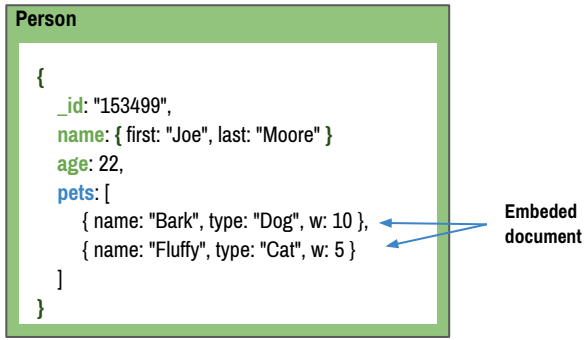
Modelado de Datos

Bases de Datos 2024

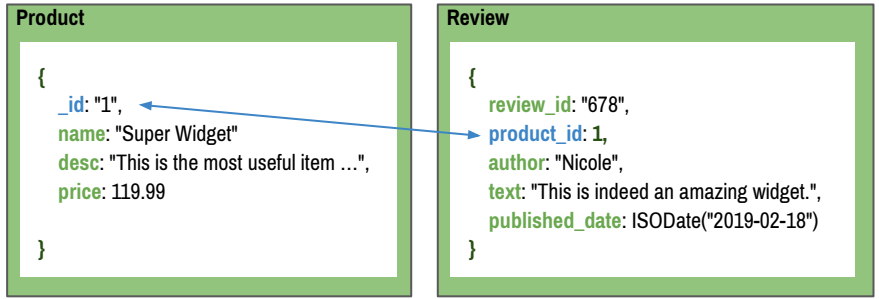
Estrategias de modelado de datos

- Decisión clave para el modelado de relaciones
  - Anidar (embed) datos VS. usar Referencias (references)

Modelo de datos Anidado



Modelo de datos usando Referencias



Modelado de Relaciones

- Relaciones One-to-One
  - Modelado con Documentos Anidados
  - Modelado con Referencias de Documentos
- Relaciones One-to-Many
  - Modelado con Documentos Anidados
  - Modelado con Referencias de Documentos

- Principio General in MongoDB
  - Datos que se accede juntos deben ser almacenado juntos

Modelado de Relaciones - One-To-One

Modelado con Referencias de Documentos

```
db.student.insertOne(
 {
      "_id": "jmoore",
      "name": "James Moore"
 }
)

db.address.insertOne(
 {
      _id": "a1"
      "student_id": "jmoore"
      "street": "123 Sesame St",
      "city": "Anytown",
      "zip": "12345"
 }
)
```

Modelado con Documentos Anidados

```
db.student.insertOne(
 {
      "_id": "jmoore",
      "name": "James Moore",
      "address" : {
           "street": "123 Sesame St",
           "city": "Anytown",
           "zip": "12345"
      }
 }
)
```

Modelado de Relaciones - One-To-Many

- Modelado con Referencias de Documentos

```
db.student.insertOne( {
        "_id": "jmoore",
        "name": "James Moore"
   } )

db.address.insertOne( {
        _id": "a1"
        "student_id": "jmoore"
        "street": "123 Sesame St",
        "city": "Anytown",
        "zip": "12345"
} )

db.address.insertOne( {
        _id": "a2"
        "student_id": "jmoore"
        "street": "321 Some Other Street ",
        "city": "Boston",
        "zip": "45678"
} )
```

- Modelado con Documentos Anidados

```
db.student.insertOne(
 {
      "_id": "jmoore",
      "name": "James Moore",
      "address" : [
         {
              "street": "123 Sesame St",
              "city": "Anytown",
              "zip": "12345"
         },
         {
              "street": "321 Some Other Street ",
              "city": "Boston",
              "zip": "45678"
         }
       ]
 }
)
```

Modelado de Relaciones - One-To-Many - Referencias

Modelado con arreglo de Referencias

```
db.products.insertOne( {
            "_id": "product1",
            name: "left-handed smoke shifter",
            manufacturer: "Acme Corp",
            catalog_number: 1234,
            parts: [ "part1", "partN"]
} )

db.parts.insertMany( [ {
            "_id": "part1",
            partno: "123-aff-456",
            name: "#4 grommet",
            qty: 94,
            price: 3.99
      }, {
            "_id": "partN",
            partno: "123-aff-678",
            name: "#5 grommet",
            qty: 94,
            price: 3.29
} ] )
```

Modelado con Referencias de Documentos

```
db.hosts.insertOne( {
            _id : "host1",
            name : "goofy.example.com",
            ipaddr : "127.66.66.66"
} )

db.logmsg.insertMany( [
   {
            _id: 1000001,
            time : ISODate("2014-03-28T09:42:41"),
            message : "cpu is on fire!",
            id_host: "host1"
   },
   {
            _id: 1000002,
            time : ISODate("2014-03-28T09:49:41"),
            message : "cpu is iddle!",
            id_host: "host1"
   }
] )
```

Embedding VS Referencing

	Embedding	Referencing
Ventajas	Una sola query para recuperar datos Una sola operación para update/delete data	Puede evitar duplicación de datos Documentos más pequeños
Desventajas	Duplicación de datos Documentos grandes	Se necesita join a nivel de aplicación

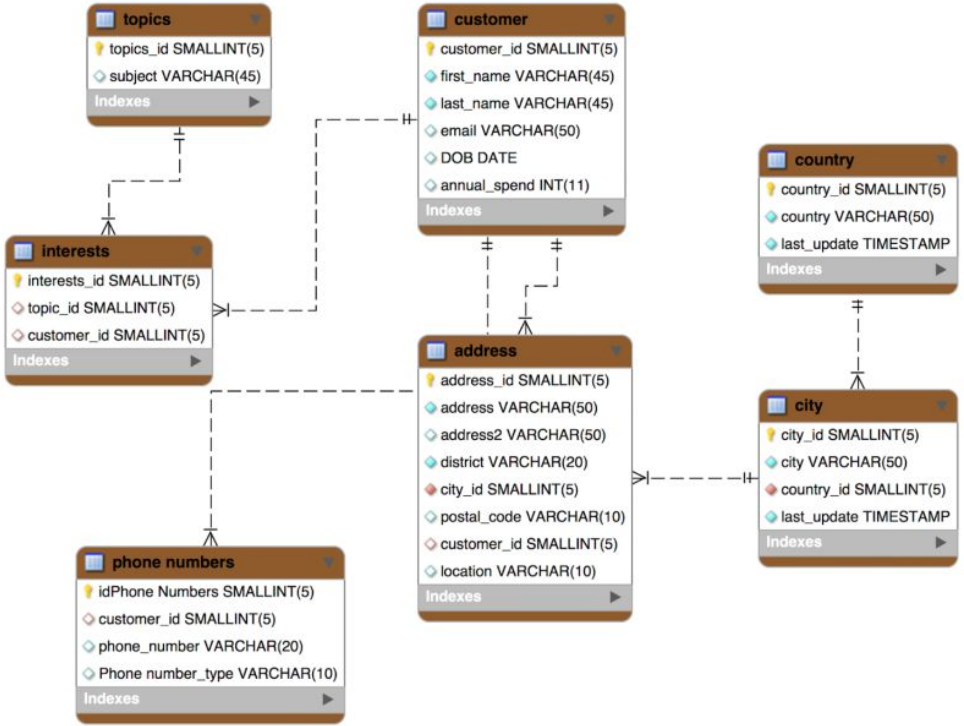
Modelado de Datos Dirigido por Queries

- Identificar entidades, atributos y relaciones
- Identificar las queries importantes
  - Y analizar la carga de trabajo de las queries (por ejemplo, frecuencia y latencia)
- Crear el modelo de datos aplicando las estrategias Anidar datos y Referencias
  - Hay que tener en mente las queries más importantes para crear el modelo de datos.
  - El modelo de datos debe permitir satisfacer las queries o patrones de acceso más importantes de manera eficiente
    - Para este curso, una “consulta es eficiente” si se puede responder en un sola query sin $lookup

IT’S DEMO TIME

Modelado de Datos Dirigido por Queries: Ejemplo

- Dado el diagrama junto con las queries más importantes crear un modelo datos en MongoDB



- Query 1:

  Listar el id, nombre, apellido y teléfonos (número y tipo) de los clientes

- Query 2:

  Listar los clientes (nombre, apellido y email) de una ciudad en particular

- Query 3:

  Listar los clientes (nombre, apellido y email) interesados en un tópico en particular

Modelado de Datos Dirigido por Queries: Ejemplo

- Query 1: Listar el id, nombre, apellido y teléfonos (número y tipo) de los clientes
  - Entidades: customer y phone_numbers
  - Relación: One-To-Many
  - Estrategia: Documentos Anidados
- Query 2: Listar los clientes (nombre, apellido y email) de una ciudad en particular
  - Entidades: customer, address, city, y country
  - Relación: One-To-Many (entre customer y address)
  - Estrategia: Documentos Anidados
- Query 3: Listar los clientes (nombre, apellido y email) interesados en un tópico en particular
  - Entidades: customer, interests, topics
  - Relación: Many-To-Many (entre customer y topics)
  - Estrategia: Documentos Anidados

Modelado de Datos Dirigido por Queries: Ejemplo

Modelo de datos en MongoDB

```
db.customer.insertOne( {
   customer_id: "1",
   name: { first: "John", last: "Moore" },
   email: "jmoore@example.com",
   annual_spend: 50000,
   phone_numbers: [
      {
         type: "Home",
         number: "238479823749"
      }
   ],
   addresses: [
      {
         address: "sample address",
         address2: "sample address2",
         district: "sample district",
         city: "sample city",
         country: "sample country",
         postal_code: "79878",
         location: "sample location"
      }
   ],
   topics: [ "topic 1", "topic 2" ]
} )
```

Query 1

```
db.customer.find( { } , { customer_id: 1, name: 1, phone_numbers: 1, _id:0 } )
```

Query 2

```
db.customer.find(
   { addresses: {
      $elemMatch: { city: "sample city", country: "sample country"}
   } },
   {name: 1, email: 1, _id:0}
)
```

Query 3

```
db.customer.find( { topics: { $in: [ "topic 1" ] } }, { name: 1, email: 1, _id:0 } )
```

Temas a estudiar

- Próxima clase
  - Indices
  - Introducción a bases de datos NoSQL, incluye teorema CAP
- Referencias
  - Modelado de datos