HW# NoSQL & MongoDB

1) 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. If the sensor measurements have a fixed and well-defined structure, the relational model is more suitable for efficient querying and analysis, while MongoDB is better for a more flexible and dynamic schema design. Since the sensor measurements in this case have a well-defined and fixed structure, the relational model is the better choice.

2) For each of the following applications

a. IoT

b. E-commerce

c. Gaming

d. Finance

Propose an appropriate Relational Model or MongoDB database schema. For each application, clearly justify your choice of database.

Ans.

a. IoT:

A time-series database schema is suitable for an IoT application, as it handles time-stamped data and allows for efficient analysis of such data. This can be implemented in either a Relational Model or MongoDB.

b. E-commerce:

A Relational Model database schema is appropriate for an E-commerce application because it handles structured data well, and allows for efficient querying and data management.

c. Gaming:

For a Gaming application, a NoSQL database schema such as a document-based schema in MongoDB is suitable because it provides more flexibility in schema design and faster data retrieval.

d. Finance:

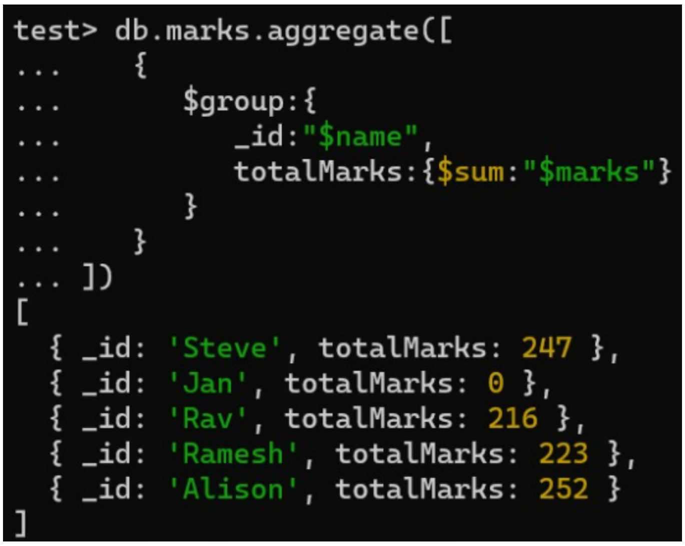
A Relational Model database schema is suitable for a Finance application as it allows for efficient management and querying of structured data while enforcing data integrity constraints.

3) 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"})
```

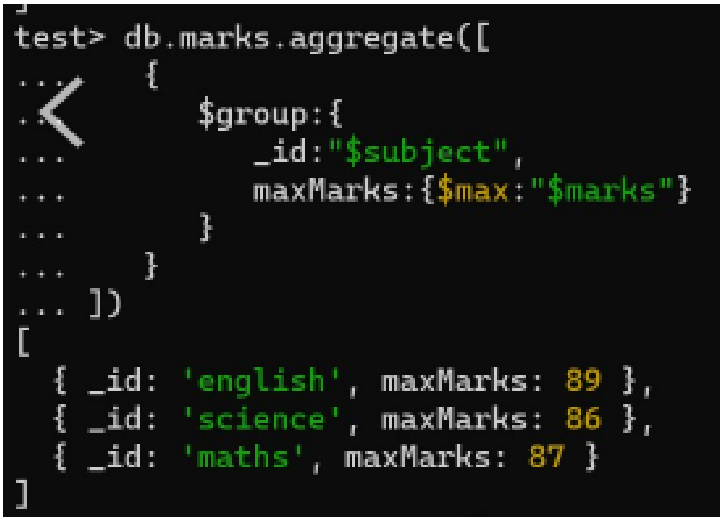
Give MongoDB statements (with results) for the following queries

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



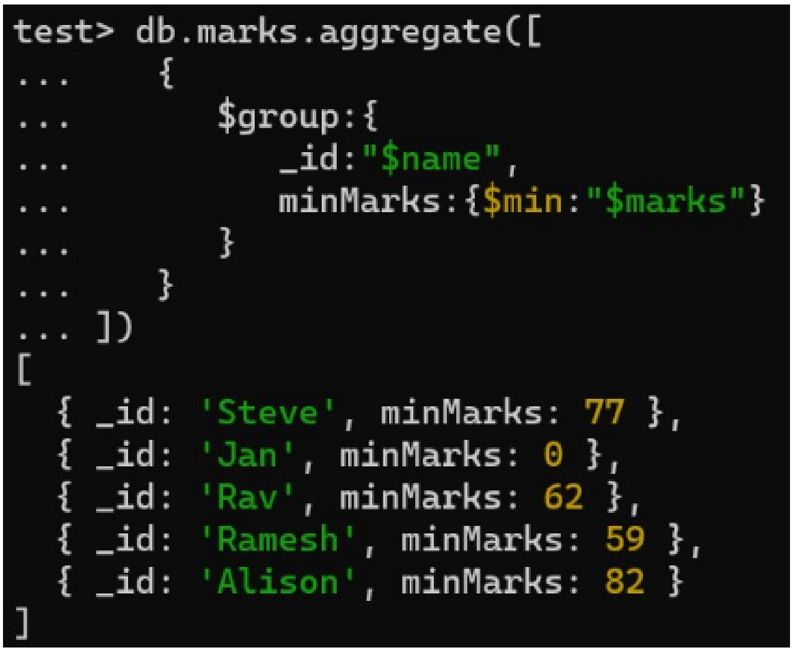
```
test> db.marks.aggregate([
...     {
...         $group:{
...             _id:"$name",
...             totalMarks:{$sum:"$marks"}
...         }
...     }
... ])
[
  { _id: 'Steve', totalMarks: 247 },
  { _id: 'Jan', totalMarks: 0 },
  { _id: 'Rav', totalMarks: 216 },
  { _id: 'Ramesh', totalMarks: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

- Find the maximum marks scored in each subject.



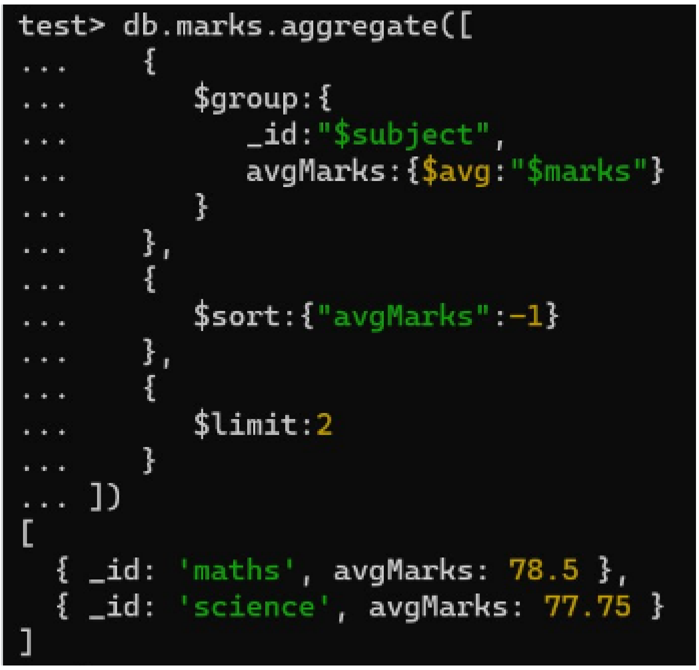
```
test> db.marks.aggregate([
...     {
...         $group:{
...             _id:"$subject",
...             maxMarks:{$max:"$marks"}
...         }
...     }
... ])
[
  { _id: 'english', maxMarks: 89 },
  { _id: 'science', maxMarks: 86 },
  { _id: 'maths', maxMarks: 87 }
]
```

- Find the minimum marks scored by each student.



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

- Find the top two subjects based on average marks.



```
test> db.marks.aggregate([
...    {
...       $group:{
...          _id:"$subject",
...          avgMarks:{$avg:"$marks"}
...       }
...    },
...    {
...       $sort:{"avgMarks":-1}
...    },
...    {
...       $limit:2
...    }
... ])
[
  { _id: 'maths', avgMarks: 78.5 },
  { _id: 'science', avgMarks: 77.75 }
]
```