BD05

Student:Peng simin 2023020207

Chen yinuo2023020101

Xu hongye2023020204

1. embedded documents, references, and mixed strategies
2. because MONGODB is a Nosql database, so it hasn't the fixed format to create table. So it is "schema-less".
3. In given example, we can see different rows have different data(type), and fields. That mean in this database the format is not fixed and we can add various data or field in this table. So it is "dynamic

4.
```
db.employees.insertMany([
  {empno: 'E55', lastname: 'Clark', firstname:'STEFF',hours: 15, gender:'FRAU'},
  {empno: 'E44', lastname: 'SUSAN', firstname:'DAVENC',hours: 22, gender:'FRAU'},
  {empno: 'E66', lastname: 'HUNK', firstname:'XU',hours: 16, gender:'MANNER'},
  {empno: 'E77', lastname: 'JOY', firstname:'WU',hours: 40, gender:'FRAU'}
]);
db.employees.find({empno: 'E66'});
```

OUTPUT:

```
{
      "acknowledged" : true,
      "insertedIds" : [
            ObjectId("67068aaee23c0750e8698f2e"),
            ObjectId("67068aaee23c0750e8698f2f"),
            ObjectId("67068aaee23c0750e8698f30"),
            ObjectId("67068aaee23c0750e8698f31")
      ]
}
{ "_id" : ObjectId("67068aaee23c0750e8698f30"), "empno" : "E66", "lastname" : "HUNK", "firstname" : "XU", "hours" : 16, "gender" : "MANNER" }
```

5.

(a)value,volume,variety,velocity,veracity

(b) Big Data encompasses not only structured data (such as that found in relational databases) but also semi-structured data (like XML and JSON formats) and unstructured data (such as text, images, audio, and video).

(c) MongoDB's flexible schema design, document-oriented storage, indexing capabilities, aggregation pipelines, and sharding technology make it easy to deal with variety data.

6.(b)

i) Find the books which were published between 2019 and 2024.

```
 db.books.find({
      year: {
            $gte: 2019,
            $lte: 2024
      }
})
```

ii) Find the books whose book_id is not equal to 552020.

```
db.books.find({
      book_id: { $ne: "552020" }
})
```

iii) Find the books by D. Sullivan or whose isbn is equal to 9780134023212.

```
 db.books.find({
      $or: [
            { author: "D. Sullivan" },
            { ISBN: "9780134023212" }
      ]
})
```

iv) Find the books by whose isbn is equal to 9876543210 or to 0123456789.

```
 Db.books.find({
      $or:[ {isbn:"9876543210"},
      {isbn:" 0123456789"}]
      })
```

v) Find the books whose title contains the string “SQL”.

```
 db.books.find({
      title: /SQL/
})
```

vi) Find the number of books published by Addison-Wesley:

```
 db.books.countDocuments({
      publisher: "Addison-Wesley"
})
```

vii) Find the books published in 2019 and whose title contains the string “Mortals”, sorted by title in alphabetical order.

```
 db.books.find({
      year: 2019,
      title: /Mortals/
}).sort({
      title: 1
})
```

viii) Add another field named “subject” which is set to “computing” to the books published in 2019.

```
db.books.updateMany(
      { year: 2019 },
```

```
        { $set: { subject: "computing" } }
)
```