes B. no

(3) (1 point) Which clauses must be changed always if we change the dicing:

a. `SELECT`: A. yes B. no
b. `FROM`: A. yes B. no
c. `JOIN`: A. yes B. no
d. `WHERE`: A. yes B. no
e. `GROUP BY`: A. yes B. no

(4) (1 point) Which clauses must be changed always for roll-up:

a. `SELECT`: A. yes B. no
b. `FROM`: A. yes B. no
c. `JOIN`: A. yes B. no
d. `WHERE`: A. yes B. no
e. `GROUP BY`: A. yes B. no

(5) (1 point) Which clauses must be changed always for drill-down:

a. `SELECT`: A. yes B. no
b. `FROM`: A. yes B. no
c. `JOIN`: A. yes B. no
d. `WHERE`: A. yes B. no
e. `GROUP BY`: A. yes B. no

(6) For each of the statements below, say whether it more typical for OLAP or for OLTP applications.

a. (½ point) The workload is read-mostly or read-only. A. OLTP B. OLAP
b. (½ point) Most queries only touch a small fraction of the data. A. OLTP B. OLAP
c. (½ point) The schema is normalized in order to keep the data consistent. A. OLTP B. OLAP
d. (½ point) The database includes a lot of historic data. A. OLTP B. OLAP
e. (½ point) The database contains operational data. A. OLTP B. OLAP

(7) (4½ points) Consider the following SQL queries.

(Q1)
```
SELECT Model, Color, SUM(Value)
FROM CarSales
GROUP BY Model, Color WITH CUBE
```

(Q2)
```
SELECT Model, Color, SUM(Value)
FROM CarSales
GROUP BY Model, Color WITH ROLLUP
```

(Q3)
```
SELECT Model, Color, SUM(Value)
FROM CarSales
GROUP BY Model, Color WITH DRILLDOWN
```

(Q4)
```
(
    SELECT Model, Color, SUM(Value)
    FROM CarSales
    GROUP BY Model, Color
) UNION ALL (
    SELECT NULL AS Model, Color, SUM(Value)
    FROM CarSales
    GROUP BY Color
) UNION ALL (
    SELECT Model, NULL as Color, SUM(Value)
    FROM CarSales
    GROUP BY Model
) UNION ALL (
    SELECT NULL, NULL, SUM(Value)
    FROM CarSales
)
```

(Q5)
```
(
    SELECT Model, Color, SUM(Value)
    FROM CarSales
    GROUP BY Model, Color
) UNION ALL (
    SELECT Model, NULL, SUM(Value)
    FROM CarSales
    GROUP BY Color
) UNION ALL (
    SELECT NULL, NULL, SUM(Value)
    FROM CarSales
)
```

(Q6)
```
SELECT Model, Color, SUM(Value)
FROM CarSales
GROUP BY GROUPING SETS(
    (Model, Color),
    (Model),
    (Color),
    ()
)
```

(Q7)
```
SELECT Model, Color, SUM(Value)
FROM CarSales
GROUP BY GROUPING SETS(
    (Model, Color),
    (Model),
    ()
)
```

(Q8)
```
SELECT Model, Color, SUM(Value)
FROM CarSales
GROUP BY GROUPING SETS(
    (Model, Color),
    ()
)
```

(Q9)
```
SELECT Model, Color, SUM(Value)
FROM CarSales
GROUP BY GROUPING SETS(
    (Model, Color)
)
```

We say that two queries are equivalent if they return the same *set* of rows no matter which database instance they are run on. (Note: Since we only care about sets of rows, their order does not matter.) Errornous queries are not equivalent to any other query.

Mark each of the following statements as either true or false. On the answer sheet, mark your answers with a checkmark (✓).

a. Q1 is equivalent to Q4.
b. Q1 is equivalent to Q5.
c. Q1 is equivalent to Q6.
d. Q1 is equivalent to Q7.
e. Q1 is equivalent to Q8.
f. Q1 is equivalent to Q9.
g. Q2 is equivalent to Q4.
h. Q2 is equivalent to Q5.
i. Q2 is equivalent to Q6.
j. Q2 is equivalent to Q7.
k. Q2 is equivalent to Q8.
l. Q2 is equivalent to Q9.
m. Q3 is equivalent to Q4.
n. Q3 is equivalent to Q5.
o. Q3 is equivalent to Q6.
p. Q3 is equivalent to Q7.
q. Q3 is equivalent to Q8.
r. Q3 is equivalent to Q9.

M General questions (15 points)

(1) (1 point) Which of the following was proposed in the early 1970s?

A. HBase
B. Relational algebra
C. Map Reduce
D. DynamoDB

(2) (2 points) Compare SQL and NoSQL systems. Which of the following statements are correct? On the answer sheet, mark the correct answers with a checkmark (✓).

a. Compared to SQL ones, NoSQL systems are usually designed for simpler scaling out.

b. Triple store, wide column store, graph database and document store are all NoSQL paradigms.

c. In the context of the CAP theorem, relational databases often compromise consistency.

d. There are NoSQL systems that are ACID compliant.

(3) (2 points) On the answer sheet, complete the table with the four missing names for the multiples of the unit *byte*.

Multiple	**Unit**
10^6	megabyte
10^9	gigabyte
10^{12}	terabyte
10^{15}	???
10^{18}	???
10^{21}	???
10^{24}	???

(4) (2 points) Which of the following is true observing the trends of the last 60 years? On the answer sheet, mark the correct answers with a checkmark (✓).

a. Throughput increased faster than latency decreased.

b. Throughput increased faster than storage capacity.

c. Storage capacity increased faster than latency decreased.

d. Latency decreased by several orders of magnitude.

(5) (1 point) Batch processing is required to overcome the slow improvement of which of the following metrics?

A. Storage capacity
B. Throughput
C. Latency
D. Availability

(6) (1 point) A Bloom filter is constructed on top a of a set of values. Later on, it produces a positive response when tested against a possibly new value x. Which of the following is true?

A. Value x is for sure in the set.
B. Value x might or might not be in the set.
C. Value x is for sure not in the set.

D. A Bloom filter cannot be applied to a set.

(7) (1 point) Which one of these database paradigms is *not* considered to be NoSQL?
   A. Triple stores
   B. Wide column stores
   C. Relational databases
   D. Graph databases

(8) (1 point) What do 'C', 'A' and 'P' stand for in the CAP theorem? Report your answer in the answer sheet.

(9) (1 point) Suppose you want to choose a language for a protocol, for which a typical message contains 10 strings and 20 numerical values. Which of the following choices is going to lead to minimal network usage, assuming you can compress messages when sending them?
   A. XML
   B. JSON
   C. Protocol buffers
   D. CSV

(10) (3 points) Mark each of the following statements as either true or false. On the answer sheet, mark your answers with a checkmark (✓).
   a. Amdahl's law applies when the problem size is constant
   b. Gustafson's law assumes that the serial part can be split among multiple processors
   c. The probability of hitting the tail latency is higher when using a single server than when simply splitting the request across hundreds of servers
   d. Average latency is the same as tail latency
   e. Tail latency is driven by the latency of just a few requests among the many performed
   f. By using hedged and deferred hedged requests, the tail latency problem will disappear even when more servers are added

Empty Pages

You can use the following pages at your convenience.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Big Data
Fall Semester 2017
Lecturer(s): Ghislain Fourny
Assistant(s): Alexandr Nigay, Damien Desfontaines, Ingo Müller, Konstantin Taranov, Marco Ancona, Renato Marroquín

ETH Zurich

Date: January 25, 2018
Last update: January 19, 2018

Exam

Answer Sheets

Name: _____ **Legi-Nr:** _____

Question:	A	B	C	D	E	F	G	H	I	J	K	L	M	Total
Points:	19	17	13	11	13	12	11	12	12	19	15	12	15	181
Score:														

Reminders

- Write all your answers on this stack of papers ("answer sheets").
- Write your name and Legi number on the cover page of the answer sheets.
- Please write your Legi number on all other answer sheets.
- Do not separate the individual sheets of this stack. If there are pages missing, you will not receive any points for these pages.
- Hand in this stack of answer sheets.

A Block storages

A.1.a: ◯ true ◯ false
A.1.b: ◯ true ◯ false
A.1.c: ◯ true ◯ false
A.1.d: ◯ true ◯ false
A.1.e: ◯ true ◯ false
A.1.f: ◯ true ◯ false
A.1.g: ◯ true ◯ false
A.1.h: ◯ true ◯ false
A.2.a: ◯ true ◯ false
A.2.b: ◯ true ◯ false
A.2.c: ◯ true ◯ false
A.2.d: ◯ true ◯ false
A.2.e: ◯ true ◯ false
A.2.f: ◯ true ◯ false
A.2.g: ◯ true ◯ false
A.3.a: ◯ true ◯ false
A.3.b: ◯ true ◯ false
A.3.c: ◯ true ◯ false
A.3.d: ◯ true ◯ false
A.4.a: ◯ true ◯ false
A.4.b: ◯ true ◯ false
A.4.c: ◯ true ◯ false
A.4.d: ◯ true ◯ false
A.4.e: ◯ true ◯ false
A.5.a: ◯ NameNode ◯ DataNode
A.5.b: ◯ NameNode ◯ DataNode
A.5.c: ◯ NameNode ◯ DataNode
A.5.d: ◯ NameNode ◯ DataNode

Question A.6:

B Object storage and key-value stores

B.1.a: ◯ true ◯ false
B.1.b: ◯ true ◯ false
B.1.c: ◯ true ◯ false
B.1.d: ◯ true ◯ false
B.1.e: ◯ true ◯ false
B.1.f: ◯ true ◯ false
B.1.g: ◯ true ◯ false
B.1.h: ◯ true ◯ false
B.1.i: ◯ true ◯ false
B.1.j: ◯ true ◯ false
B.1.k: ◯ true ◯ false
B.1.l: ◯ true ◯ false
B.2.a: ◯ true ◯ false
B.2.b: ◯ true ◯ false
B.2.c: ◯ true ◯ false
B.2.d: ◯ true ◯ false
B.2.e: ◯ true ◯ false
B.2.f: ◯ true ◯ false
B.3: ◯ A ◯ B ◯ C ◯ D
B.4: ◯ A ◯ B ◯ C ◯ D ◯ E
B.5.a: ◯ true ◯ false
B.5.b: ◯ true ◯ false
B.5.c: ◯ true ◯ false
B.5.d: ◯ true ◯ false

C HBase

C.1: ◯ A ◯ B ◯ C ◯ D
C.2: ◯ A ◯ B ◯ C ◯ D ◯ E
C.3: ◯ A ◯ B ◯ C ◯ D
C.4: ◯ A ◯ B ◯ C ◯ D

Question C.5:

Query: `get 'phrases', '491'`			
Row key	**Column**	**Time-stamp**	**Value**

Query: `get 'phrases', '900'`			
Row key	**Column**	**Time-stamp**	**Value**

Query: `get 'phrases', '743'`			
Row key	**Column**	**Time-stamp**	**Value**

D Apache Spark and YARN

D.1.a: ◯ true ◯ false
D.1.b: ◯ true ◯ false
D.1.c: ◯ true ◯ false
D.1.d: ◯ true ◯ false
D.2.a: ◯ at most one ◯ may be many ◯ none
D.2.b: ◯ at most one ◯ may be many ◯ none
D.2.c: ◯ at most one ◯ may be many ◯ none
D.2.d: ◯ at most one ◯ may be many ◯ none
D.2.e: ◯ at most one ◯ may be many ◯ none
D.2.f: ◯ at most one ◯ may be many ◯ none
D.3.a: ◯ true ◯ false
D.3.b: ◯ true ◯ false
D.3.c: ◯ true ◯ false
D.3.d: ◯ true ◯ false
D.4.a: ◯ Yes ◯ No
D.4.b: ◯ Yes ◯ No
D.4.c: ◯ Yes ◯ No
D.4.d: ◯ Yes ◯ No
D.5.a: ◯ A ◯ B ◯ C ◯ D
D.5.b: ◯ A ◯ B ◯ C ◯ D
D.5.c: ◯ A ◯ B ◯ C ◯ D
D.5.d: ◯ A ◯ B ◯ C ◯ D

E Document stores

E.1.a: ◯ true ◯ false
E.1.b: ◯ true ◯ false
E.1.c: ◯ true ◯ false
E.1.d: ◯ true ◯ false
E.1.e: ◯ true ◯ false
E.1.f: ◯ true ◯ false

Question E.2:

Question E.3:

E.4.a: ◯ yes ◯ no
E.4.b: ◯ yes ◯ no
E.4.c: ◯ yes ◯ no
E.4.d: ◯ yes ◯ no
E.4.e: ◯ yes ◯ no
E.5.a: ◯ yes ◯ no
E.5.b: ◯ yes ◯ no
E.5.c: ◯ yes ◯ no
E.5.d: ◯ yes ◯ no
E.5.e: ◯ yes ◯ no
E.5.f: ◯ yes ◯ no
E.5.g: ◯ yes ◯ no
E.5.h: ◯ yes ◯ no
E.5.i: ◯ yes ◯ no

F Well-formedness

F.1.a: ◯ true ◯ false
F.1.b: ◯ true ◯ false
F.1.c: ◯ true ◯ false
F.1.d: ◯ true ◯ false
F.1.f: ◯ true ◯ false
F.1.g: ◯ true ◯ false
F.1.h: ◯ true ◯ false
F.1.i: ◯ true ◯ false
F.1.j: ◯ true ◯ false
F.1.k: ◯ true ◯ false
F.1.l: ◯ true ◯ false
F.2.a: ◯ true ◯ false
F.2.b: ◯ true ◯ false
F.2.c: ◯ true ◯ false
F.2.d: ◯ true ◯ false
F.2.e: ◯ true ◯ false
F.2.f: ◯ true ◯ false
F.2.g: ◯ true ◯ false
F.2.h: ◯ true ◯ false
F.2.i: ◯ true ◯ false
F.2.j: ◯ true ◯ false
F.2.k: ◯ true ◯ false
F.2.l: ◯ true ◯ false
F.3.a: ◯ true ◯ false
F.3.b: ◯ true ◯ false
F.3.c: ◯ true ◯ false
F.3.d: ◯ true ◯ false

G XML validation

G.1: ◯ A ◯ B ◯ C ◯ D
G.2: ◯ A ◯ B ◯ C ◯ D
G.3: ◯ A ◯ B ◯ C ◯ D
G.4: ◯ A ◯ B ◯ C ◯ D
G.5.a: ◯ true ◯ false
G.5.b: ◯ true ◯ false
G.5.c: ◯ true ◯ false
G.5.d: ◯ true ◯ false
G.6.a: ◯ true ◯ false
G.6.b: ◯ true ◯ false
G.6.c: ◯ true ◯ false
G.6.d: ◯ true ◯ false
G.7.a: ◯ valid ◯ not valid
G.7.b: ◯ valid ◯ not valid
G.7.c: ◯ valid ◯ not valid
G.7.d: ◯ valid ◯ not valid

H XQuery

H.1: ◯ A ◯ B ◯ C ◯ D
H.2.a: ◯ true ◯ false
H.2.b: ◯ true ◯ false

H.2.c: ◯ true ◯ false
H.2.d: ◯ true ◯ false
H.3: ◯ A ◯ B ◯ C ◯ D
H.4: ◯ A ◯ B ◯ C ◯ D
H.5: ◯ A ◯ B ◯ C ◯ D
H.6: ◯ A ◯ B ◯ C ◯ D
H.7: ◯ A ◯ B ◯ C ◯ D
H.8: ◯ A ◯ B ◯ C ◯ D

I Graph databases

I.1: ◯ true ◯ false
I.2: ◯ true ◯ false
I.3: ◯ true ◯ false
I.4: ◯ A ◯ B ◯ C ◯ D
I.5: ◯ A ◯ B ◯ C ◯ D
I.6: ◯ true ◯ false
I.7: ◯ A ◯ B ◯ C
I.8: ◯ A ◯ B ◯ C
I.9: ◯ A ◯ B ◯ C ◯ D
I.10: ◯ true ◯ false

J Relational databases

J.1: ◯ A ◯ B ◯ C ◯ D
J.2: ◯ A ◯ B ◯ C ◯ D
J.3: ◯ A ◯ B ◯ C ◯ D
J.4: ◯ A ◯ B ◯ C ◯ D
J.5: ◯ A ◯ B ◯ C ◯ D
J.6.a: ◯ true ◯ false
J.6.b: ◯ true ◯ false
J.6.c: ◯ true ◯ false
J.6.d: ◯ true ◯ false
J.6.e: ◯ true ◯ false
J.6.f: ◯ true ◯ false
J.6.g: ◯ true ◯ false
J.6.h: ◯ true ◯ false
J.6.i: ◯ true ◯ false
J.6.j: ◯ true ◯ false
J.6.k: ◯ true ◯ false
J.6.l: ◯ true ◯ false
J.7: ◯ A ◯ B ◯ C ◯ D
J.8: ◯ A ◯ B ◯ C ◯ D
J.9: ◯ A ◯ B ◯ C ◯ D
J.10: ◯ A ◯ B ◯ C ◯ D
J.11: ◯ A ◯ B ◯ C ◯ D
J.12: ◯ A ◯ B ◯ C ◯ D
J.13: ◯ A ◯ B ◯ C ◯ D
J.14: ◯ A ◯ B ◯ C ◯ D

K MapReduce

K.1: ◯ A ◯ B ◯ C ◯ D
K.2: ◯ A ◯ B ◯ C ◯ D
K.3.a: ◯ true ◯ false
K.3.b: ◯ true ◯ false
K.3.c: ◯ true ◯ false
K.3.d: ◯ true ◯ false
K.3.e: ◯ true ◯ false
K.3.f: ◯ true ◯ false
K.3.g: ◯ true ◯ false
K.3.h: ◯ true ◯ false
K.3.i: ◯ true ◯ false
K.3.j: ◯ true ◯ false
K.3.k: ◯ true ◯ false
K.3.l: ◯ true ◯ false
K.4: ◯ A ◯ B ◯ C ◯ D
K.5: ◯ A ◯ B ◯ C ◯ D
K.6: ◯ A ◯ B ◯ C ◯ D
K.7.a: ◯ A ◯ B ◯ C ◯ D
K.7.b: ◯ A ◯ B ◯ C ◯ D
K.7.c: ◯ A ◯ B ◯ C ◯ D

Question K.7.d:

L Data Warehousing

L.1.a: ◯ dimension ◯ measure
L.1.b: ◯ dimension ◯ measure
L.1.c: ◯ dimension ◯ measure
L.1.d: ◯ dimension ◯ measure
L.1.e: ◯ dimension ◯ measure
L.2.a: ◯ yes ◯ no
L.2.b: ◯ yes ◯ no
L.2.c: ◯ yes ◯ no
L.2.d: ◯ yes ◯ no
L.2.e: ◯ yes ◯ no
L.3.a: ◯ yes ◯ no
L.3.b: ◯ yes ◯ no
L.3.c: ◯ yes ◯ no
L.3.d: ◯ yes ◯ no
L.3.e: ◯ yes ◯ no
L.4.a: ◯ yes ◯ no
L.4.b: ◯ yes ◯ no
L.4.c: ◯ yes ◯ no
L.4.d: ◯ yes ◯ no
L.4.e: ◯ yes ◯ no
L.5.a: ◯ yes ◯ no
L.5.b: ◯ yes ◯ no
L.5.c: ◯ yes ◯ no
L.5.d: ◯ yes ◯ no
L.5.e: ◯ yes ◯ no
L.6.a: ◯ OLTP ◯ OLAP
L.6.b: ◯ OLTP ◯ OLAP
L.6.c: ◯ OLTP ◯ OLAP
L.6.d: ◯ OLTP ◯ OLAP
L.6.e: ◯ OLTP ◯ OLAP
L.7.a: ◯ true ◯ false

L.7.b: ◯ true ◯ false
L.7.c: ◯ true ◯ false
L.7.d: ◯ true ◯ false
L.7.e: ◯ true ◯ false
L.7.f: ◯ true ◯ false
L.7.g: ◯ true ◯ false
L.7.h: ◯ true ◯ false
L.7.i: ◯ true ◯ false
L.7.j: ◯ true ◯ false
L.7.k: ◯ true ◯ false
L.7.l: ◯ true ◯ false
L.7.m: ◯ true ◯ false
L.7.n: ◯ true ◯ false
L.7.o: ◯ true ◯ false
L.7.p: ◯ true ◯ false
L.7.q: ◯ true ◯ false
L.7.r: ◯ true ◯ false

M General questions

M.1: ◯ A ◯ B ◯ C ◯ D
M.2.a: ◯ true ◯ false
M.2.b: ◯ true ◯ false
M.2.c: ◯ true ◯ false
M.2.d: ◯ true ◯ false

Question M.3:

Multiple	**Unit**
10^6	megabyte
10^9	gigabyte
10^{12}	terabyte
10^{15}	
10^{18}	
10^{21}	
10^{24}	

M.4.a: ◯ true ◯ false
M.4.b: ◯ true ◯ false
M.4.c: ◯ true ◯ false
M.4.d: ◯ true ◯ false
M.5: ◯ A ◯ B ◯ C ◯ D
M.6: ◯ A ◯ B ◯ C ◯ D
M.7: ◯ A ◯ B ◯ C ◯ D

Question M.8:

M.9: ◯ A ◯ B ◯ C ◯ D
M.10.a: ◯ true ◯ false
M.10.b: ◯ true ◯ false
M.10.c: ◯ true ◯ false
M.10.d: ◯ true ◯ false
M.10.e: ◯ true ◯ false
M.10.f: ◯ true ◯ false