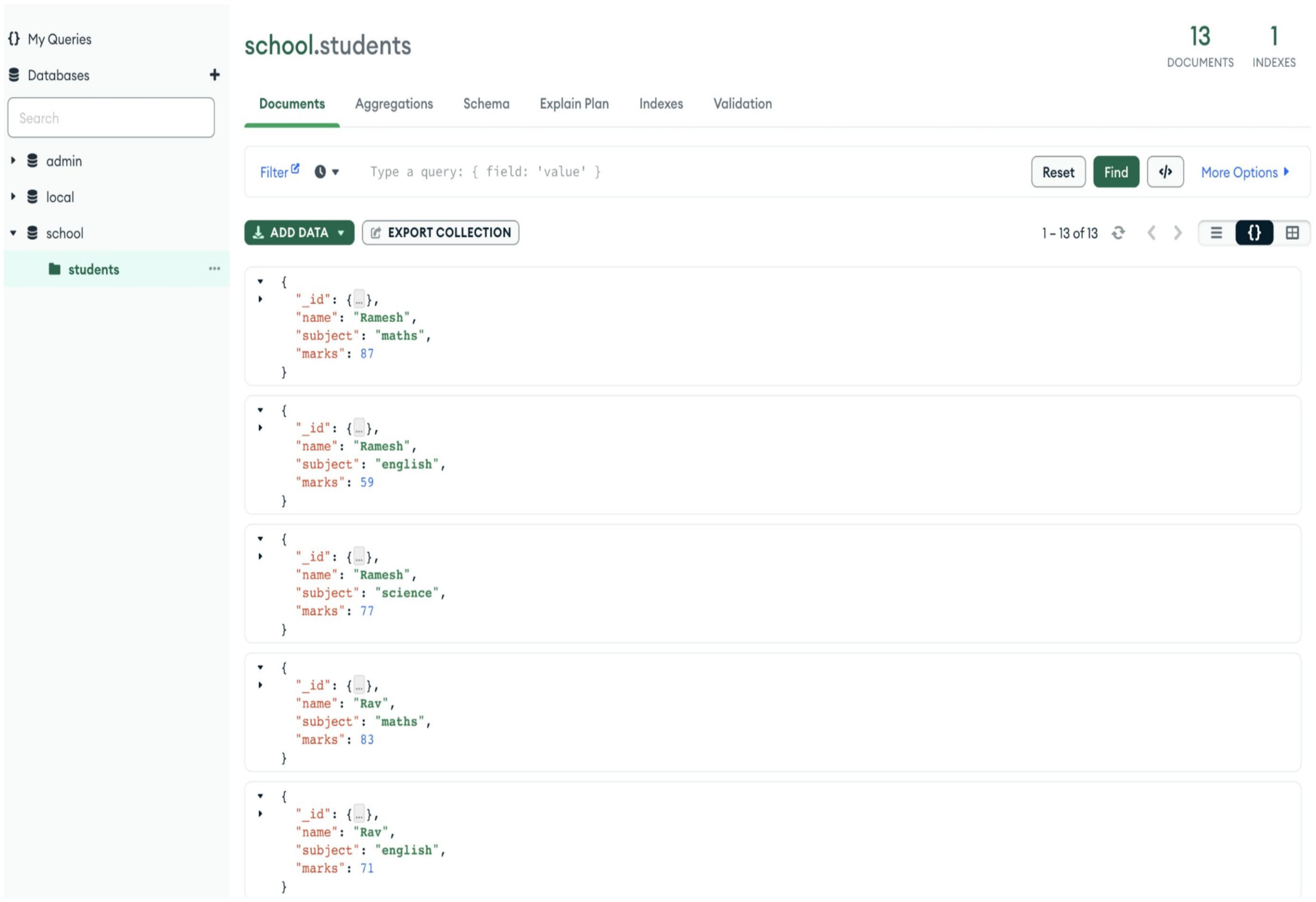
Warat Poovorakit
6410545771

HW# NoSQL & MongoDB

1. If the sensor measurements have a well-defined structure, it's better to use a relational database. If the data has varying fields or is unstructured, MongoDB is a good choice for unstructured or semi-structured data where flexibility and scalability are essential. MongoDB stores data as JSON-like documents, which can be nested and have varying fields, making it easier to store and query complex data structures.

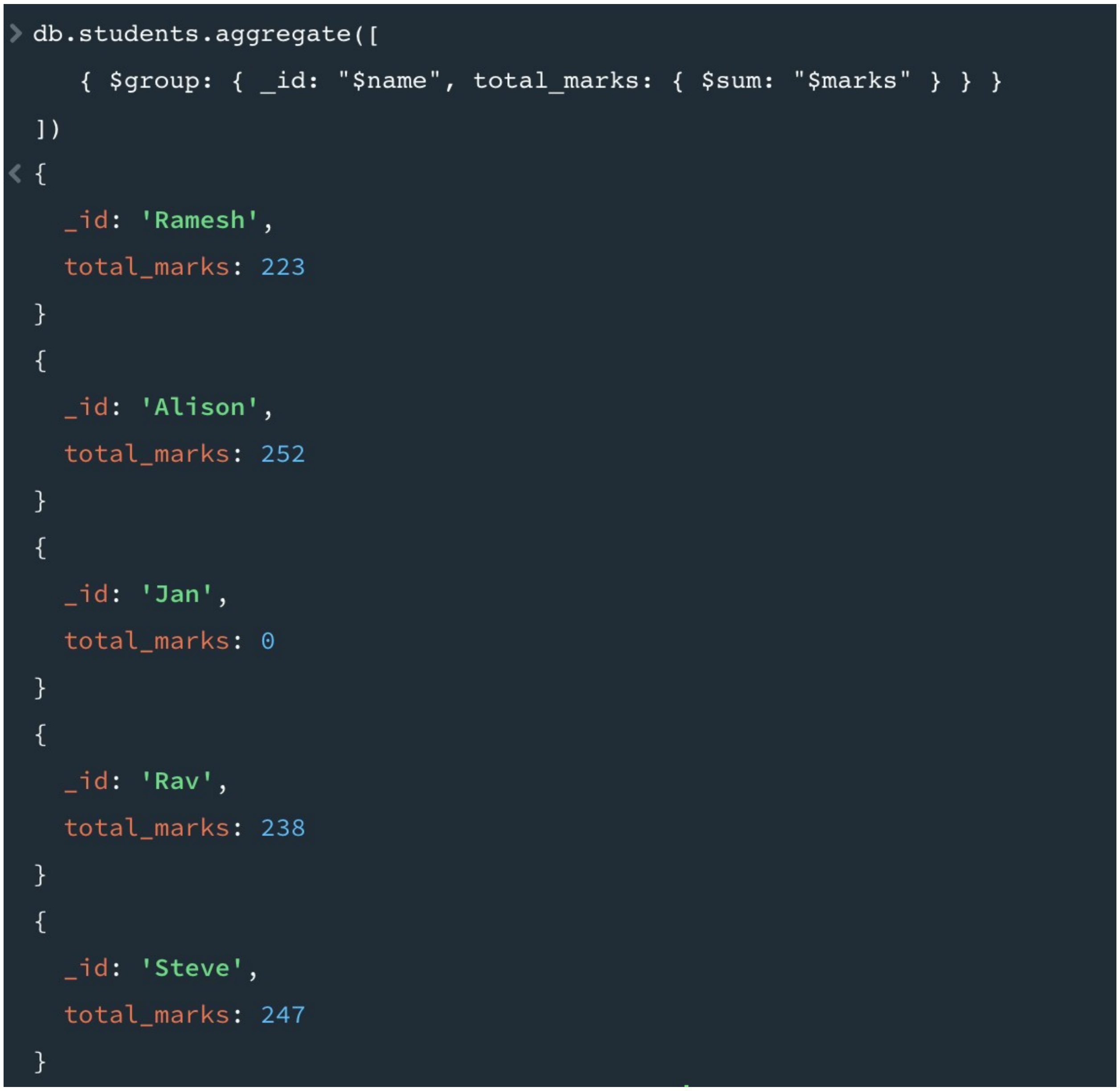
2. For IoT and Gaming applications, MongoDB's flexible document model is well-suited for storing unstructured and complex data. On the other hand, for E-commerce and Financial applications, the relational model is a better choice for storing structured data that requires robust transaction management, strict data integrity, and compliance with regulatory requirements. However, the actual choice of database may vary depending on the specific requirements of the application.

3. Insert 13 students data



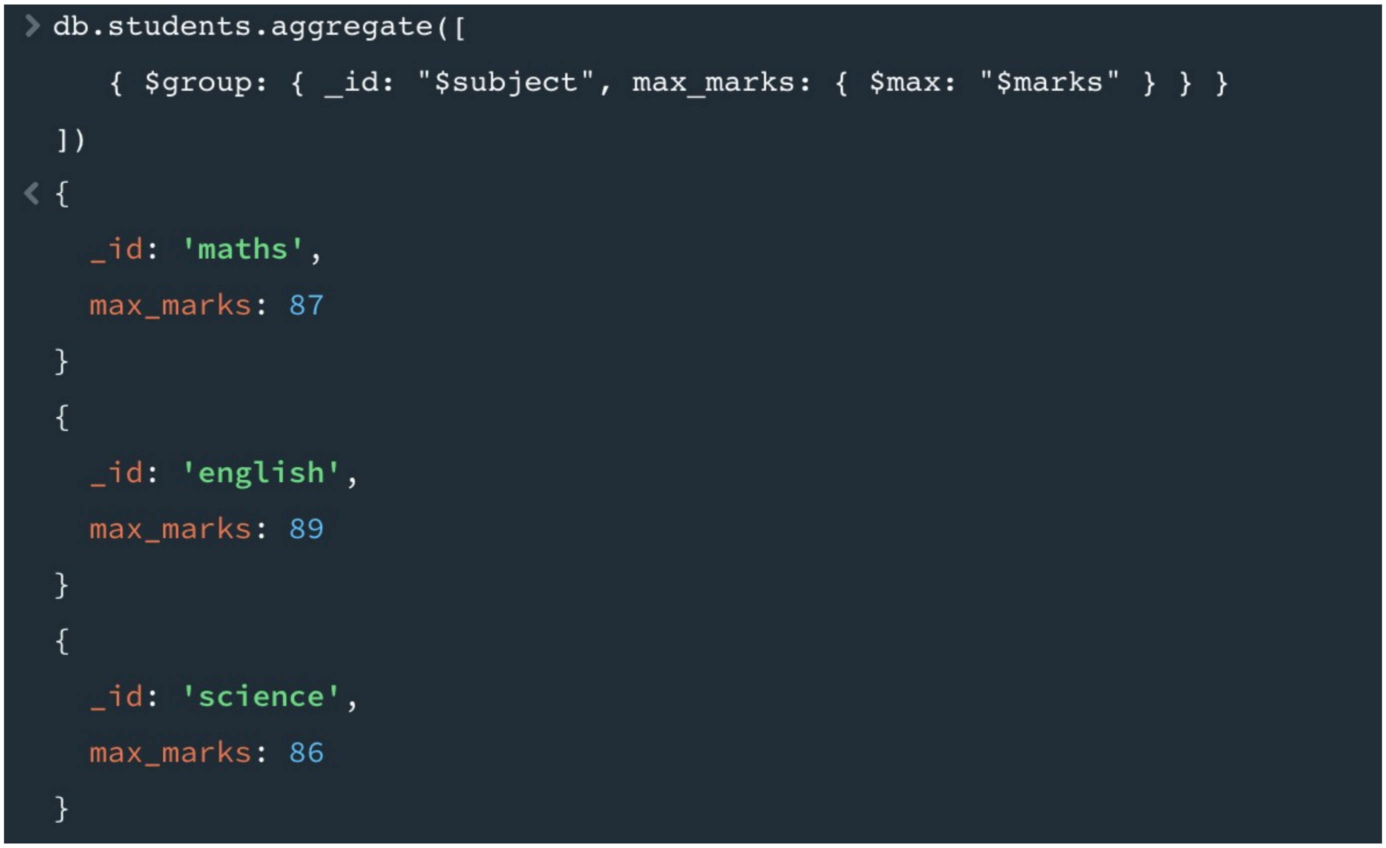
```
{
  "_id": {…},
  "name": "Ramesh",
  "subject": "maths",
  "marks": 87
}
{
  "_id": {…},
  "name": "Ramesh",
  "subject": "english",
  "marks": 59
}
{
  "_id": {…},
  "name": "Ramesh",
  "subject": "science",
  "marks": 77
}
{
  "_id": {…},
  "name": "Rav",
  "subject": "maths",
  "marks": 83
}
{
  "_id": {…},
  "name": "Rav",
  "subject": "english",
  "marks": 71
}
```

• **Find the total marks for each student across all subjects.**



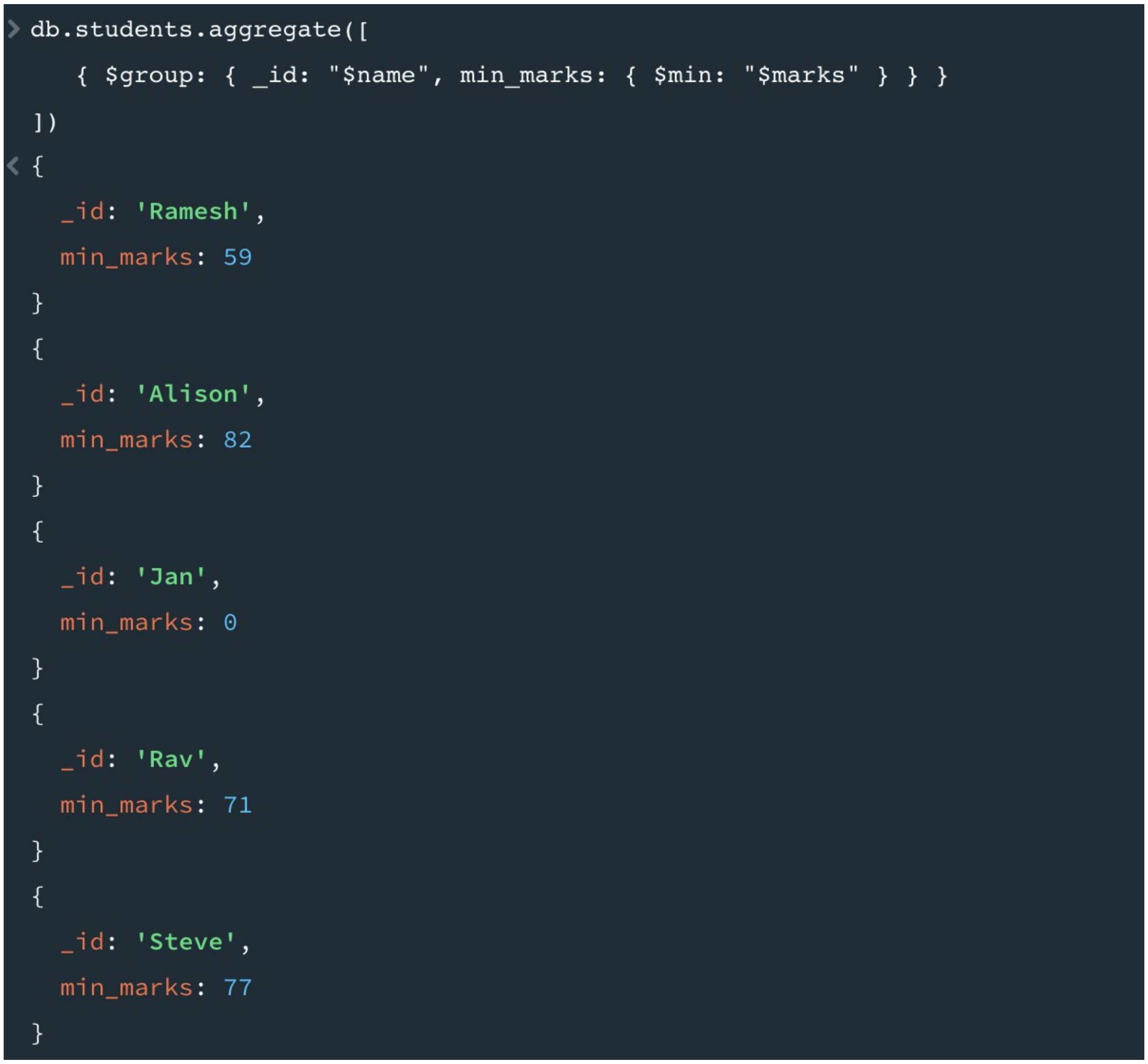
```
> db.students.aggregate([
    { $group: { _id: "$name", total_marks: { $sum: "$marks" } } }
  ])
< {
    _id: 'Ramesh',
    total_marks: 223
  }
  {
    _id: 'Alison',
    total_marks: 252
  }
  {
    _id: 'Jan',
    total_marks: 0
  }
  {
    _id: 'Rav',
    total_marks: 238
  }
  {
    _id: 'Steve',
    total_marks: 247
  }
```

• **Find the maximum marks scored in each subject.**



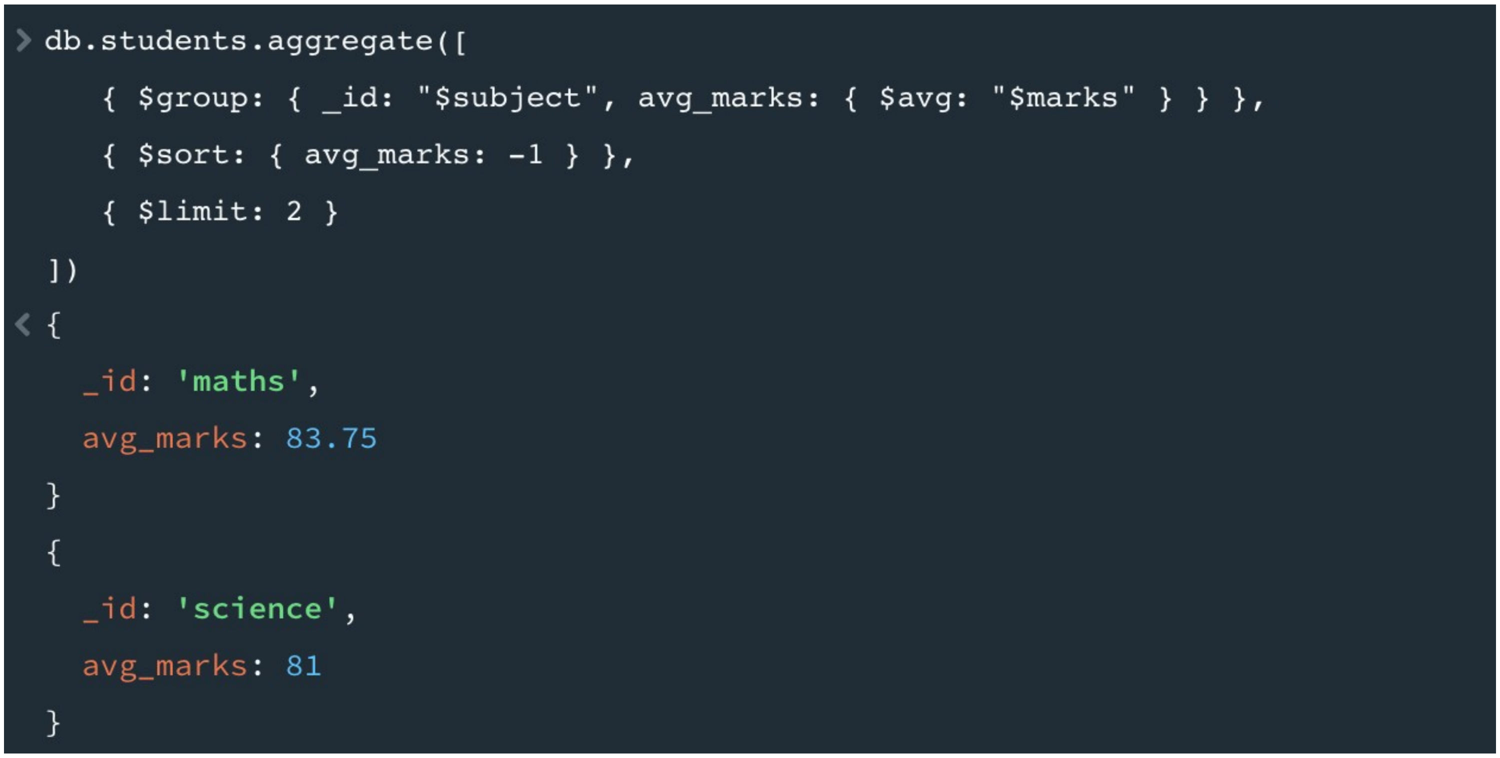
```
> db.students.aggregate([
     { $group: { _id: "$subject", max_marks: { $max: "$marks" } } }
  ])
< {
    _id: 'maths',
    max_marks: 87
  }
  {
    _id: 'english',
    max_marks: 89
  }
  {
    _id: 'science',
    max_marks: 86
  }
```

• **Find the minimum marks scored by each student.**



```
> db.students.aggregate([
    { $group: { _id: "$name", min_marks: { $min: "$marks" } } }
  ])
< {
    _id: 'Ramesh',
    min_marks: 59
  }
  {
    _id: 'Alison',
    min_marks: 82
  }
  {
    _id: 'Jan',
    min_marks: 0
  }
  {
    _id: 'Rav',
    min_marks: 71
  }
  {
    _id: 'Steve',
    min_marks: 77
  }
```

- **Find the top two subjects based on average marks.**



```
> db.students.aggregate([
    { $group: { _id: "$subject", avg_marks: { $avg: "$marks" } } },
    { $sort: { avg_marks: -1 } },
    { $limit: 2 }
  ])
< {
    _id: 'maths',
    avg_marks: 83.75
  }
  {
    _id: 'science',
    avg_marks: 81
  }
```