CS257 Final Fall 2020

Name : Sida Zhong
StudID: 013931476

Instructions:

1. This final is due 11:59pm PST, Dec 16, 2020.
2. To complete the final, print it out, fill in your answers on the final, and scan it back (or take pictures of the pages and make a pdf) into a file Final.pdf where the total size is less than 10MB.
3. If you don't have a printer, copy and paste each problem into a word processor document. Then after each problem write your solution. Make a less than 10MB Final.pdf file of the result and submit that.
4. Use the same submit mechanism as for the homeworks to submit your completed final.
5. Each problem on this final is worth the same amount (3pts).
6. If you have a question on the interpretation of a problem on the final, you can email me at chris@pollett.org.
7. Due to the coronavirus this is an open book, open internet final.

    - What that means is that you can consult any static (on the order of static for weeks) source of information related to the final material.

    - You cannot directly or indirectly ask another person how to do any problem off the final.

    - To receive credit on problems that make use of your personal information, you need to have correctly filled in that personal information.

    - When you submit your completed final, you are asserting all of the work in the final is your own.

1. Give an example situation framed in terms of a database where: (a) you would use a bitmap index over a Bloom filter (0.5 example, 1pt why example works), (b) you would use a Bloom Filter over a Bitmap index (0.5 example, 1pt why example works).

(a) Suitable bitmap database example:

ID	GENDER	MARRIED
1	femal	no
2	femal	yes
3	male	yes
4	femal	yes
5	femal	yes

Bitmap example:

ID	FEMALE	MARRIED
1	1	0
2	1	1
3	0	1
4	1	1
5	1	1

Bitmap FEMALE: 11011
Bitmap MARRIED: 01111

Explanation:

When the database value is similar to a binary-valued attribute, bitmap index is more appropriate. Bitmap is suitable for processing low distinctive values, and the result can be compressed into binary code, so that addition is very fast.

(b) suitable Bloom Filter database example:

ID	ADDRESS	NAME
1	Mountain View, CA	Alice
2	47900 Bayside Pkwy	Bob
3	48350 Fremont Blvd	Micky
4	47900 Bayside Pkwy	Minne
5	329 Norwich Ave	Sida

Bloom Filter example:

size: "For a 1% error rate, we need about 9.6 bits/key" * 5keys = 48 bits
hash function number: log2(0.01) = 7
Bloom filter : 011100101100110011110010111011……. (48)

Explanation:

When the value of the database is very unique, it cannot be expressed in binary, so bitmap is not suitable. Bloom Filter is suitable for processing high distinctive values. Generate Bloom Filter size and hash function according to data size. When new data is to be inserted, hash the value and convert it into binary code. If all the newly written binary code conflicts with the Bloom Filter binary code, it is a positive false.

2. Explain and give a concrete example (involving a database of your childhood toys) of how to create a database and a collection in MongoDB (1pt). Give commands to insert several items into this collection, and give an example of querying these items and returning the result (1pt). Explain how map reduce aggregation can be done in MongoDB (1pt).

(a) Concrete example of toys:

ID	NAME	PRICE	ATTRIBUTE
1	yo-yo	10	{“color”:”red”,”weight”:1bl}
2	transformers	40	{“color”:”red”,”weight”:2bl}
3	musicBox	80	{“color”:yellow,”weight”:2bl}

Creating database:

```
use toysDB
```

Creating collection:

```
db.createCollection("toysCollection")
```

(b) Insert commands:

```
db.toysCollection.insert([
        {
                NAME: "yo-yo",
                PRICE: "10",
                ATTRIBUTE: {“color”:”red”,”weight”:1bl}
        },
        {
                NAME: "transformers",
                PRICE: "40",
                ATTRIBUTE: {“color”:”red”,”weight”:2bl}
        },
        {
                NAME: "musicBox",
                PRICE: "80",
                ATTRIBUTE: {“yellow”:”red”,”weight”:2bl}
        }
])
```

Querying items:

```
db.toysCollection.findOne({NAME: "yo-yo"})
```

Returning result:

```
{
        "_id" : ObjectId("5dd6542170fb13eec3963bf0"),
        "NAME" : "yo-yo",
```

```
        "PRICE" : "10",
        ATTRIBUTE: {“color”:”red”,”weight”:1bl}
}
```

(c) map reduce explanation

Mapreduce is very suitable for processing embedded documents. It consists of two parts, the first part is map function, its purpose is to filter and sort the qualified data set, The second part is reduce function, which summarizes the searched results. For example, the “attribute” column stores some embedded documents. If we want to find all red color toys, and sort the toys in reverse price order, we need to use mapreduce.

Reduce function:

```
var map = function() {
        if(this.ATTRIBUTE.color==”red”){
                emit(this._id, this.PRICE);
        }
};

var reduce = function(_id, price) {
        return price.reverse();
};
```

result:

```
{
        "_id" : ObjectId("5dd6542170fb13eec3963bf1"),
        "NAME" : "transformers",
        "PRICE" : "40",
        ATTRIBUTE: {“color”:”red”,”weight”:2bl}
},
{
        "_id" : ObjectId("5dd6542170fb13eec3963bf0"),
        "NAME" : "yo-yo",
        "PRICE" : "10",
        ATTRIBUTE: {“color”:”red”,”weight”:1bl}
}
```

3. Suppose R had 2,000,000 tuples and 8 fit into a block of (rightmost digit of your id +1)*1024 bytes. (a) How many blocks and bytes does R take to store? (b) If the key is 16 bytes long and the record pointer 8 bytes long, approximately how many index records can fit in a block? (c) If we have a sparse index on R, how many blocks and bytes would the index file take?

(a) How many blocks does R take to store?
2,000,000 tuples / 8 fit = 250000 blocks

How many bytes does R take to store?
Block size: (rightmost digit of studID: 6) + 1 * 1024 = 7168 bytes
250000 blocks * 7168 bytes = 1792000000 bytes

(b) How many index records can fit in a block?
7168 bytes / (16 byte key + 8 byte pointer) = 298.66 = 299 index records per block

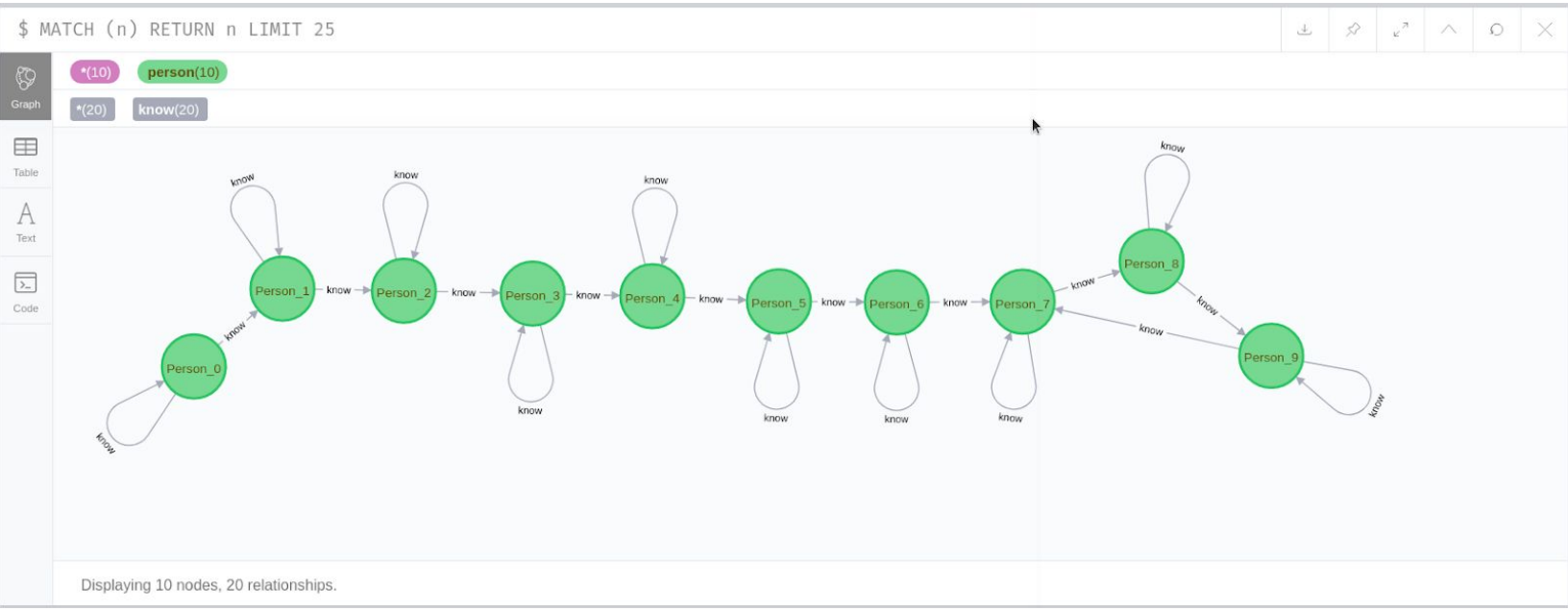
(c) Sparse index on R, how many blocks and bytes would the index file take?
sparse index is one index entry per block
250000 blocks / 299 indexes = 836.12 blocks = 837 blocks

4. Consider 10 people: Person 0, Person 1, ..., Person 9. For i=0,..., 8, Person {i} Knows Person {i+1} and Person {i} Knows Person {i} (they know themselves). Let j be the next-to-rightmost digit of your student id. We also have Person 9 Knows Person {j}. Show the CYPHER commands needed to create this graph in Neo4j (1pt). Express as a CYPHER query, everyone known by at least two people (1pt). Express as a CYPHER query everyone who knows someone who knows Person {k} where k is the last digit of your student id (1pt).

Show the CYPHER commands needed to create this graph in Neo4j
j = next-to-rightmost digit studID = 7

```
CREATE (n:person { name: "Person_0"})
CREATE (n:person { name: "Person_1"})
CREATE (n:person { name: "Person_2"})
CREATE (n:person { name: "Person_3"})
CREATE (n:person { name: "Person_4"})
CREATE (n:person { name: "Person_5"})
CREATE (n:person { name: "Person_6"})
CREATE (n:person { name: "Person_7"})
CREATE (n:person { name: "Person_8"})
CREATE (n:person { name: "Person_9"})
MATCH (self:person) WHERE self.name = 'Person_0' CREATE (self)-[r:know]->(self)
MATCH (self:person) WHERE self.name = 'Person_1' CREATE (self)-[r:know]->(self)
MATCH (self:person) WHERE self.name = 'Person_2' CREATE (self)-[r:know]->(self)
MATCH (self:person) WHERE self.name = 'Person_3' CREATE (self)-[r:know]->(self)
MATCH (self:person) WHERE self.name = 'Person_4' CREATE (self)-[r:know]->(self)
MATCH (self:person) WHERE self.name = 'Person_5' CREATE (self)-[r:know]->(self)
MATCH (self:person) WHERE self.name = 'Person_6' CREATE (self)-[r:know]->(self)
MATCH (self:person) WHERE self.name = 'Person_7' CREATE (self)-[r:know]->(self)
MATCH (self:person) WHERE self.name = 'Person_8' CREATE (self)-[r:know]->(self)
MATCH (self:person) WHERE self.name = 'Person_9' CREATE (self)-[r:know]->(self)
MATCH (first:person),(second:person) WHERE first.name = 'Person_0' AND second.name =
'Person_1' CREATE (first)-[r:know]->(second)
MATCH (first:person),(second:person) WHERE first.name = 'Person_1' AND second.name =
'Person_2' CREATE (first)-[r:know]->(second)
MATCH (first:person),(second:person) WHERE first.name = 'Person_2' AND second.name =
'Person_3' CREATE (first)-[r:know]->(second)
MATCH (first:person),(second:person) WHERE first.name = 'Person_3' AND second.name =
'Person_4' CREATE (first)-[r:know]->(second)
MATCH (first:person),(second:person) WHERE first.name = 'Person_4' AND second.name =
'Person_5' CREATE (first)-[r:know]->(second)
MATCH (first:person),(second:person) WHERE first.name = 'Person_5' AND second.name =
'Person_6' CREATE (first)-[r:know]->(second)
MATCH (first:person),(second:person) WHERE first.name = 'Person_6' AND second.name =
'Person_7' CREATE (first)-[r:know]->(second)
```

MATCH (first:person),(second:person) WHERE first.name = 'Person_7' AND second.name = 'Person_8' CREATE (first)-[r:know]->(second)
MATCH (first:person),(second:person) WHERE first.name = 'Person_8' AND second.name = 'Person_9' CREATE (first)-[r:know]->(second)
MATCH (first:person),(second:person) WHERE first.name = 'Person_9' AND second.name = 'Person_7' CREATE (first)-[r:know]->(second)



Express as a CYPHER query, everyone known by at least two people
Query:
MATCH (p1:person)-[:know]->(p2:person)
WITH p1,count(p2) as count
WHERE count > 1
RETURN p1

Result:
Person 0,1,2,3,4,5,6,7,8,9

Express as a CYPHER query everyone who knows someone who knows Person {k} where k is the last digit of your student id
Query:
MATCH (p1:person)-[:know]->(p2:person), (p2:person)-[:know]->(p3:person)
WHERE p3.name = "Person_7"
RETURN p1

Result:
Person 5,6,8,9

5. Explain and give an example using your name of how to use XQUERY FLWOR expressions to (a) return the results of computing an XPath expression where a salary attribute of an employee tag is greater than the year you were born in as specified in a LET clause (1pt), (b) format the results of computing a query in <answer> tags (1pt), (c) compute the join on some attributes of two XML documents (1pt).

Sida zhong birthday 1989.

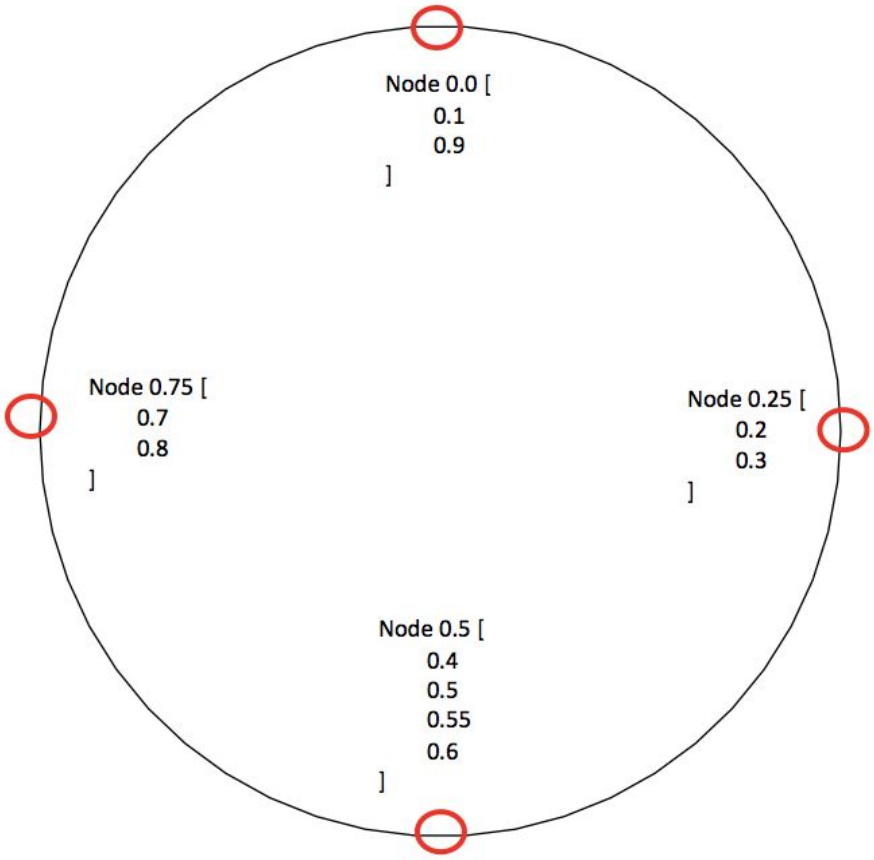
The example shows two xml files, employee.xml and college.xml. Using FLWOR to get the name tag in employee.xml, which corresponds to the email tag in college.xml. The condition is that all employee salaries are greater than 1989.

```
//employee.xml
<?xml version="1.0" encoding="UTF-8"?>
<employee salary="1980">
        <name ID="1">Sida</name>
</employee>
<employee salary="2000">
        <name ID="2">Chris</name>
</employee>

//college.xml
<?xml version="1.0" encoding="UTF-8"?>
<college>
        <email ID="1">sida9567@gmail.com</email>
</college>
<college>
        <email ID="2">chris@pollett.org</email>
</college>


let $salary := 1989
for $employee in doc("employee.xml")//employee
for $college in doc("college.xml")//college
where $employee/@salary > $salary and $employee/name/@ID=$college/email/@ID
return <answer> {$employee/name, $college/email} </answer>
```

6. Explain and give an example of the following concepts: (a) consistent hashing (1pt), (b) stabilization (as related to key value stores) (1pt), (c) gossiping (as related to key value stores) (1pt).

(a) consistent hashing

Consistent hashing would be like a set of nodes that have a ring topology, the purpose is to solve remapping nodes problem.



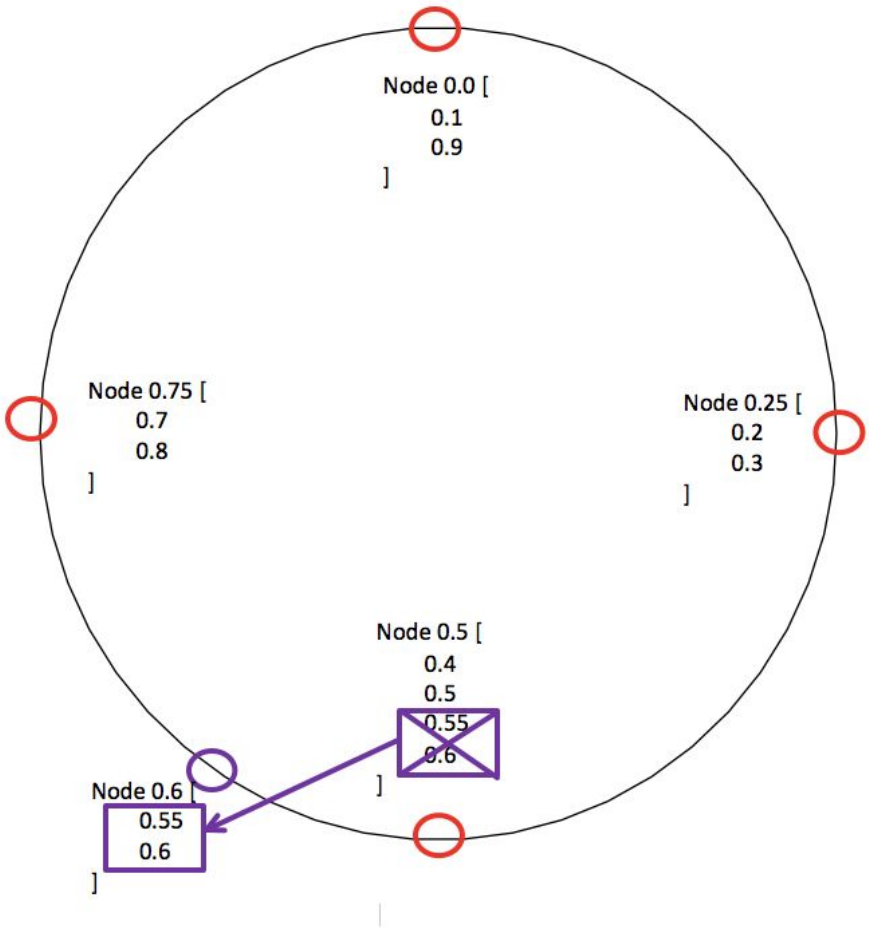
place four nodes at the N,S,E,W points of a circle.
0.0 0.25 0.50 0.75

generate random 10 keys
0.1 0.2 0.3 0.4 0.5
0.55 0.6 0.7 0.8 0.9

put the keys into the circle
0.0 => [0.1,0.9]
0.25 => [0.2,0.3]
0.50 => [0.4,0.5,0.55,0.6]
0.75 => [0.7,0.8]

(a) stabilization

The repartition of key-values over nodes if the number of nodes changes is called stabilization. consistent would minimize the number of fluctuation in the hash-node mapping, which will help with the stabilization period.

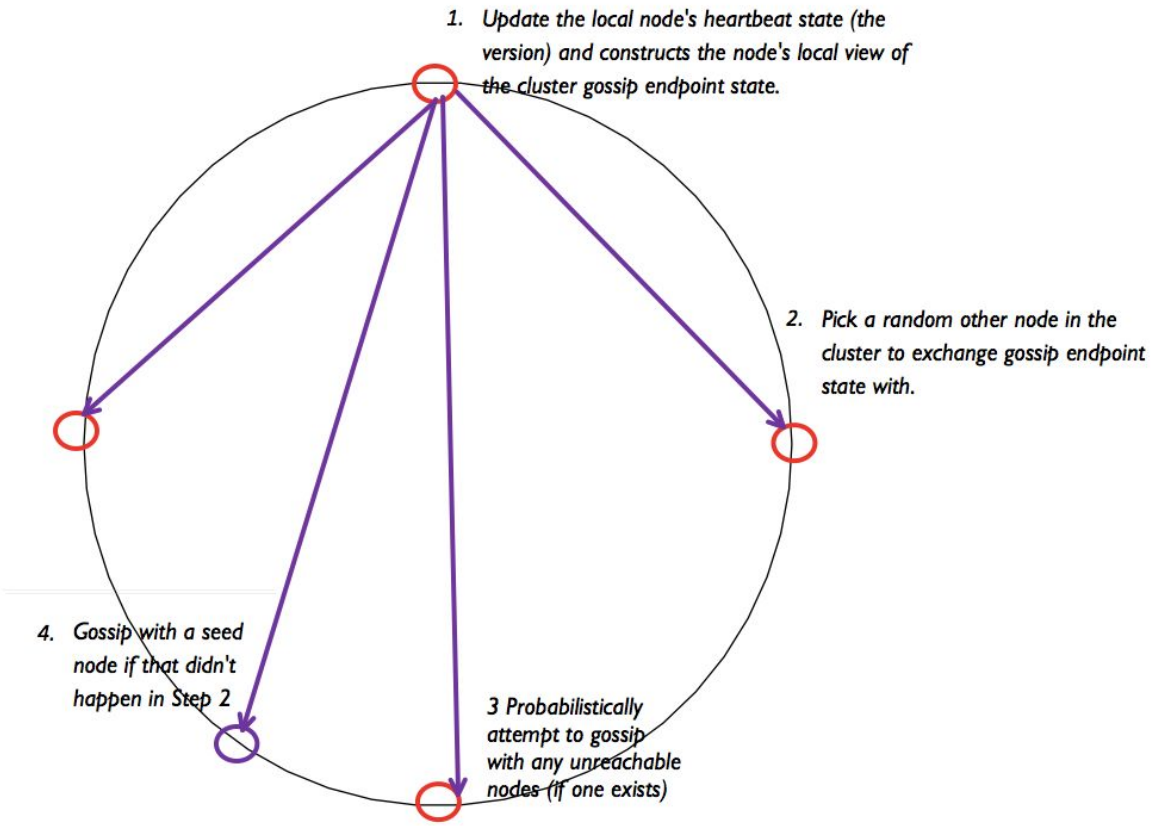


Pick a random new node location (0.60) and add it to the ring.
0.0 => [0.1,0.9]
0.25 => [0.2,0.3]
0.50 => [0.4,0.5]
0.60 => [0.55,0.6]
0.75 => [0.7,0.8]

Efficiency:
So if a new node is added roughly k / (n+1) keys will be closest to the new node and need to be remapped. This is much smaller than k−n keys.
(k/n+1) = 10/(4+1) = count moved key([0.55,0.6]) = 2 VS (k-n) = 10 - 4 = 6
Conclusion: 2 < 6 remapped keys much smaller than (k-n)

(a) gossiping

The gossiping is a periodic task that constantly checks whether each node is communicating with several other nodes in the cluster. The purpose is to allow all nodes on the cluster to be aware of the overall state of the cluster. thereby to prevent crashes.



7. For the Inmon 1996 definition of data warehouse given in class, for each of the parts of the definition, give an explanation and concrete example (1pt). Give a SQL query on a data warehouse involving the cube operator that could be used by management in a decision-making process (say why) (1pt). Briefly explain what a star schema is with regard to data warehouses (1pt).

for each of the parts of the definition, give an explanation and concrete example (1pt).
Subject-oriented: The data in the data warehouse can be analyzed for a particular subject area, which may be specifically for the marketing department or the advertising department.

Integrated: A company has many systems such as a sales system, a marketing system, a customer system, and a storage system. These systems have different definitions of user name. Some are firstname lastname, some are given name, middle name, and some have only a single field called username. These data in different formats should be integrated into the same format in the data warehouse.

time-variant: A user changed his phone, address, and email a few months ago, but the data warehouse will keep all the historical contact records of this user.

nonvolatile collection: The data in the data warehouse can only be read, and history records cannot be tampered with. If a user has a bad credit history, this record will always be kept in the data warehouse for analysis.

Give a SQL query on a data warehouse involving the cube operator that could be used by management in a decision-making process (say why) (1pt).

table: world_population

CONTINENT	COUNTRY	CITY	SALES
Asia	China	Beijing	21M
Asia	China	Shanghai	24M
Asia	Japan	Tokyo	9M
Europe	Franch	Paris	2M
Europe	German	Berlin	3M
America	USA	San Jose	1M

query:

```
select CONTINENT, COUNTRY, CITY, sum(SALES)
from world_population
group by  cube (CONTINENT, COUNTRY, CITY)
```

Cube operation will give the combination of all fields after the group, if there are 3 fields, it will give 7 combinations. Managers can do data analysis based on different combinations.
CONTINENT
CONTINENT + COUNTRY
CONTINENT + CITY
CONTINENT + COUNTRY + CITY
COUNTRY
COUNTRY + CITY
CITY

Briefly explain what a star schema is with regard to data warehouses (1pt).
The star schema consists of two important parts. The first one is called the dimension table. A star schema has many dimension tables. The second part is called the fact table. The fact table is composed of many foreign keys of the dimension table. The fact table also has its own dependent attributes. Star schema is a type of data warehouse schema, which can integrate and analyze data from multiple dimension tables by foreign keys.

8. Modify the Hadoop Map Reduce job from class (you can cut and paste that code as your starting point) so that it computes for each term the number of the documents that have more than the first non zero digit of your student id occurrences of that term (1pt map, 1pt reduce). Explain how to compile and run your program (1pt).
first non zero digit of studID=1

map

```
public void map(Object key, Text value, Context context)
throws IOException, InterruptedException
    {
       // normalize document case, get rid of non word chars
       String document = value.toString().toLowerCase().replaceAll("[^a-z\\s]", "");
       String[] words = document.split(" ");
       for (String word : words) {
          Text textWord = new Text(word);
          IntWritable one = new IntWritable(1);
          context.write(textWord, one);
       }
    }
```

reduce

```
public void reduce(Text key, Iterable<IntWritable> values,
Context context) throws IOException, InterruptedException
    {
       int sum = 0;
       IntWritable result = new IntWritable();
       for (IntWritable val: values) {
                sum += val.get();
       }
       // documents that have more than 1 terms
       int max_number_terms = 1;
       if(sum > max_number_terms){
                result.set(sum);
                context.write(key, result);
       }
    }
```

compile

```
javac -classpath `yarn classpath` -d . terms.java
jar -cf terms.jar terms.class 'terms$map.class' 'terms$reduce.class'
hadoop jar terms.jar WordCount ~/test.txt ~/output
hadoop fs -cat ~/output/part-r-00000
```

9. What are the four steps in the Total Data Quality Management cycle? (1pt) Make up an example scenario involving airline data. Walk through the four steps of TDQM in terms of your airline scenario (1pt). Briefly explain the orchestration pattern used in process management (1pt).

What are the four steps in the Total Data Quality Management cycle?
Definition: quality dimensions
Measurement: percentages of data of given quality across each dimension
Analysis: causes of lack of data quality
Improvement: specific actions such as random input audits to be taken to fix lack of data quality

Make up an example scenario involving airline data. Walk through the four steps of TDQM in terms of your airline scenario (1pt).
Background: The luggage stored in the cabin of the airplane is limited. Generally, airlines stipulate that each person can only bring two luggages, extra money will be charged for more than two luggages. However, in many cases, the plane is not full of passengers, the size of each luggage is different. So in many cases, although someone has charged for extra luggage, there is still room to store more luggage in the cabin of the aircraft.

Definition: “carry on luggages” Each person can only bring exactly two luggages.

Measurement: In one hundred flight records, there are 80% of cases where the luggage box in the airplane cabin is not full. However 20% of passengers charged for extra luggage.

Analysis: The plane is not full of passengers, The size of the luggage is different.

Improvement: The number of boxes that each person can carry is dynamically adjusted according to the total number of passengers. Each passenger needs to take a picture of the luggage, and then the machine can estimate the size of the luggage.

Briefly explain the orchestration pattern used in process management (1pt).
The orchestrator pattern is responsible for coordinating the interaction among different services and subprocesses, such as invoking, combining, and receiving updates of process and services.

For example, a shopping company only sells products in the customer system, but the company needs another company’s inventory system to deliver goods, and also needs the supplier company's system to replenish the goods. At this time, an orchestration pattern is needed to coordinate these three different systems. To ensure that the customer’s shopping process is not interrupted.

10. Briefly explain the difference between predictive and descriptive analytics (1pt). Give an example of a technique connected with each (1pt) and explain how that technique works (1pt).

Briefly explain the difference between predictive and descriptive analytics (1pt).
Descriptive analytics is more focused on the past, what already happened in the past. It usually provides some real-time data based on past data.

Predictive analytics is more focused on the future, what will happen in the future. It usually provides data trends and patterns.

Give an example of a technique connected with each (1pt)
Descriptive analytics: Data Virtualization.

Predictive analytics: Machine learning.

explain how that technique works (1pt).
Data Virtualization for descriptive analytics: Obtains and integrates information source data from multiple systems, Using hierarchical clustering and k-means clustering to provide a Data Virtualization, and provides graphs, charts, reports to customers.

Machine learning for predictive analytics: According to the customers credit history, income and stocks, use machine learning to identify loan risks and opportunities. Provide customers with a financially acceptable proposal.