ITIS 5166
Network-Based Application Development

Data Modeling

Data Modeling

- In MongoDB
  - an entity is modeled as a document
  - an attribute of an entity is modeled as a field of the document
- A relationship may exist between entities
  - one-to-one relationship
  - one-to-many relationship
  - many-to-many relationship

Model Relationships

- MongoDB captures relationships between data with
- Embedded model: store related data in a single document
- Normalized model: store related data in separate collections, and reference the _id field of one document in the other
- Embedded model has the best performance
- Normalized model provides more flexibility for query

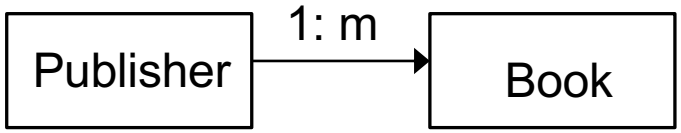
One-to-One Relationship

- Using embedded model for one-to-one relationship unless the document exceeds the maximum document size of 16 MB
- Example: Each theater has an address. The theater document embeds its address as a subdocument.

```
{
    _id: "59a47286cfa9a3a73e51e72d",
    capacity: 1000,
    address: {
              street1: "45235 Worth Ave.",
              city: "California",
              state: "MD",
              zipcode: 20619
            }
}
```

One-to-Many Relationship

- Example: each book has a publisher, and a publisher may publish multiple books



- Embed “one” inside “many” as a subdocument
- Embed “many” as an array of subdocuments in “one”
- Store _id of “many” as an array inside “one”
- Store _id of “one” inside “many”

Embed One Inside Many

- Embedding the publisher as a subdocument in each book document
- There are duplicated data
- Any change to a publisher requires updating all of its books

```
{
    title: "MongoDB: The Definitive Guide",
    author: [ "Kristina Chodorow", "Mike Dirolf" ],
    published_date: ISODate("2010-09-24"),
    pages: 216,
    language: "English",
    publisher: {
                name: "O'Reilly Media",
                founded: 1980,
                location: "CA"
              }
}
```

```
{
    title: "50 Tips and Tricks for MongoDB Developer",
    author: "Kristina Chodorow",
    published_date: ISODate("2011-05-06"),
    pages: 68,
    language: "English",
    publisher: {
                name: "O'Reilly Media",
                founded: 1980,
                location: "CA"
              }
}
```

Embed Many Inside One

- Embedding the books as an array of subdocuments in the publisher document
- Easy to find all books published by a publisher
- Not easy to find all books written by a particular author
- The document may exceed the maximum document size of 16 MB

Reference Many in One

- Store books and publishers in separate collections
- Store the _id of books as an array in the publisher document
- if the number of books per publisher is unbounded, this data model would lead to mutable, growing arrays

```
{
    name: "O'Reilly Media",
    founded: 1980,
    location: "CA",
    books: [123456789, 234567890, ...]
}
```

```
{
    _id: 234567890,
    title: "50 Tips and Tricks for MongoDB Developer",
    author: "Kristina Chodorow",
    published_date: ISODate("2011-05-06"),
    pages: 68,
    language: "English"
}
```

```
{
    _id: 123456789,
    title: "MongoDB: The Definitive Guide",
    author: [ "Kristina Chodorow", "Mike Dirolf" ],
    published_date: ISODate("2010-09-24"),
    pages: 216,
    language: "English"
}
```

Reference One in Many

- Store books and publishers in separate collections
- Store the _id of publisher in the book document

```
{
    _id: "oreilly",
    name: "O'Reilly Media",
    founded: 1980,
    location: "CA"
}
```

```
{
    _id: 123456789,
    title: "MongoDB: The Definitive Guide",
    author: [ "Kristina Chodorow", "Mike Dirolf" ],
    published_date: ISODate("2010-09-24"),
    pages: 216,
    language: "English",
    publisher_id: "oreilly"
}
```

Summary

- Use embedded model for
  - one-to-one relationship
  - one-to-many relationship when
    - the “many” documents always appear with or are viewed in the context of “one” document
    - and embedding “many” as an array of subdocuments in “one” does not exceed the maximum document size
- Use reference model otherwise
  - flexible query is needed
  - the embedded documents are unbounded in a one-to-many relationship
  - many-to-many relationships