

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.

Relational model because the information seems like it will not change overtime so database should have strict and solid schema.

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.

MongoDB because the information is very dynamic and it has a lot of fields that seem like can be changed overtime and there is no confirmation that those field will be in every class projects.

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.

MongoDB because the information from two-dimension recorded in a time-sequence is a kind of data that should be contained in three-dimension matrix and MongoDB offers very flexible data container like objects in JSON.

4. Propose an appropriate Relational Model or MongoDB database schema

Here is the MongoDB schema inspired from "Battle Cats" game:

```
CAT
{
  "_id": ObjectId,
  "name": String,
  "form": String,
  "cost": Int,
  "stats": {
    "level": Int,
    "health": Int,
    "damage": Int,
    "range": Int,
    "speed": Int,
    "attack_speed": Int,
    "target": String,
    ...
  },
  "abilities": [
    {
      "name": String,
      "description": String,
      ...
    },
    ...
  ]
}
```

5.

Find the total marks for each student across all subjects.

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

Find the maximum marks scored in each subject.

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

Find the minimum marks scored by each student.

```
> db.collection.aggregate([{$group: {_id: "$subject", "min": {$min: "$marks"}}}])
< { _id: 'maths', min: 62 }
  { _id: 'english', min: 0 }
  { _id: 'science', min: 71 }
```

Find the top two subjects based on average marks.

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