NOM KOCET

Prénom Kim

Promo 2020

Date 14/12/18

KOCET Kim
M1-2018-TD BD

17,

MATIÈRE Data Structuring and NO SQL DB.

1)

index	element	value
0	tmin	27
1	tmax	32
2	tmin	54
3	tmax	10

This is an un-tidy data because multiple variables are stored in one columns (here , tmin and tmax should be columns).

index	F 0-14	F 14-25	M 0-14	M 14-25
0				
1				
2				
3				

Considering the above dataset. The columns are composed of multiple variables (which are the sex (M for masculin, F for feminin) and the range of age (0-14 years old, and 14 - 25 years old).

index	< 10 $	10 - 20 $	< 20 $
0	3	5	7
1	4	6	8.

This is an un-tidy data because the columns headers are values, not names.

index	name	income
0	Vincent	20
1	Franck	30

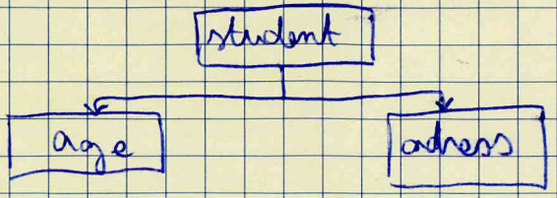
index	name	age
0	Vincent	31
1	Marie	40

This is an un-tidy data because one observational unit is stored in multiple tables (the name)

2) A stands for atomicity. One transaction either takes effect, or is roll back.
C stands for consistency. The transaction brings the system from one valid state to another.
I stands for isolated. The changes are not visible before the end of the transaction.
D stands for durability. The results of the transaction are memorized by the system.

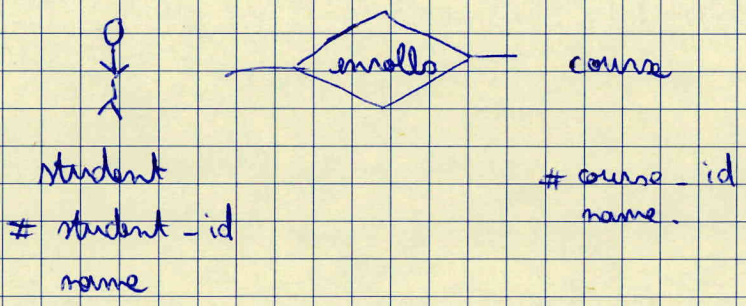
3) XML queries are based on a tree-schema.

```
< student >
    < age > < / age >
    < adress > < / adress >
< / student >
```



4) MongoDB is a document-based schema.
The database in SQL is also called database in MongoDB.
The table is called a collection.
The row is called a document.

5) To model a foreign key in MongoDB, we can add an attribute in the collection which would be an array.
This array would contain a sub-collection.
For example, suppose we want to represent the relation "one student enrolls in one course".



The collection "student" would be composed of the items "student-id", "student name" and an array containing the id and the name of the course.

```
{ '_id' : ...
  'student_name' : ...
  'course' : [ 'id course' : ...
               'name course' : ...
             ]
}
```

In MongoDB we can also merge two datasets to create an other dataset, which would contain the keys of the two datasets. We use the aggregation pipeline $ lookup to do so.

6) a. Suppose the database name is 'db' and the collection name is 'col'.

```
db.col.find({"keywords": {$in: ["python"]}})
```

b. `db.col.find({"authorname": 'A.BC'})`

c.
```
db.col.update(
        {"author": "DEF"},
        {"time": {$inC: 100}})
```

7) The syntax of XML and Json are not the same.
XML works with braquets. For example, that could look like this:

```
<id> 100 </id>
<name> Vincent </name>
```

The schema of XML is like a tree.

For JSON, the data is organized in a form of a list of document. For example, that could look like this:

NOM KOCET

Prénom Kim

Promo 2020

Date 14/12/18

MATIÈRE Data Structuring and No SQL DB.

```
{   id : "01",
    name : "Vincent"
}
{   id : "02",
    name : "Franck"
}
```

SO WHAT DOES IT CHANGE?

8) Suppose we have a user data which gives us the name of the user, the movies he watched and the rate he gave to it.

name	movie watched	rate
elsa	a	10
elsa	b	7
wael	a	8

id	user name
1	Elsa
2	wael

And an actor data which gives the name, and the movies he participated in.

id	actor name	movies
1	Pitt	a
2	DiCaprio	a
3	Rambo	b

we could merge these two datasets using the column "movie name" to create an other data.
For example, we could perform some aggregation to get the average rate of a movie, or the number of users that watched it, or the number of actor that participated in ...

id	movie name	avg_rate	actor in ...
0	a	9	
1	b	7	
2	c	null	

we could also create an other data collection that gives the number of movies watched by an user.

id	name	number of movies watched
0	elsa	2
1	wael	1

This could bring some business questions, for example :

* who is the most active user ?
* what is the most popular movie ?
* does the cast have an impact on the popularity of a movie ? (highest average rate)

9) We could use a NoSQL database for example, since the size of the data would be very large.
We would also collect relevant informations about the article such as the main topic (physic, mathematics ...), the date of creation, the date of last update, the author name.

article id	topic	date	author
0	maths	01-01-2018	Tom
1	physics	01-03-2017	Tom
2	food	04-05-2018	David

we would also ensure that we get a tidy data.
The content of the article would be very heavy, so we would create an other data set containing the article's text, the images... But we store those items in a different data set as the previous one so that we don't need to load all the text, the images, ... to get the relevant informations about the article.

article id	text	image_url
0	"this is a text"	"url"
1	"some text"	"link"
2	"blah"	"link"

MODEL LINKS ?

efrei PARIS
Ecole d'ingénieurs du numérique

Ne rien inscrire dans ce cadre

Prénom ~~XX~~ Kim

Nom KOCET

Promotion 2020

Groupe Big Data

M1

Data structuring and NoSQL databases

ST2DST

DE - 1h45 min

Date Horaire

Sujet proposé par :

Calculatrice autorisée : ☐ **OUI** ☒ **NON**

Documents autorisés : ☐ **OUI** ☒ **NON** **Type de documents :**

Ordinateur portable autorisé : ☐ **OUI** ☒ **NON**

Internet : ☐ **OUI** ☒ **NON**

Traducteur électronique, dictionnaire : ☐ **OUI** ☐ **NON**

Consigne :

Merci de restituer uniquement : **les copies quadrillées à rendre accompagnées de l'annexe**

Rappel :

- Tous les appareils électroniques (téléphones portables, ordinateurs, tablettes, montres connectées ...) doivent être éteints et rangés.
- Il est interdit de communiquer.
- Toute fraude ou tentative de fraude fera l'objet d'un rapport de la part du surveillant et sera sanctionnée par la note zéro, assortie d'une convocation devant le conseil de discipline. Aucune contestation ne sera possible. Tous les documents et supports utilisés frauduleusement devront être remis au surveillant.
- Aucune sortie de la salle d'examen ne sera autorisée avant la moitié de la durée de l'épreuve.

Ne rien inscrire dans ce cadre

1. Provide 4 different examples of « Un-tidy » data (you can sketch them if necessary)
2. Explain the meaning of ACID letters for SQL transactions
3. Provide examples of XML queries and explain what they do
4. What is MongoDB schema hierarchy? (like Database/Table/Row for SQL)
5. How to model a "Foreign Key" in MongoDB? What are different modelling options? For example, how to store hierarchical data, like a list of employees with management hierarchy?
6. For a data

   {'_id': ObjectId('5ba207f488811f06f433e7f7'), 'author': 'ABC', 'author_name': 'A. B C', 'icon_filename': 'icon1.png', 'text': 'Some text', 'time': 1234487, 'keywords': ['test', 'python', 'mongodb'], 'responses': [{'author': 'DEF', 'author_name': 'D. E F', 'icon_filename': 'icon2.png', 'text': 'thanks a lot!'}, {'author': 'ABC', 'author _name': 'A. BC', 'icon_filename': 'icon1.png', 'text': 'cheers!', 'responses': [{'aut hor': 'ABC', 'author_name': 'D. EF', 'text': 'Thanx again'}]}]}

   write the Mongo query to

   a. Fetch only the elements with a keyword 'python'
   b. Fetch only the elements with "author_name" == 'A.BC'
   c. Increase time by 100 for author "DEF"
7. What is a difference between XML and JSON?
8. You have data on
   - Users watching and rating movies
   - Actors playing in the movies

   What are your different options to model these data and store it in a single SQL or NoSQL database (sketch it if necessary)? What databases can you use? Provide examples of "business" questions that are better adapted to each storage/modeling option.
9. You have downloaded a copy of Wikipedia (or of an encyclopedia), how would you store it? Why? What are other options and how organize in this case (SQL, NoSQL, or something else)?