Bases de datos de documentos


```
{
  "data": [
    {
      "id": "X999_Y999",
      "from": {
        "name": "Tom Brady", "id": "X12"
      },
      "message": "Looking forward to 2010!",
      "actions": [
        {
          "name": "Comment",
          "link": "http://www.facebook.com/X999/posts/Y999"
        },
        {
          "name": "Like",
          "link": "http://www.facebook.com/X999/posts/Y999"
        }
      ],
      "type": "status",
      "created_time": "2010-08-02T21:27:44+0000",
      "updated_time": "2010-08-02T21:27:44+0000"
    }]
}
```


Bases de Datos No Relacionales

Instituto de Computación, FING, UdelaR – 2023

CC-BY Lorena Etcheverry lorenae@fing.edu.uy

Una base de datos de documentos
es una base **no relacional**
que almacena datos como
documentos estructurados,
típicamente
XML y **JSON**

Agenda

- Algo de historia
- JavaScript Object Notation (JSON)
- JSON Databases: MongoDB como ejemplo

Algo de historia

XML: la vedette de los 2000s

Un lenguaje de marcado para representar datos.

XML 1.0 W3C recommendation en Febrero 1998.

¿para qué se pensó y para que se usa/usó?

Separar datos de presentación en la web 2.0

Agregar metadatos y esquema (dar estructura).

Intercambio de datos entre aplicaciones.

Interoperabilidad!

Estándares asociados a XML

XPath

Una sintaxis para recuperar partes del documento: filtros, comodines.

XQuery

Un lenguaje de consulta sobre XML. El SQL de XML!

XML Schema

Un documento XML especial que describe la estructura de otro documento XML

XSLT (eXtensible Stylesheet Language Transformations)

Un lenguaje de transformaciones. Permite generar otros formatos (ej: HTML)

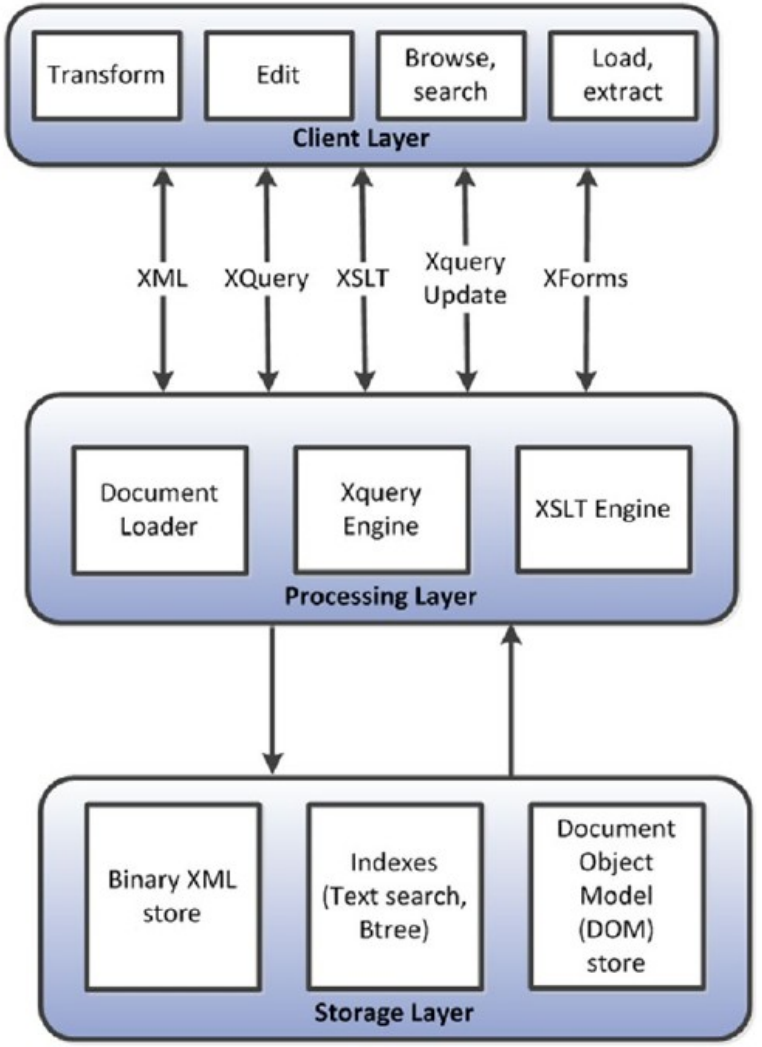
DOM (Document Object Model)

Una API orientada a objetos para interactuar con documentos XML.

Y como muchas aplicaciones generaban XML ...

Surgen las BDs de documentos para gestionar colecciones de documentos XML.

Muchas soluciones comerciales.



pero XML tiene algunos inconvenientes...

```
<!DOCTYPE glossary PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
 <glossary><title>example glossary</title>
  <GlossDiv><title>S</title>
   <GlossList>
    <GlossEntry ID="SGML" SortAs="SGML">
     <GlossTerm>Standard Generalized Markup
                Language</GlossTerm>
     <Acronym>SGML</Acronym>
     <Abbrev>ISO 8879:1986</Abbrev>
     <GlossDef>
      <para>A meta-markup language, used to create markup
            languages such as DocBook.</para>
      <GlossSeeAlso OtherTerm="GML">
      <GlossSeeAlso OtherTerm="XML">
     </GlossDef>
     <GlossSee OtherTerm="markup">
    </GlossEntry>
   </GlossList>
  </GlossDiv>
 </glossary>
```

"Some languages can be read by humans, but not by machines,

while others can be read by machines but not by humans.

XML solves this problem by being readable to neither."

Post by deleted, Reddit [1]

[1] https://www.reddit.com/r/ProgrammerHumor/comments/1mbgih/xml/

Mientras tanto, en el mundo de los desarrolladores de aplicaciones web ...

Javascript se impone como lenguaje de programación tanto del lado del cliente como del servidor (ej: Node.js).

AJAX: mensajería asíncrona entre cliente y servidor (originalmente pensado sobre XML).

Aparece **Javascript Object Notation (JSON)** como mecanismo para serializar objetos.

JSON: The Fat-Free alternative to XML [1]

[1]http://www.json.org/xml.html

Estructura de un doc JSON

objeto

clave

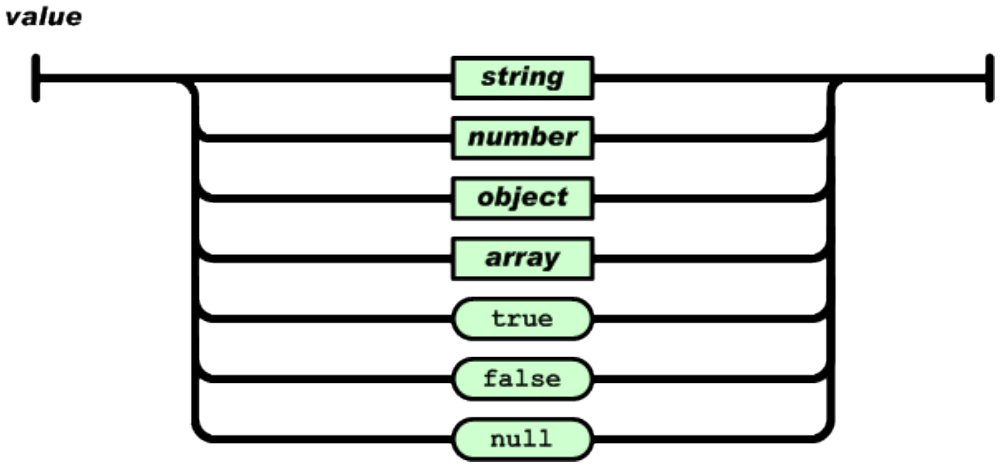
valor

arreglo (lista de valores)

```
{ "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "SortAs": "SGML",
          "GlossTerm": "Standard Generalized Markup
                       Language",
          "Acronym": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language, used to
                     create markup languages such as DocBook.",
            "GlossSeeAlso": ["GML", "XML"]
          },
          "GlossSee": "markup"
        }
      }
    }
  }
}
```

Las *claves* son *strings*.

Los *valores* pueden ser cualquier elemento de la notación.



XML vs JSON (1)


xml
Término de búsqueda
json
Término de búsqueda
Añadir comparación
Todo el mundo
2004 - hoy
Todas las categorías
Búsqueda web
Interés a lo largo del tiempo
100
75
50
25
Media
1 ene 2004
1 mar 2011
mar 2022
xml 10
json 18
Los valores de este punto tienen datos parciales (incompletos)

XML vs JSON (2)

```
<!DOCTYPE glossary PUBLIC "-//OASIS//DTD DocBook
V3.1//EN">
 <glossary><title>example glossary</title>
  <GlossDiv><title>S</title>
   <GlossList>
    <GlossEntry ID="SGML" SortAs="SGML">
     <GlossTerm>Standard Generalized Markup
                Language</GlossTerm>
     <Acronym>SGML</Acronym>
     <Abbrev>ISO 8879:1986</Abbrev>
     <GlossDef>
      <para>A meta-markup language, used to create
markup
            languages such as DocBook.</para>
      <GlossSeeAlso OtherTerm="GML">
      <GlossSeeAlso OtherTerm="XML">
     </GlossDef>
     <GlossSee OtherTerm="markup">
    </GlossEntry>
   </GlossList>
  </GlossDiv>
 </glossary>
```

```
{ "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "SortAs": "SGML",
          "GlossTerm": "Standard Generalized Markup
                                Language",
          "Acronym": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language, used to
                     create markup languages such as
                     DocBook.",
            "GlossSeeAlso": ["GML", "XML"]
           },
          "GlossSee": "markup"
        }
      }
    }
  }
}
```

Algunas ventajas de JSON

- JSON es **menos verboso**: más liviano para el intercambio de datos y más rápido su parseo.

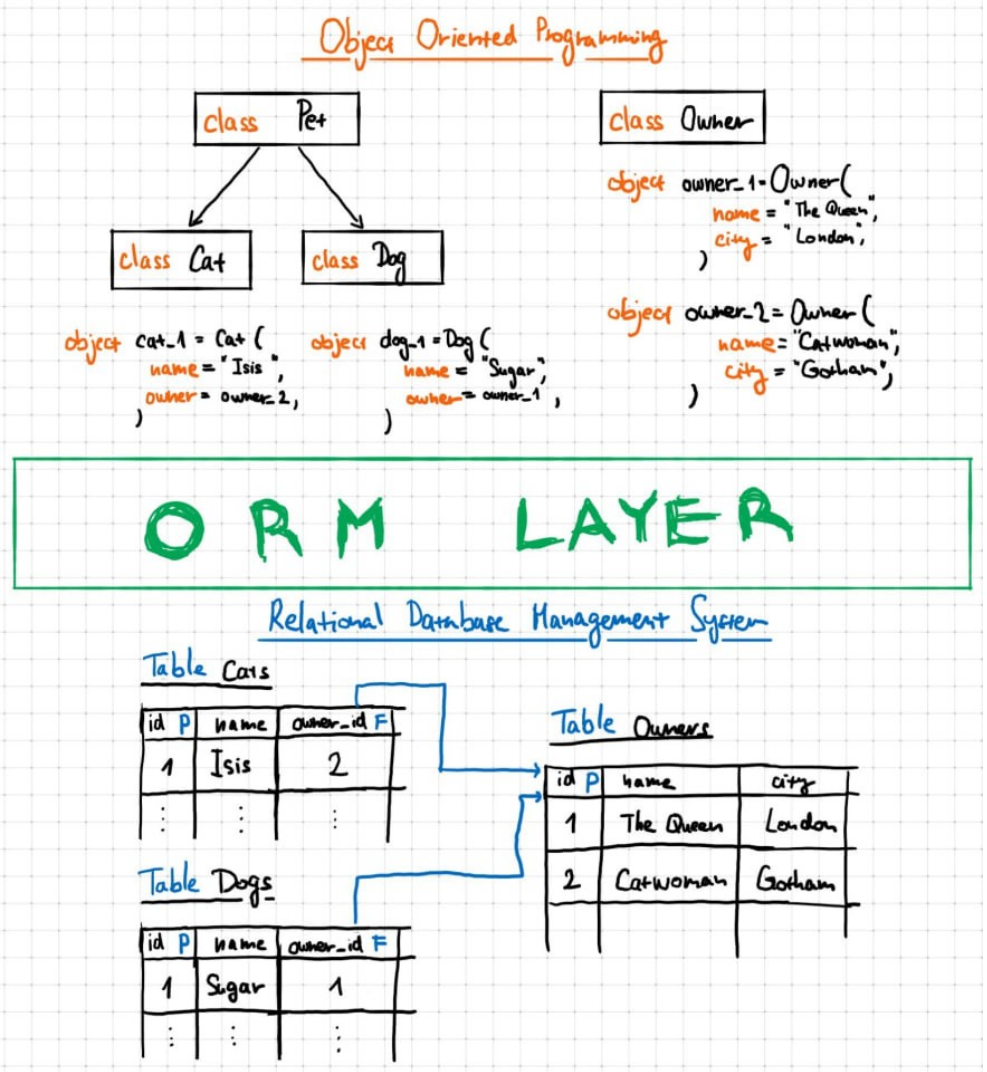
- JSON representa **directamente** el objeto: más sencillo para el desarrollador

- No tengo un esquema fijo (salvo q use JSON Schema), pero tengo **algo de estructura**.

- Si desarrollo en Javascript, intercambio JSON y almaceno JSON tengo un ***full stack* Javascript**.

- Si desarrollo OO y almaceno JSON **no tengo *impedance mismatch*** entre el modelo OO y el relacional.

Object- relational impedance mismatch



Describe la dificultad que surge al tratar de integrar sistemas de bases de datos relacionales con lenguajes de programación orientados a objetos.

Algunos “problemas”

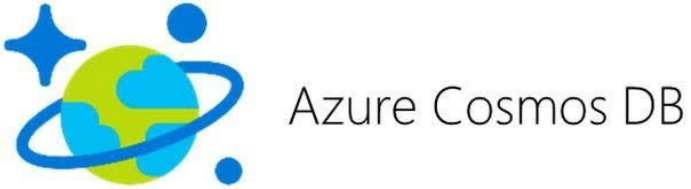
1. problemas de granularidad
2. problema de la herencia
3. problema de la identidad
4. problema de la asociación y navegación de datos

Ireland, C., & Bowers, D. (2015, May). Exposing the myth: object-relational impedance mismatch is a wicked problem. In DBKDA 2015, The Seventh International Conference on Advances in Databases, Knowledge, and Data Applications (pp. 21-26). IARIA XPS Press.

JSON Databases (a.k.a Document databases): MongoDB como ejemplo

JSON Databases

- A diferencia de otras BD No Relacionales no hay un *manifesto* o conjunto de características que deben satisfacer
- El único requisito: almacenar JSON!!



Terminología y conceptos

BD relacional

BD documental

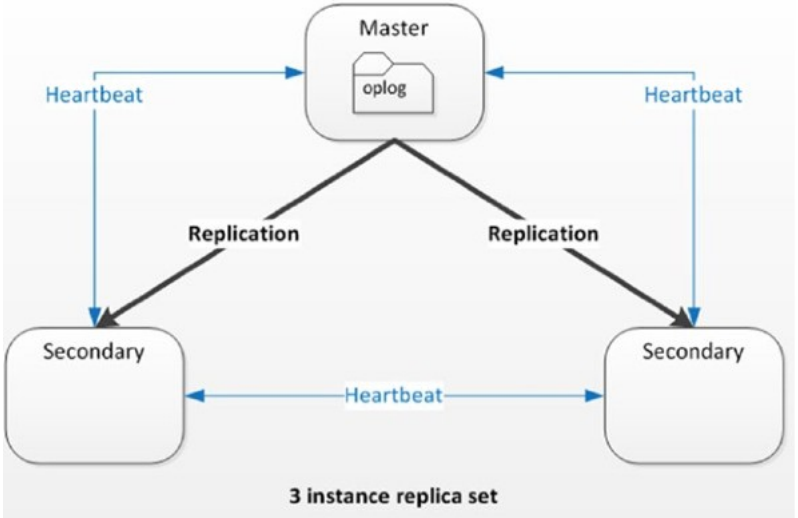
Tabla	Colección
Fila	Documento
Columna	Campo (*field*)
SQL Join	Documentos embebidos y referencias
SQL Group-by (agregación)	*Aggregation pipeline*

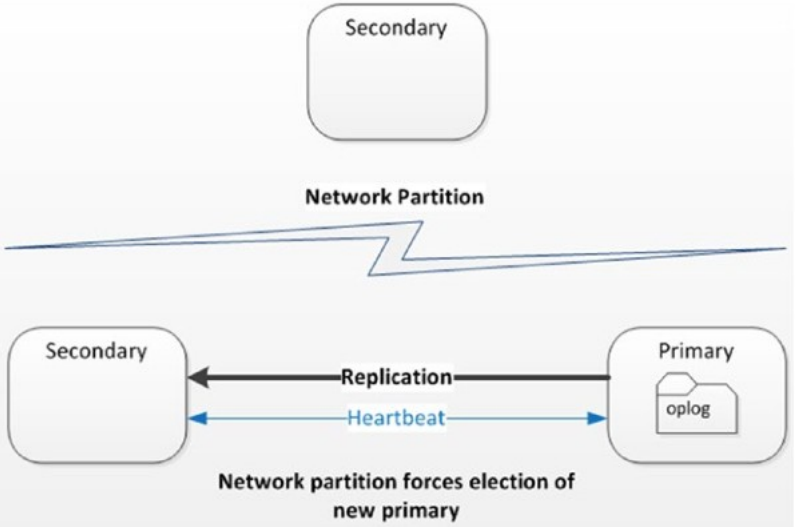
Algunas características técnicas

- Mecanismo de réplicas
- Particionamiento horizontal (sharding)
- Soporte a transacciones

Réplicas

- MongoDB propone un mecanismo de réplicas organizado en *replica sets*.
  - Recuperación automática ante particiones de red (se elige un nuevo nodo primario)

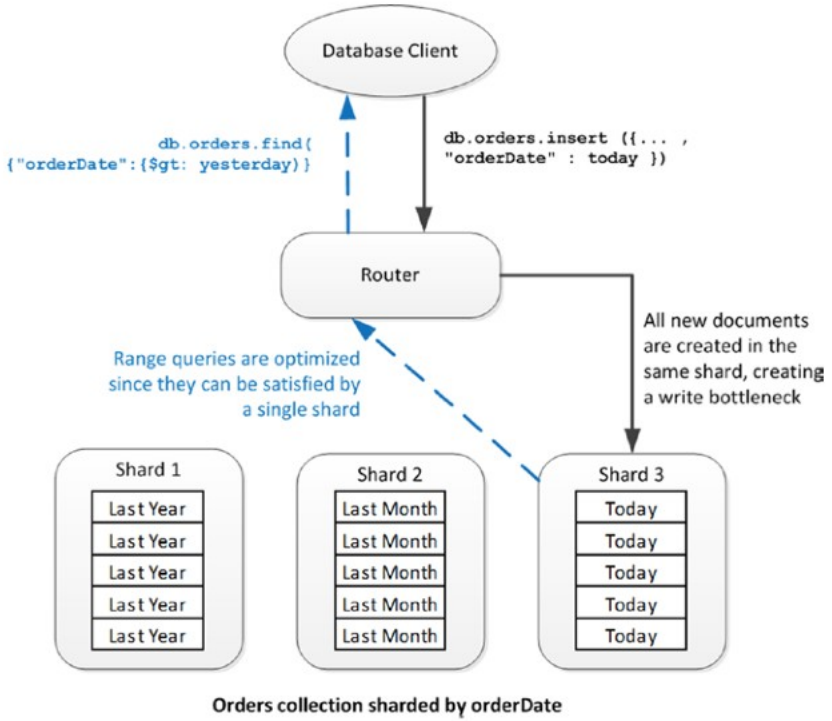




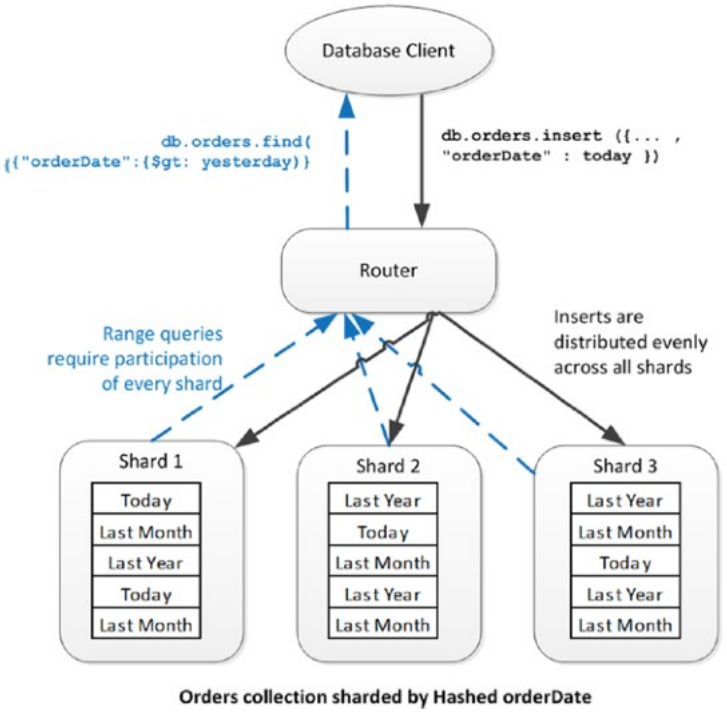
Particionamiento de datos (*sharding*)

- MongoDB soporta al menos dos tipos de sharding:
  - Por rangos de clave (key-based):
    - Más eficiente la lectura de rangos
  - Por hash sobre la clave (hash-based)
    - Mejora la distribución en los inserts

Sharding: ejemplos



Por rangos de clave (key-based)



Por hash sobre la clave (hash-based)

Consistencia y transacciones

- MongoDB implementa *locks*
  - *Cambia la granularidad en versiones, a partir de 3.0 a nivel de documento*
- Consistencia estricta en modo *single node*
- Consistencia eventual en *replica sets*

- A partir de la versión 4.0 soporta transacciones multi-documento
  - *Más info en la Documentación de MongoDB*

Resumiendo

- Las bases de datos de documentos se inspiran en las ideas de las bases OO
- El éxito de las bases documentales JSON se basa fundamentalmente en su alto nivel de adopción por los desarrolladores.
  - Brindan más estructura que las key-value stores.
  - Son muy útiles cuando no existe un esquema fijo (eg: etapas tempranas de desarrollo, esquemas variados)
  - Evitan el *impedance mismatch* OO - relacional

Resumiendo (2)

- MongoDB en particular presenta varias características interesantes:
  - Particionamiento horizontal (*sharding*)
  - Redundancia (*replica sets*)

Material adicional

- Documentación en el sitio de MongoDB
  - https://docs.mongodb.com/

  Cursos online en el sitio de MongoDB
- https://university.mongodb.com/courses/catalog