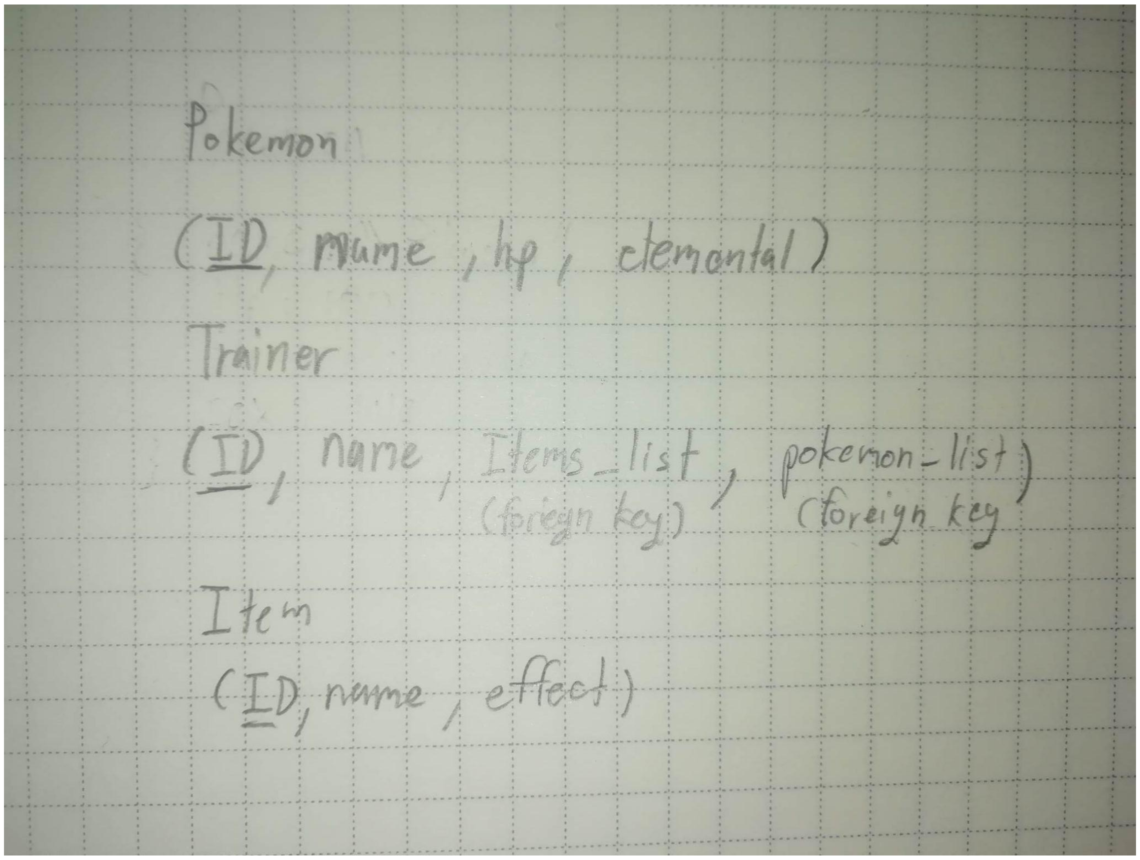
1) They should use a relational model because requirements are already strict and seem to not have any other requirements so a relational model will suit this.

2) MongoDB as you can see from the requirement they contain the word ‘may’ which means they are not confident enough in what data will look like in the database, so mongoDB will suit this since there is no need to predefined schema like relational model does.

3) I prefer mongoDB over relational models if we need to anticipate large volumes of data. While if we had to relate data from different sensors I prefer relational models.

4) Gaming with relational databases to relate pokemons and items to what trainer it belongs to.



Pokemon

(<u>ID</u>, name, hp, elemantal)

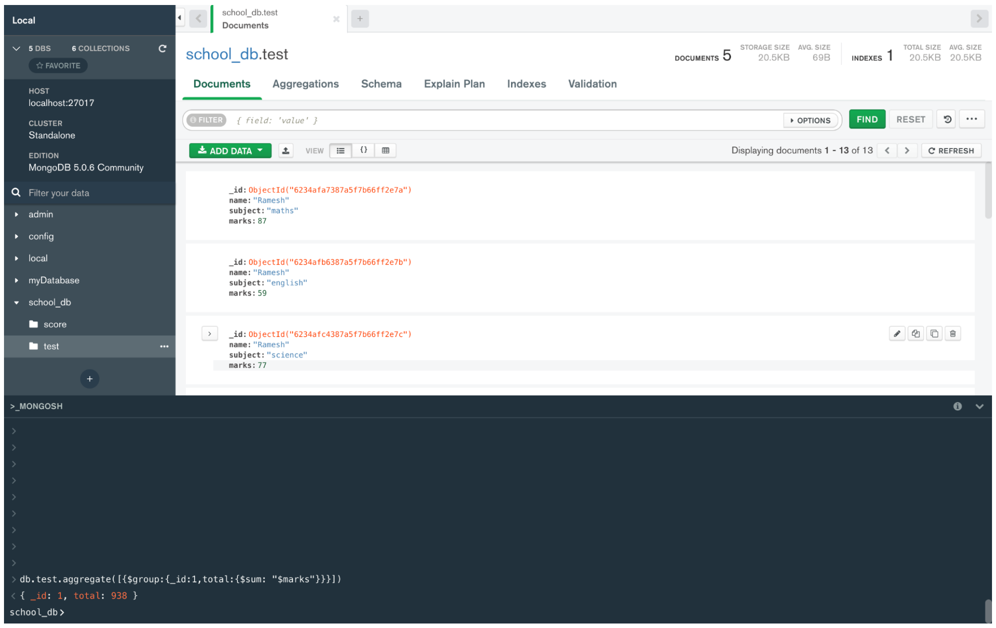
Trainer

(<u>ID</u>, name, Items_list (foreign key), pokemon-list (foreign key))

Item

(<u>ID</u>, name, effect)

5)



HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 5.0.6 Community

school_db.test
Documents

school_db.test

DOCUMENTS 5 STORAGE SIZE 20.5KB AVG. SIZE 69B INDEXES 1 TOTAL SIZE 20.5KB AVG. SIZE 20.5KB

{ field: 'value' }

Displaying documents 1 - 13 of 13

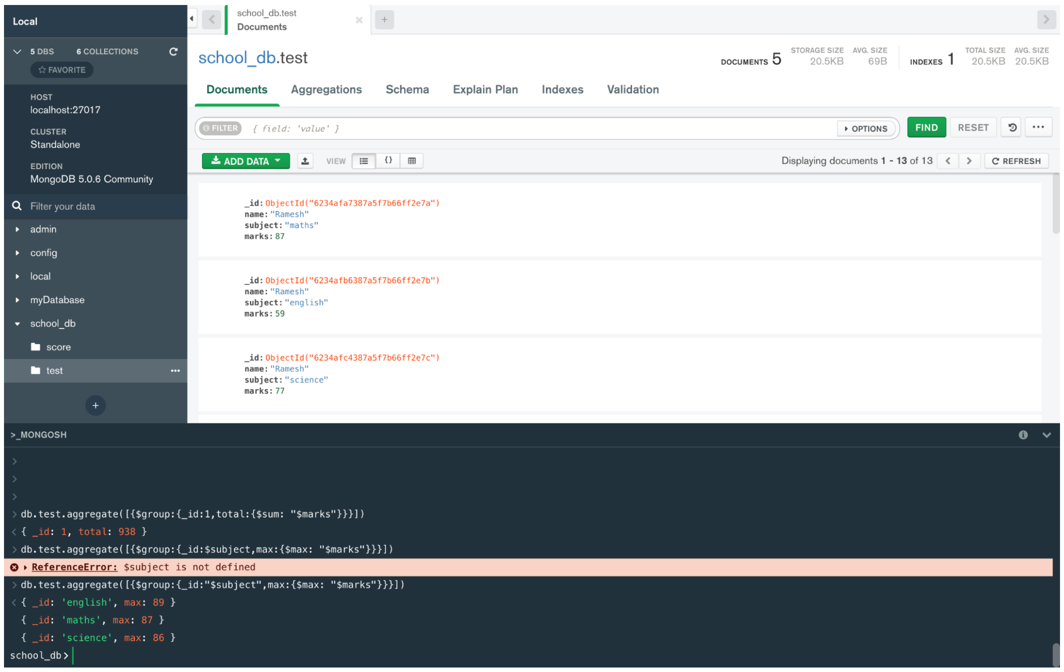
```
_id: ObjectId("6234afa7387a5f7b66ff2e7a")
name: "Ramesh"
subject: "maths"
marks: 87

_id: ObjectId("6234afb6387a5f7b66ff2e7b")
name: "Ramesh"
subject: "english"
marks: 59

_id: ObjectId("6234afc4387a5f7b66ff2e7c")
name: "Ramesh"
subject: "science"
marks: 77
```

```
>
>
>
>
>
>
>
>
>
> db.test.aggregate([{$group:{_id:1,total:{$sum: "$marks"}}}])
< { _id: 1, total: 938 }
school_db>
```

Find the total marks for each student across all subjects.



HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 5.0.6 Community

school_db.test
Documents

school_db.test

DOCUMENTS 5 STORAGE SIZE 20.5KB AVG. SIZE 69B INDEXES 1 TOTAL SIZE 20.5KB AVG. SIZE 20.5KB

{ field: 'value' }

Displaying documents 1 - 13 of 13

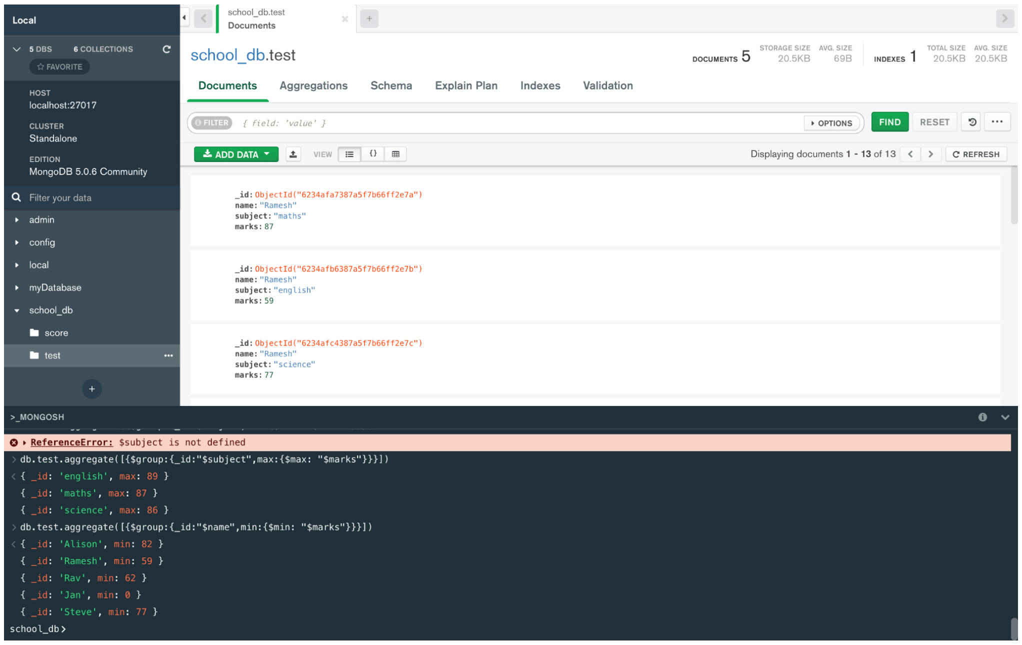
```
_id: ObjectId("6234afa7387a5f7b66ff2e7a")
name: "Ramesh"
subject: "maths"
marks: 87

_id: ObjectId("6234afb6387a5f7b66ff2e7b")
name: "Ramesh"
subject: "english"
marks: 59

_id: ObjectId("6234afc4387a5f7b66ff2e7c")
name: "Ramesh"
subject: "science"
marks: 77
```

```
>
>
>
> db.test.aggregate([{$group:{_id:1,total:{$sum: "$marks"}}}])
< { _id: 1, total: 938 }
> db.test.aggregate([{$group:{_id:$subject,max:{$max: "$marks"}}}])
ReferenceError: $subject is not defined
> db.test.aggregate([{$group:{_id:"$subject",max:{$max: "$marks"}}}])
< { _id: 'english', max: 89 }
  { _id: 'maths', max: 87 }
  { _id: 'science', max: 86 }
school_db>
```

Find the maximum marks scored in each subject.



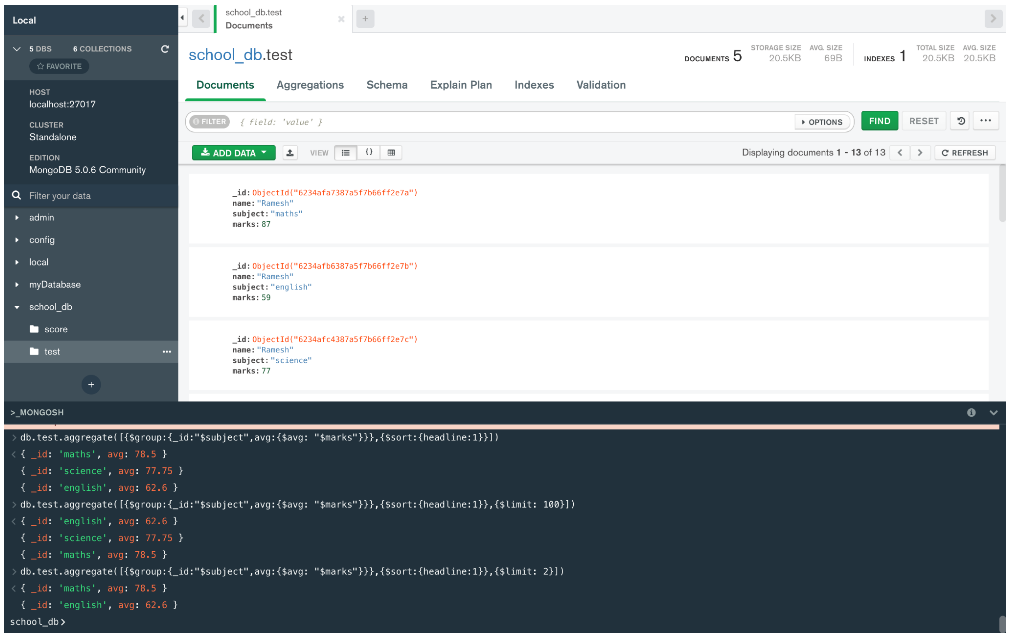
```
_id: ObjectId("6234afa7387a5f7b66ff2e7a")
name: "Ramesh"
subject: "maths"
marks: 87

_id: ObjectId("6234afb6387a5f7b66ff2e7b")
name: "Ramesh"
subject: "english"
marks: 59

_id: ObjectId("6234afc4387a5f7b66ff2e7c")
name: "Ramesh"
subject: "science"
marks: 77
```

```
>_MONGOSH
ReferenceError: $subject is not defined
> db.test.aggregate([{$group:{_id:"$subject",max:{$max: "$marks"}}}])
< { _id: 'english', max: 89 }
  { _id: 'maths', max: 87 }
  { _id: 'science', max: 86 }
> db.test.aggregate([{$group:{_id:"$name",min:{$min: "$marks"}}}])
< { _id: 'Alison', min: 82 }
  { _id: 'Ramesh', min: 59 }
  { _id: 'Rav', min: 62 }
  { _id: 'Jan', min: 0 }
  { _id: 'Steve', min: 77 }
school_db>
```

Find the minimum marks scored by each student.



```
_id: ObjectId("6234afa7387a5f7b66ff2e7a")
name: "Ramesh"
subject: "maths"
marks: 87

_id: ObjectId("6234afb6387a5f7b66ff2e7b")
name: "Ramesh"
subject: "english"
marks: 59

_id: ObjectId("6234afc4387a5f7b66ff2e7c")
name: "Ramesh"
subject: "science"
marks: 77
```

```
>_MONGOSH
> db.test.aggregate([{$group:{_id:"$subject",avg:{$avg: "$marks"}}},{$sort:{headline:1}}])
< { _id: 'maths', avg: 78.5 }
  { _id: 'science', avg: 77.75 }
  { _id: 'english', avg: 62.6 }
> db.test.aggregate([{$group:{_id:"$subject",avg:{$avg: "$marks"}}},{$sort:{headline:1}},{$limit: 100}])
< { _id: 'english', avg: 62.6 }
  { _id: 'science', avg: 77.75 }
  { _id: 'maths', avg: 78.5 }
> db.test.aggregate([{$group:{_id:"$subject",avg:{$avg: "$marks"}}},{$sort:{headline:1}},{$limit: 2}])
< { _id: 'maths', avg: 78.5 }
  { _id: 'english', avg: 62.6 }
school_db>
```

Find the top two subjects based on average marks.