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

- should use relational model because there are a few data.

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

-should use MongoDB because there are combination of data .

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

- should use MongoDB because there is a big scale of data

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 - should use mongoDB because it flexible and can handling data.

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})

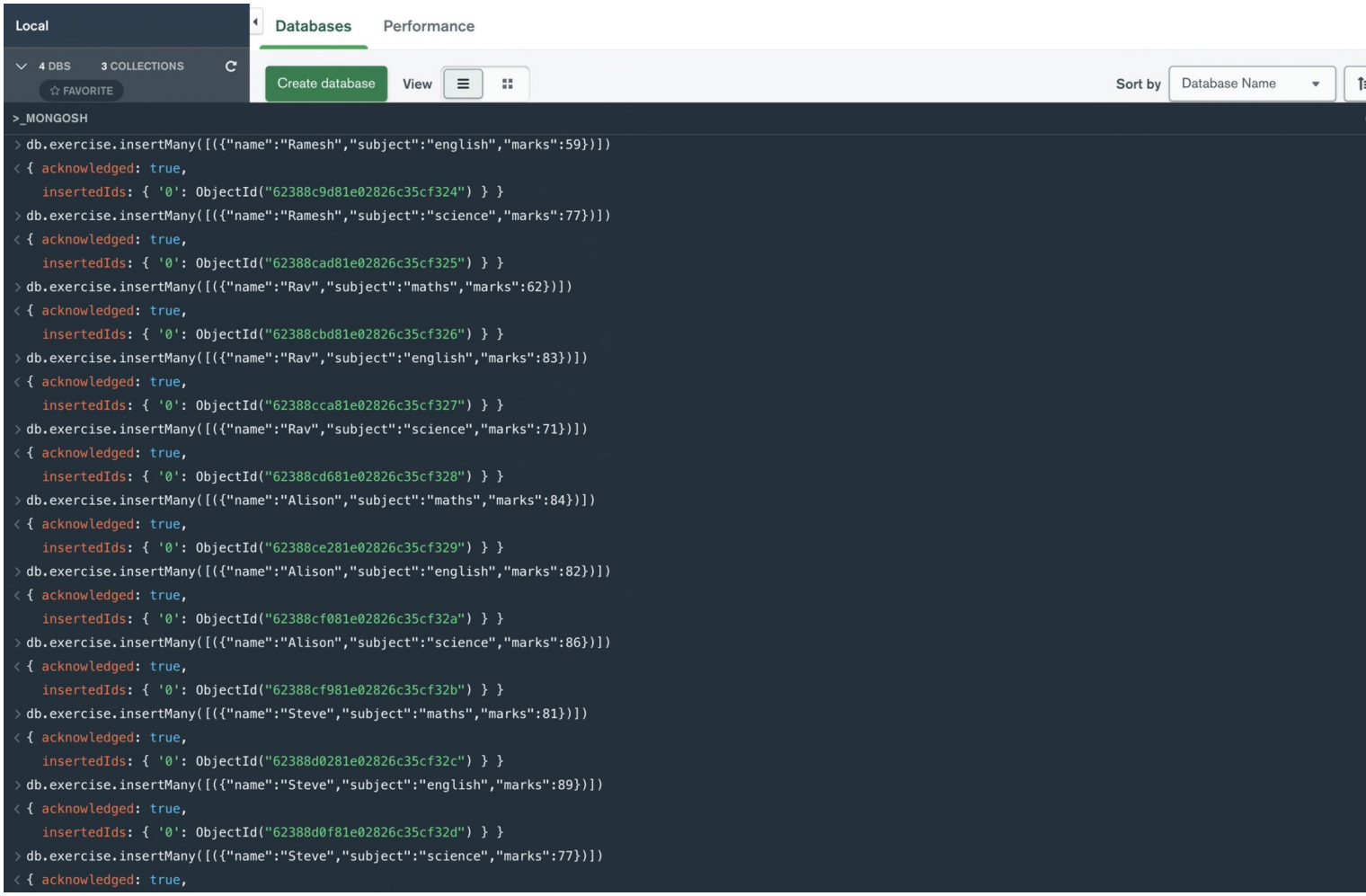
11) ({"name":"Steve","subject":"english","marks":89})

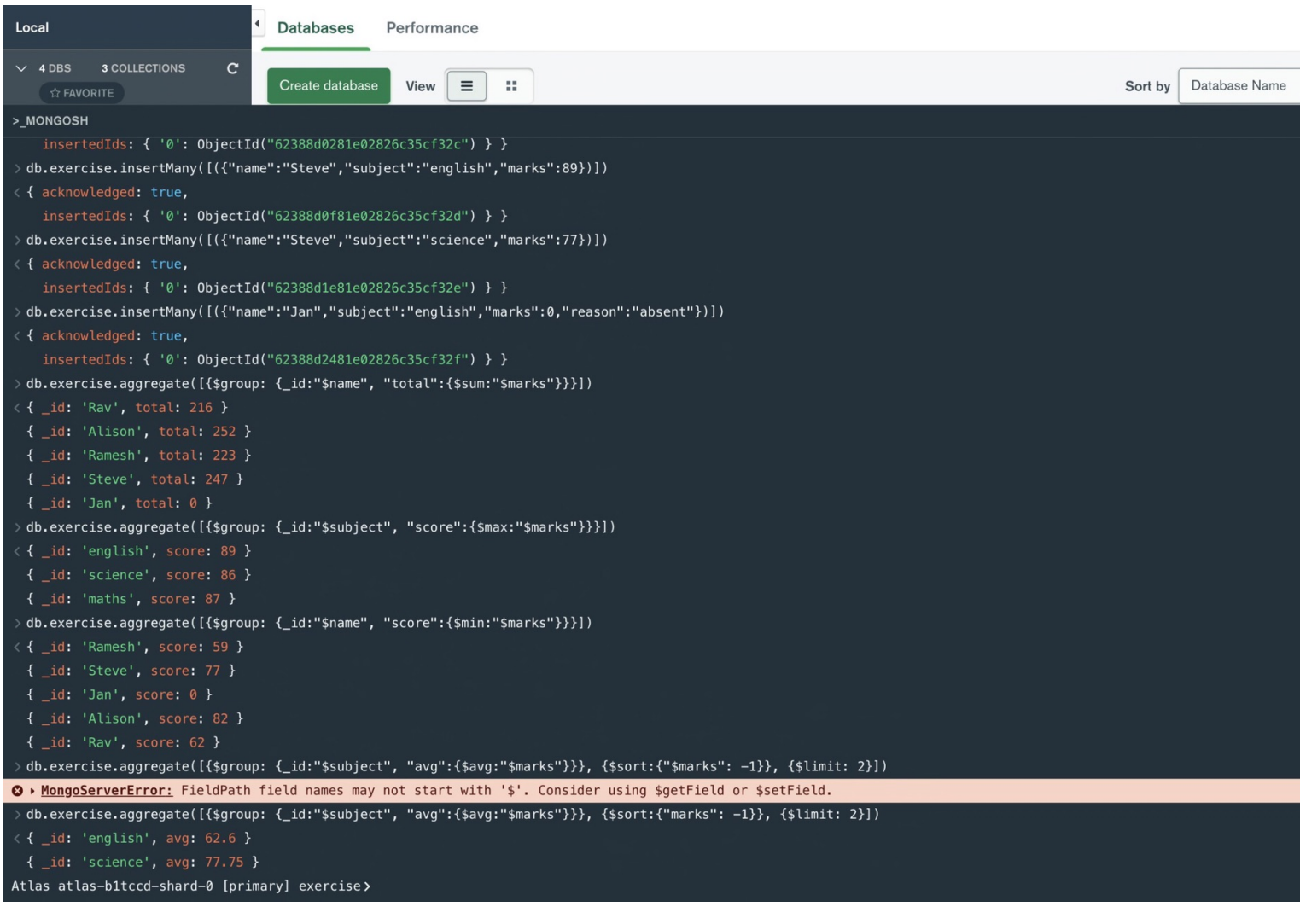
12) ({"name":"Steve","subject":"science","marks":77})

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.
- Find the maximum marks scored in each subject.
- Find the minimum marks scored by each student.
- Find the top two subjects based on average marks.


Local
Databases
Performance
4 DBS
3 COLLECTIONS
FAVORITE
Create database
View
Sort by
Database Name
>_MONGOSH
> db.exercise.insertMany([{"name":"Ramesh","subject":"english","marks":59}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388c9d81e02826c35cf324") } }
> db.exercise.insertMany([{"name":"Ramesh","subject":"science","marks":77}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388cad81e02826c35cf325") } }
> db.exercise.insertMany([{"name":"Rav","subject":"maths","marks":62}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388cbd81e02826c35cf326") } }
> db.exercise.insertMany([{"name":"Rav","subject":"english","marks":83}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388cca81e02826c35cf327") } }
> db.exercise.insertMany([{"name":"Rav","subject":"science","marks":71}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388cd681e02826c35cf328") } }
> db.exercise.insertMany([{"name":"Alison","subject":"maths","marks":84}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388ce281e02826c35cf329") } }
> db.exercise.insertMany([{"name":"Alison","subject":"english","marks":82}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388cf081e02826c35cf32a") } }
> db.exercise.insertMany([{"name":"Alison","subject":"science","marks":86}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388cf981e02826c35cf32b") } }
> db.exercise.insertMany([{"name":"Steve","subject":"maths","marks":81}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388d0281e02826c35cf32c") } }
> db.exercise.insertMany([{"name":"Steve","subject":"english","marks":89}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388d0f81e02826c35cf32d") } }
> db.exercise.insertMany([{"name":"Steve","subject":"science","marks":77}])
< { acknowledged: true,


Local
Databases
Performance
4 DBS
3 COLLECTIONS
FAVORITE
Create database
View
Sort by
Database Name
>_MONGOSH
insertedIds: { '0': ObjectId("62388d0281e02826c35cf32c") } }
> db.exercise.insertMany([{"name":"Steve","subject":"english","marks":89}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388d0f81e02826c35cf32d") } }
> db.exercise.insertMany([{"name":"Steve","subject":"science","marks":77}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388d1e81e02826c35cf32e") } }
> db.exercise.insertMany([{"name":"Jan","subject":"english","marks":0,"reason":"absent"}])
< { acknowledged: true,
insertedIds: { '0': ObjectId("62388d2481e02826c35cf32f") } }
> db.exercise.aggregate([{$group: {_id:"$name", "total":{$sum:"$marks"}}}])
< { _id: 'Rav', total: 216 }
{ _id: 'Alison', total: 252 }
{ _id: 'Ramesh', total: 223 }
{ _id: 'Steve', total: 247 }
{ _id: 'Jan', total: 0 }
> db.exercise.aggregate([{$group: {_id:"$subject", "score":{$max:"$marks"}}}])
< { _id: 'english', score: 89 }
{ _id: 'science', score: 86 }
{ _id: 'maths', score: 87 }
> db.exercise.aggregate([{$group: {_id:"$name", "score":{$min:"$marks"}}}])
< { _id: 'Ramesh', score: 59 }
{ _id: 'Steve', score: 77 }
{ _id: 'Jan', score: 0 }
{ _id: 'Alison', score: 82 }
{ _id: 'Rav', score: 62 }
> db.exercise.aggregate([{$group: {_id:"$subject", "avg":{$avg:"$marks"}}}, {$sort:{"$marks": -1}}, {$limit: 2}])
MongoServerError: FieldPath field names may not start with '$'. Consider using $getField or $setField.
> db.exercise.aggregate([{$group: {_id:"$subject", "avg":{$avg:"$marks"}}}, {$sort:{"marks": -1}}, {$limit: 2}])
< { _id: 'english', avg: 62.6 }
{ _id: 'science', avg: 77.75 }
Atlas atlas-b1tccd-shard-0 [primary] exercise>