1. 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.

   Ans In this case the relational model would be a better fit for this database due to its ability to handle structured data with well-defined relationships and its powerful querying capabilities.

2. For each of the following applications
   a. IoT
   b. E-commerce
   c. Gaming
   d. Finance
   Propose an appropriate Relational Model or MongoDB database schema. For each application,
   clearly justify your choice of database.

Ans a. IoT:

For an IoT application, the most effective option for managing large amounts of data produced by sensors and devices is frequently a time-series database. A relational model database structure in this situation might include several tables, such as a table for devices, a table for sensor measurements, and a table for device events.

b. E-commerce:

For an e-commerce application, a Relational Model database is typically the best choice due to the structured nature of the data. The database schema can consist of tables for customers, orders, products, and payments, with relationships established between them using foreign keys.

c. Gaming:

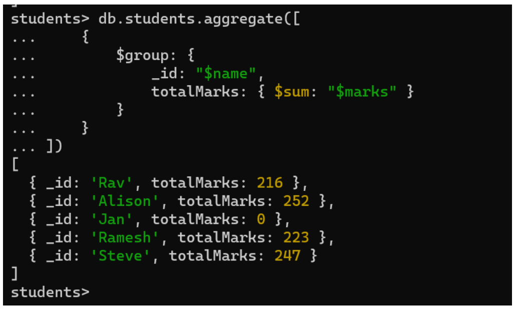
For a gaming application, a NoSQL database such as MongoDB may be the best choice due to the unstructured nature of the data. A MongoDB database schema may consist of collections for players, games, and achievements, with documents containing nested data structures.

d. Finance:

For a finance application, a Relational Model database is typically the best choice due to the importance of data consistency and the need for transactions. The database schema can consist of tables for accounts, transactions, and customers, with relationships established between them using foreign keys.

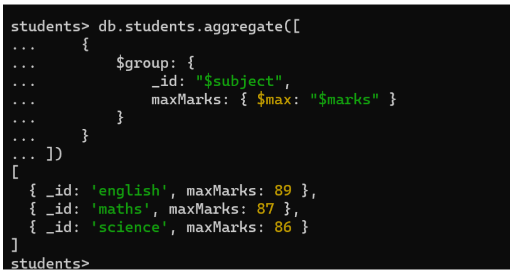
3. Create MongoDB database with following information.

3.1) Find the total marks for each student across all subjects.



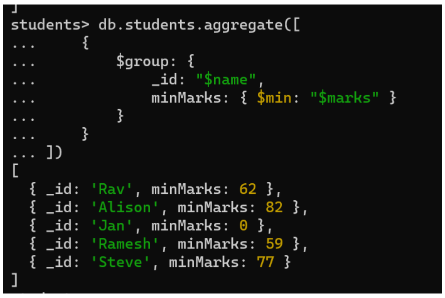
```
students> db.students.aggregate([
...     {
...         $group: {
...             _id: "$name",
...             totalMarks: { $sum: "$marks" }
...         }
...     }
... ])
[
  { _id: 'Rav', totalMarks: 216 },
  { _id: 'Alison', totalMarks: 252 },
  { _id: 'Jan', totalMarks: 0 },
  { _id: 'Ramesh',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

3.2) Find the maximum marks scored in each subject.



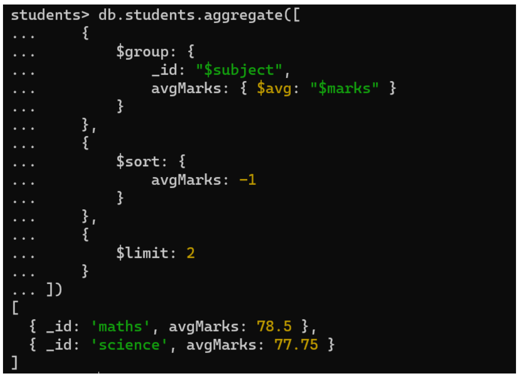
```
students> db.students.aggregate([
...     {
...         $group: {
...             _id: "$subject",
...             maxMarks: { $max: "$marks" }
...         }
...     }
... ])
[
  { _id: 'english', maxMarks: 89 },
  { _id: 'maths', maxMarks: 87 },
  { _id: 'science', maxMarks: 86 }
]
students>
```

3.3) Find the minimum marks scored by each student.



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

3.4 Find the top two subjects based on average marks.



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