Department of
Computer Engineering
Faculty of Engineering

Kasetsart University

HW#XXX: NoSQL & MongoDB

1) You're creating a database to contain information about students in a class (name and ID), and class projects done in pairs (two students and a project title). Should you use the relational model or MongoDB? Please justify your answer

Ans: Relational databases, because in this scenario it does not need changes in data so relational databases are better.

2) You're creating a database to contain information about students in a class (name and ID), and class projects. Projects may include any combination of students; they have a title and optional additional information such as materials, approvals, and milestones. Should you use the relational model or MongoDB? Please justify your answer

Ans: Due to the adaptability of mongoDB, it is more suitable for this scenario because the data may be added to the database whenever we want.

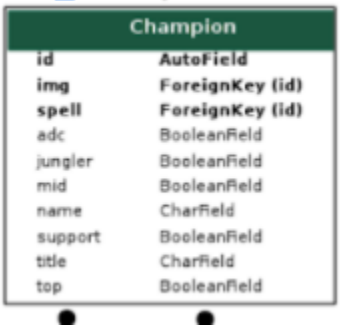
3) You're creating a database to contain a set of sensor measurements from a two-dimensional grid. Each measurement is a time-sequence of readings, and each reading contains ten labeled values. Should you use the relational model or MongoDB? Please justify your answer

Ans: Relational Database, due to two dimensional grid does not need more flexibility.

4) Choose one of the following applications

a. IoT
b. E-commerce
c. Gaming
d. Finance

Propose an appropriate Relational Model or MongoDB database schema



Gaming: MongoDB, with different variations to each champion's abilities I think using MongoDB is more suitable for this kind of application.

5) Create MongoDB database with following information

1) ({"name":"Ramesh","subject":"maths","marks":87})
2) ({"name":"Ramesh","subject":"english","marks":59})
3) ({"name":"Ramesh","subject":"science","marks":77})
4) ({"name":"Rav","subject":"maths","marks":62})
5) ({"name":"Rav","subject":"english","marks":83})
6) ({"name":"Rav","subject":"science","marks":71})
7) ({"name":"Alison","subject":"maths","marks":84})
8) ({"name":"Alison","subject":"english","marks":82})
9) ({"name":"Alison","subject":"science","marks":86})
10) ({"name":"Steve","subject":"maths","marks":81})
11) ({"name":"Steve","subject":"english","marks":89})
12) ({"name":"Steve","subject":"science","marks":77})
13) ({"name":"Jan","subject":"english","marks":0,"reason":"absent"})

Give MongoDB statements (with results) for the following queries

• Find the total marks for each student across all subjects.

```
> db.students.aggregate([{$group:{_id:"$name", "Total Marks":{$sum:"$marks"}}}])
< { _id: 'Alison', 'Total Marks': 252 }
  { _id: 'Rav', 'Total Marks': 216 }
  { _id: 'Ramesh', 'Total Marks': 223 }
  { _id: 'Jan', 'Total Marks': 0 }
  { _id: 'Steve', 'Total Marks': 247 }
```

• Find the maximum marks scored in each subject.

```
> db.students.aggregate([{$group:{_id:"$subject", "Max Marks":{$max:"$marks"}}}])
< { _id: 'english', 'Max Marks': 89 }
  { _id: 'science', 'Max Marks': 86 }
  { _id: 'maths', 'Max Marks': 87 }
```

• Find the minimum marks scored by each student.

```
> db.students.aggregate([{$group:{_id:"$name", "Min Marks":{$min:"$marks"}}}])
< { _id: 'Rav', 'Min Marks': 62 }
  { _id: 'Steve', 'Min Marks': 77 }
  { _id: 'Jan', 'Min Marks': 0 }
  { _id: 'Ramesh', 'Min Marks': 59 }
  { _id: 'Alison', 'Min Marks': 82 }
```

• Find the top two subjects based on average marks.

```
> db.students.aggregate([{$group:{_id:"$subject", "Top two Marks":{$avg:"$marks"}}}])
< { _id: 'english', 'Top two Marks': 62.6 }
  { _id: 'science', 'Top two Marks': 77.75 }
  { _id: 'maths', 'Top two Marks': 78.5 }
```