

HW: NoSQL & MongoDB

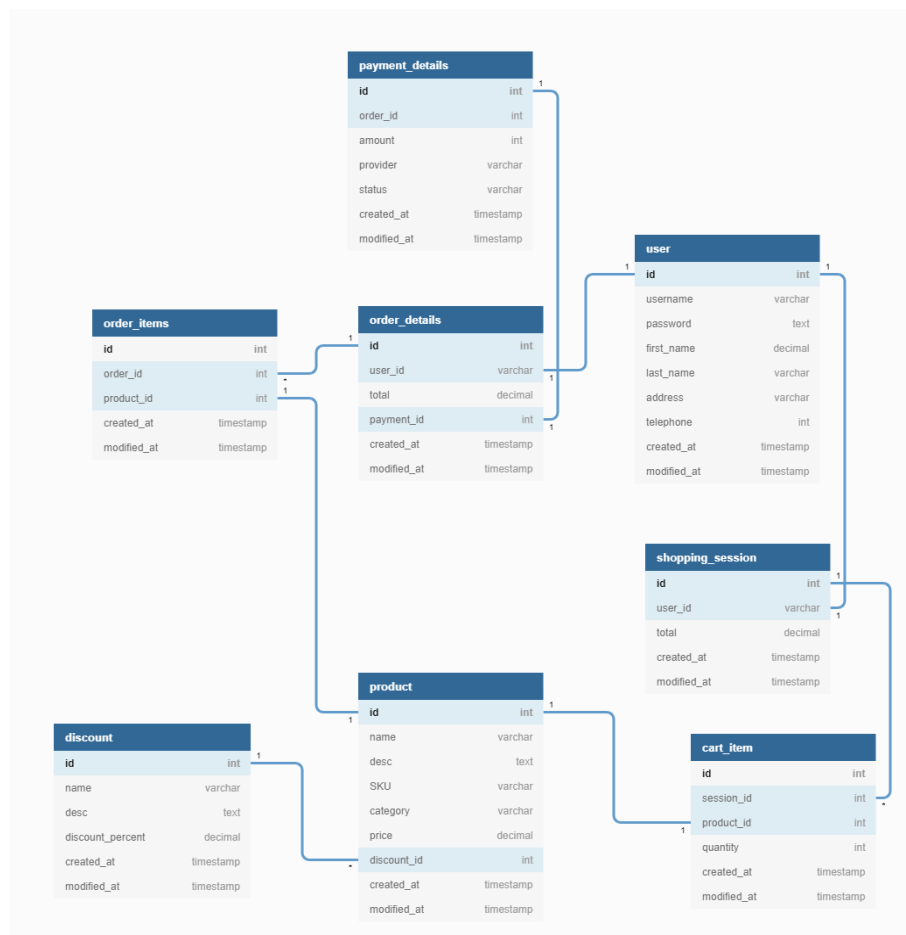
1. I choose the relational model, because this database has the predefined structure of data. And it has an uncomplicated database and not too big data.

2. I choose MongoDB, because the project may include the combination of students so the structure of data can be changed. And it has complicated and big Data so MongoDB is more efficient with this.

3. I choose MongoDB, because it is more scalable (horizontal-scaling). Which is suitable for the time-sequence readings.

4. I choose E-commerce

Relational model:



5.

Create database and collection

```
> use SubjectDatabase
< 'switched to db SubjectDatabase'
> db.createCollection("marks")
< { ok: 1 }
```

Insert data into collection

```
> db.marks.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}],)
< { acknowledged: true,
  insertedIds:
    { '0': ObjectId("623882a4ef271816a5c29b01"),
      '1': ObjectId("623882a4ef271816a5c29b02"),
      '2': ObjectId("623882a4ef271816a5c29b03"),
      '3': ObjectId("623882a4ef271816a5c29b04"),
      '4': ObjectId("623882a4ef271816a5c29b05"),
      '5': ObjectId("623882a4ef271816a5c29b06"),
      '6': ObjectId("623882a4ef271816a5c29b07"),
      '7': ObjectId("623882a4ef271816a5c29b08") } }
```

```
> db.marks.insertMany([{"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("62388335ef271816a5c29b09"),
      '1': ObjectId("62388335ef271816a5c29b0a"),
      '2': ObjectId("62388335ef271816a5c29b0b"),
      '3': ObjectId("62388335ef271816a5c29b0c"),
      '4': ObjectId("62388335ef271816a5c29b0d") } }
```

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

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

2. Find the maximum marks scored in each subject.

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

3. Find the minimum marks scored by each student.

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

4. Find the top two subjects based on average marks.

```
> db.marks.aggregate([{$group:{_id:"$subject","average marks":{$avg:"$marks"}}},{ $sort:{"average marks": -1}},{$limit:2}])  
< { _id: 'english', 'average marks': 62.6 }  
  { _id: 'science', 'average marks': 77.75 }
```