1.**Relational Model** is more suitable because it suits more on a predefined relational data that are able to be updated overtime.

2.**MongoDB** is more suitable because it suits more when you have a lot of data or complexes data NoSQL makes it easy to store all different types of data together and without having to invest time into defining what type of data you're storing in advance.

3.**MongoDB** is more suitable because NoSQL databases are built to store data without a predefined schema. This makes it easy to make significant application changes in real-time.

4.**MongoDB for IOT**

Devices(dName,dID,Enabled)

DevicesInfo(dID,Temperature,Humidity,LightIntensity)

Users(uName,uID,dID)

I choose MongoDB because the data are dynamic.

Devices Info received data from hardware dynamically and will send data to the user

One user can have many device

Devices can be turn on and off

5.

Use Q5

```
> use Q5
< 'switched to db Q5'
```

```
> db.createCollection("Students")
< { ok: 1 }
```

Insert Data

```
db.Students.insertMany([
        {"name":"Ramesh","subject":"maths","marks":87},
        {"name":"Ramesh","subject":"english","marks":59},
        {"name":"Ramesh","subject":"science","marks":77},
        {"name":"Rav","subject":"maths","marks":62},
        {"name":"Rav","subject":"english","marks":83},
        {"name":"Rav","subject":"science","marks":71},
        {"name":"Alison","subject":"maths","marks":84},
        {"name":"Alison","subject":"english","marks":82},
        {"name":"Alison","subject":"science","marks":86},
        {"name":"Steve","subject":"maths","marks":81},
        {"name":"Steve","subject":"english","marks":89},
        {"name":"Steve","subject":"science","marks":77},
        {"name":"Jan","subject":"english","marks":0,"reason":"absent"}
    ]
)
```

```
{ acknowledged: true,
  insertedIds:
   { '0': ObjectId("62382bff6909768ae0ce7487"),
     '1': ObjectId("62382bff6909768ae0ce7488"),
     '2': ObjectId("62382bff6909768ae0ce7489"),
     '3': ObjectId("62382bff6909768ae0ce748a"),
     '4': ObjectId("62382bff6909768ae0ce748b"),
     '5': ObjectId("62382bff6909768ae0ce748c"),
     '6': ObjectId("62382bff6909768ae0ce748d"),
     '7': ObjectId("62382bff6909768ae0ce748e"),
     '8': ObjectId("62382bff6909768ae0ce748f"),
     '9': ObjectId("62382bff6909768ae0ce7490"),
     '10': ObjectId("62382bff6909768ae0ce7491"),
     '11': ObjectId("62382bff6909768ae0ce7492"),
     '12': ObjectId("62382bff6909768ae0ce7493") } }
```

Find the total marks for each student across all subjects.

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

Find the maximum marks scored in each subject.

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

Find the minimum marks scored by each student.

```
db.Students.aggregate([{$group:{_id:"$name",minMark:{$max:"$marks"}}}])
{ _id: 'Jan', minMark: 0 }
{ _id: 'Ramesh', minMark: 87 }
{ _id: 'Alison', minMark: 86 }
{ _id: 'Steve', minMark: 89 }
{ _id: 'Rav', minMark: 83 }
```

Find the top two subjects based on average marks.

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