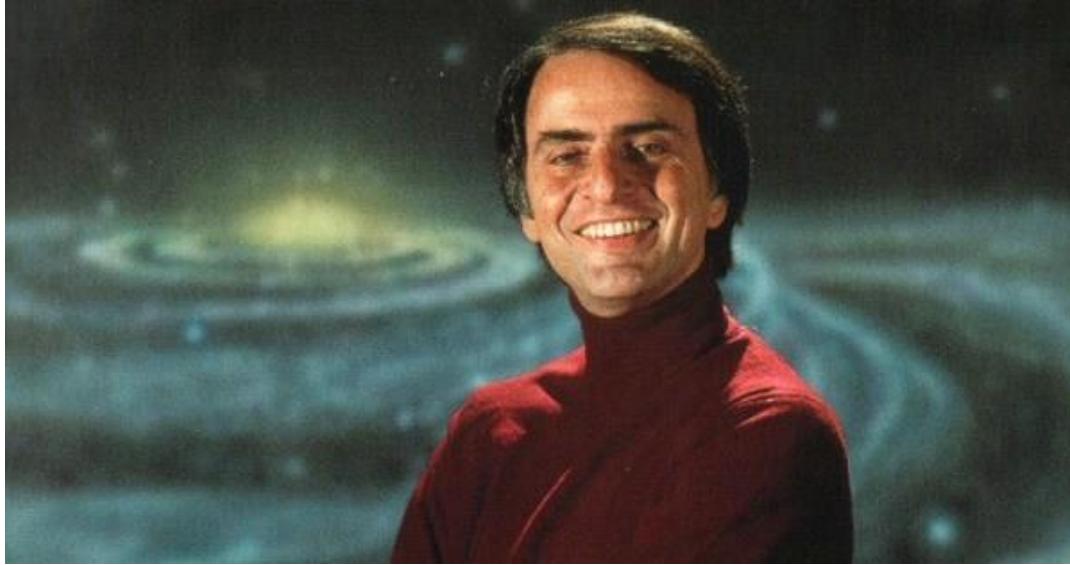




# Bases de datos no relacionales - NoSQL

Computación de Alto Desempeño  
Diego Alberto Rincón Yáñez MCSc  
[diego-rincon@javeriana.edu.co](mailto:diego-rincon@javeriana.edu.co)

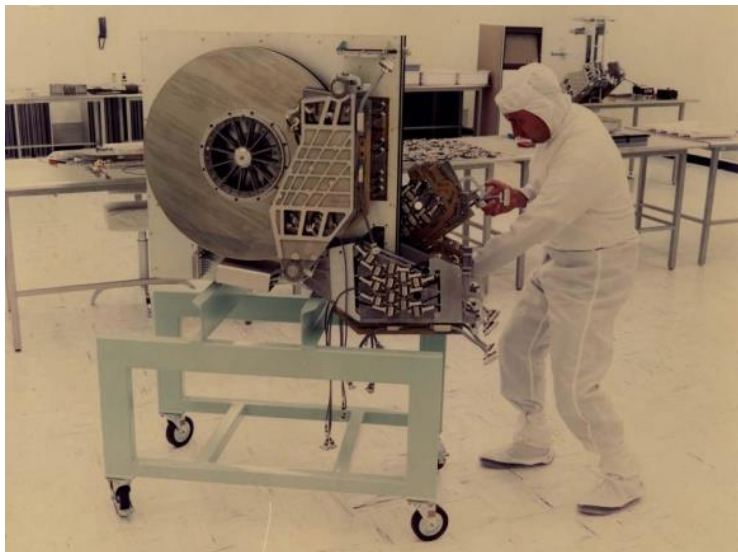


“We live in a society exquisitely dependent on science and technology, in which hardly anyone knows anything about science and technology”



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.

# Orígenes de SQL (1979)



250 Mb  
\$ 81.000/año

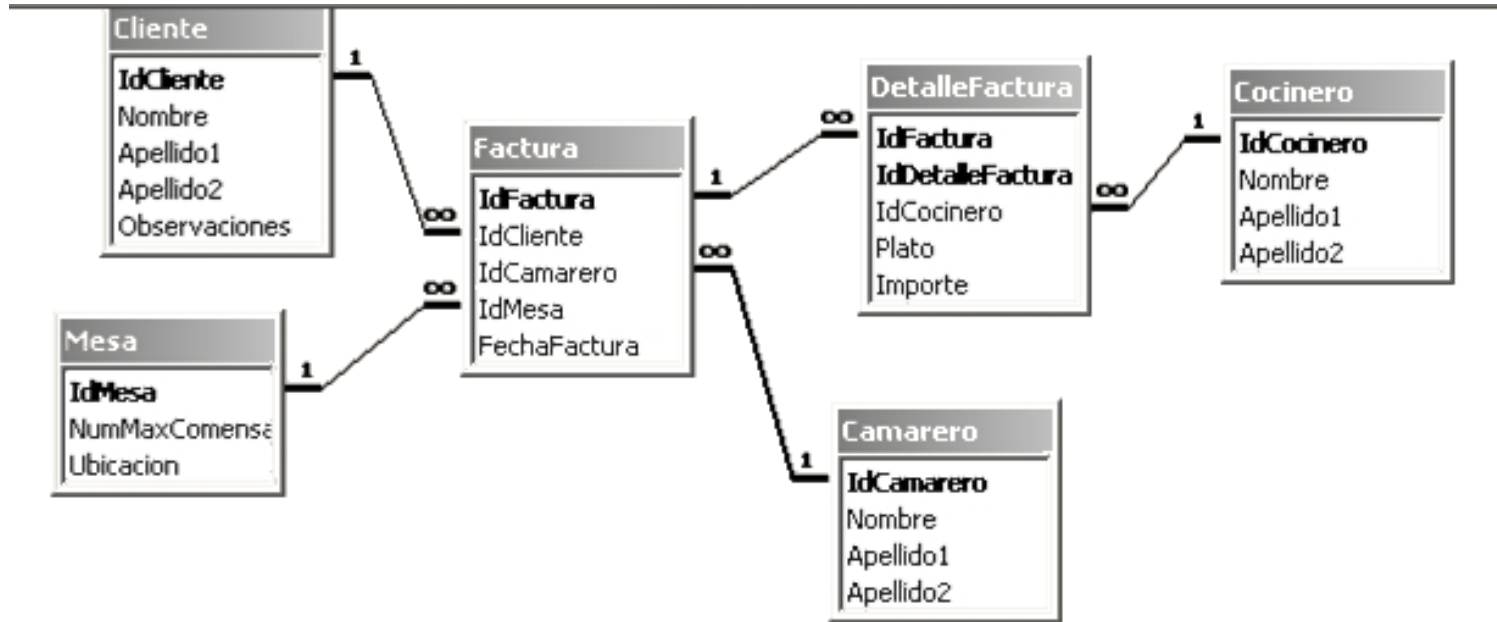
Dennis Ritchie  
Brian Kernighan  
\$ 8.000/año ... (ambos)



Diego Alberto Riri  
Twitter: @d1egoprog.

# Conceptos

## Modelo Entidad Relación



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprogram

# Variedad

- De dónde salen los datos?
- Sensores
- GPS
- Registros históricos
- ...



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.





# Variedad

- Cuándo usar NoSQL?

Los datos no son míos

No controlo lo que los genera ni cada cuánto los genera

No tengo claro qué consultas voy a querer hacer en el futuro



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.

# Agenda

- Introducción al NoSQL
- Técnicas de Modelado
- MongoDB
  - Arquitectura
  - Componentes
  - Alta Disponibilidad
- Taller Práctico



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.

# Introducción a NoSQL



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.



# El mundo ha cambiado

Datos

Riesgo

Tiempo

Costos



Diego Alberto Rincón Yáñez MCSc.  
Twitter: @d1egoprog.

# RDBMS vs NoSQL

Feature	NoSQL Databases	Relational Databases
Performance	High	Low
Reliability	Poor	Good
Availability	Good	Good
Consistency	Poor	Good
Data Storage	Optimized for huge data	Medium sized to large
Scalability	High	High (but more expensive)



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.



	NoSQL	SQL
<b>Model</b>	Non-relational Stores data in JSON documents, key/value pairs, wide column stores, or graphs	Relational Stores data in a table
<b>Data</b>	Offers flexibility as not every record needs to store the same properties	Great for solutions where every record has the same properties
	New properties can be added on the fly	Adding a new property may require altering schemas or backfilling data
	Relationships are often captured by denormalizing data and presenting it in a single record	Relationships are often captured in a using joins to resolve references across tables
	Good for semi-structured data	Good for structured data
<b>Schema</b>	Dynamic or flexible schemas Database is schema-agnostic and the schema is dictated by the application. This allows for agility and highly iterative development	Strict schema Schema must be maintained and kept in sync between application and database
<b>Transactions</b>	ACID transaction support varies per solution	Supports ACID transactions
<b>Consistency</b>	Consistency varies per solution, some solutions have tunable consistency	Strong consistency supported
<b>Scale</b>	Scales well horizontally	Scales well vertically



Diego Alberto Rincón Yáñez MCSc.  
Twitter: @d1egoprog.

# NoSQL

## Qué es?

- X **No** es relacional
- X **No** imponen estructura ni contenido
- X **No** es estándar
- X **No** ACID
- ✓ **Sí** escalable horizontalmente
- ✓ **Sí** muchos datos
- ✓ **Sí** BASE



# ACID vs. BASE

## 1. No ACID

- Atomicidad
- Consistencia absoluta
- Isolation (Aislamiento)
- Durabilidad

## 2. Sí BASE

- Basic Availability
- Soft State: Algunas actualizaciones se pueden perder
- Consistencia Eventual

NoSQL



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.

Autor: Alexandra Pomares PhD

Stop following me, you fucking freaks!



Key-Value



Key Value



Ordered Key-Value



Big Table



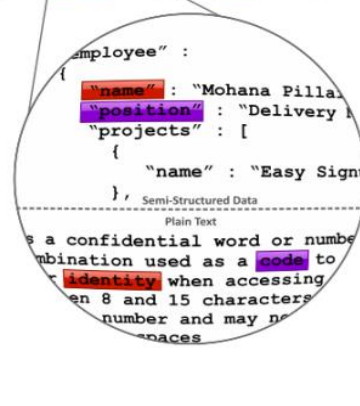
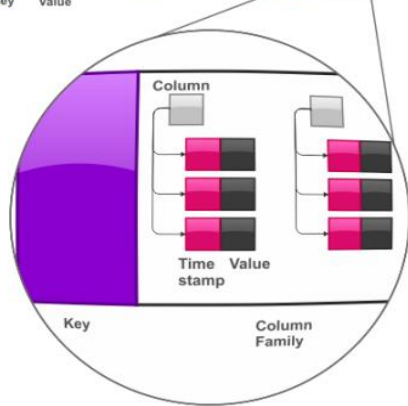
Document,  
Full-Text Search



Graph



SQL

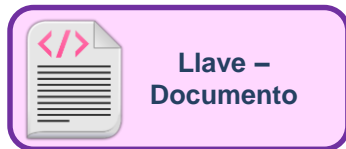


# Tipos de Bases NoSQL

Clave- Valor



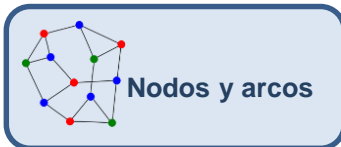
Documentos



Columnar



Grafos

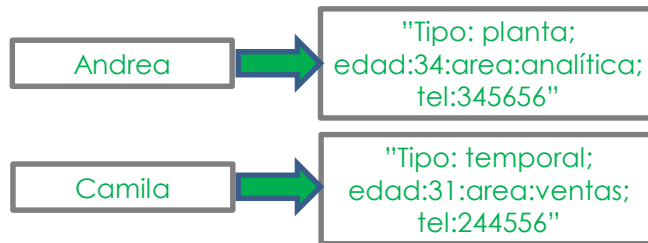


# NoSQL - Clave - Valor

## Modelo de datos



Llave	Valor
nombre	rodrigo
apellido	avella
edad	38





# NoSQL - Clave - Valor

## Cuándo usarla?



**Lectura rápida  
basada en clave**



**Buscar un valor en  
múltiples registros**

### Casos de Uso

1. Caché de páginas web
2. Almacenamiento de sesiones de usuario
3. Almacenamiento de carritos de la compra
4. Caché de base de datos
5. Contadores y estadísticas
6. Listas de elementos recientes
7. Almacenamiento para microservicios



# NoSQL - Documentos

## Modelo de datos



1. Los documentos no necesariamente tienen el mismo conjunto de atributos
2. Los documentos pueden tener atributos que aún no existen
3. Documentos autodefinidos



# NoSQL - Documentos

## NoSQL

### Modelo de datos



**Esto en realidad no es un documento**

**Son Valores compuestos anidados**  
**Colecciones - documentos - campos**

```
{
  "_id" : ObjectId("52f602d787945c344bb4bda5"),
  "nombre" : "Juan",
  "hobbies" : [
    "libros",
    "viajes",
    "fiesta"
  ],
  "amigos" : [
    {
      "nombre" : "Pedro",
      "ocupacion" : "marketing"
    },
    {
      "nombre" : "Juana",
      "ocupacion" : "libretista"
    }
  ]
}
```



# NoSQL - Documentos

## Cuándo usarla?



Aplicaciones se enfocan en unidades autocontenidas



- Cuando se requiere integridad referencial  
- Transacciones con diferentes documentos  
- Se requiere leer sólo un campo del documento

### Casos de Uso

1. Recibir los datos cuando no se conoce con anticipación la estructura - **Landing zone**
2. Guardar preferencias de usuario
3. Catálogos de productos
4. Registro de incidentes
5. ...



# NoSQL basado Columnas

## Modelo de datos


Familia de columna

1. Cada integrante de una familia de columnas puede tener un esquema diferente
2. Semi-esquema

## KeySpace

### Column Family

Key	Column Name	Column Name	Column Name
	Value	Value	Value

Key	Column Name	Column Name
	Value	Value

Key	Column Name	Column Name	Column Name	Column Name
	Value	Value	Value	Value

### Column Family

Key	Column Name	Column Name	Column Name
	Value	Value	Value

Key	Column Name	Column Name
	Value	Value

## KeySpace

--



# NoSQL basado Columnas



```
UserProfile={  
  Cassandra={ emailAddress:"casandra@apache.org", age:"20"}  
  TerryCho= { emailAddress:"terry.cho@apache.org", gender:"male"}  
  Cath= { emailAddress:"cath@apache.org", age:"20",gender:"female",address:"Seoul"}  
}
```

Relational Model	Cassandra Model
Database	Keyspace
Table	Column Family (CF)
Primary key	Row key
Column name	Column name/key
Column value	Column value

Basado en Jay Patel -2013



# NoSQL basado Columnas

## Column family tweets

Column family: tweets

1234e534-9000-67df	Text	User_id	Date
	Me gusta hacer deporte	3456	06/12/2016

2344e534-9080-56gt	Text	User_id	Date	In Reply to
	De acuerdo con	3456	06/12/2016	5687



# NoSQL basado Columnas

## Cuándo usarla?



-Donde pueden  
haber datos dispersos  
-Consultas sobre  
campos específicos



ACID requerido

### Casos de Uso

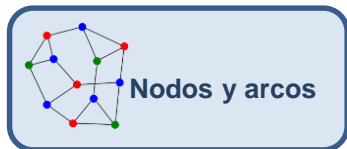
1. Sistemas de recomendación: Spotify
2. Registro de eventos
3. Motores de búsqueda





# NoSQL Grafos

## Modelo de datos



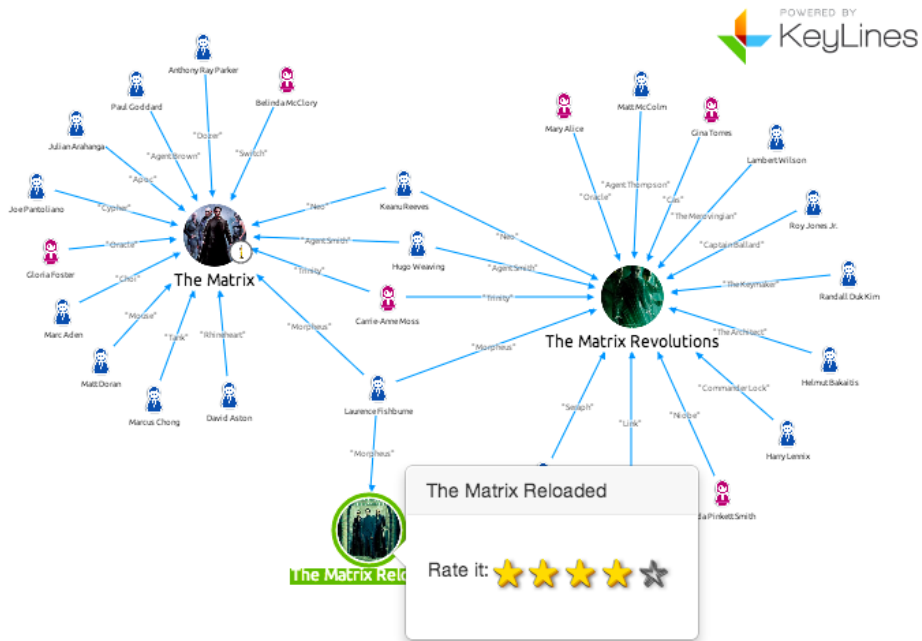
Los arcos pueden representar muchas cosas:

- Proximidad en tiempo
- Proximidad en espacio
- Relación

1. Busca representar relaciones entre cosas
2. Nodos, arcos y propiedades



# NoSQL Grafos



# NoSQL Grafos

## Cuándo usarla?



**Se necesita explorar  
cómo están relacionados  
elementos**



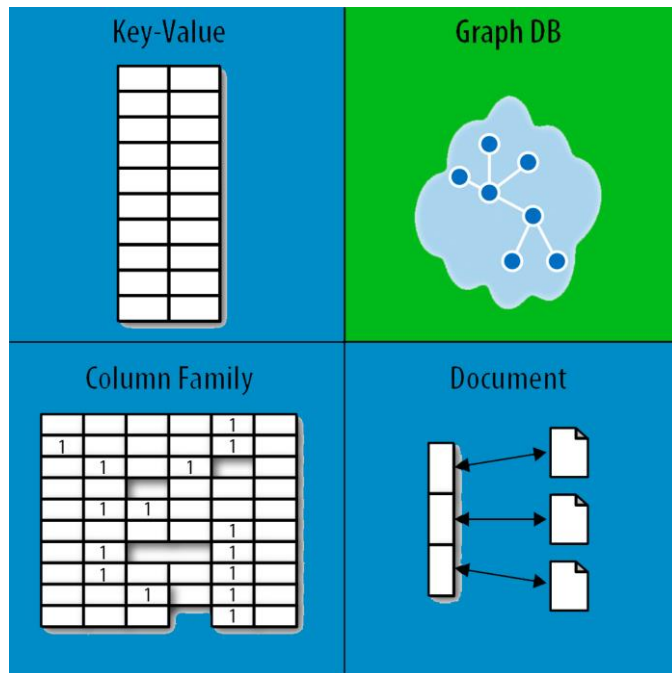
**-Se requieren actualizaciones  
masivas  
-Alta distribución de datos**

### Casos de Uso

1. Análisis de redes sociales
2. Redes de fraude
3. Redes de palabras para detección de patrones de correferencia
4. Logística
5. Caminos mínimos
6. Relaciones fuertes



# NoSQL en un sólo vistazo



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.

# Herramientas NoSQL

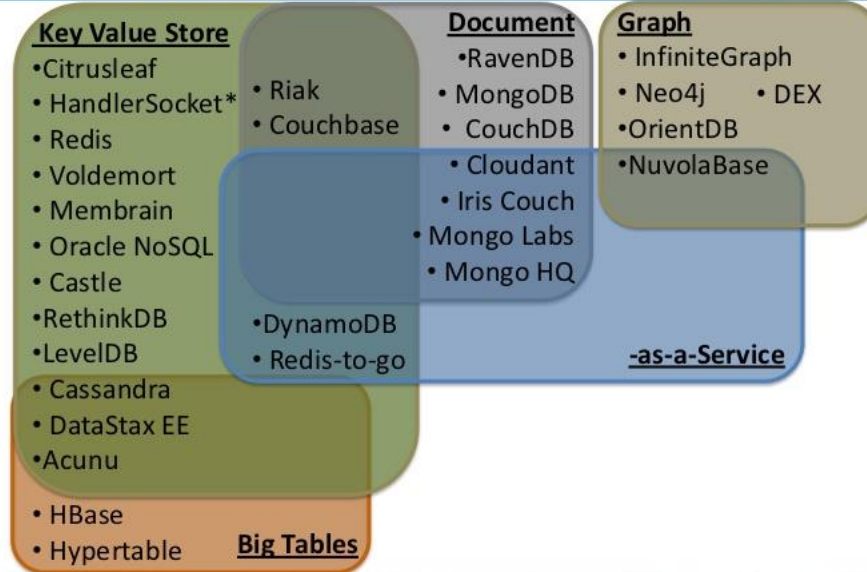
DOCUMENTOS	GRAFOS
  	 
CLAVE-VALOR	COLUMNAS EXTENDIDAS
   	    

AUTOR: PABLO SCIOLLA – VERSIÓN: MARZO 2017

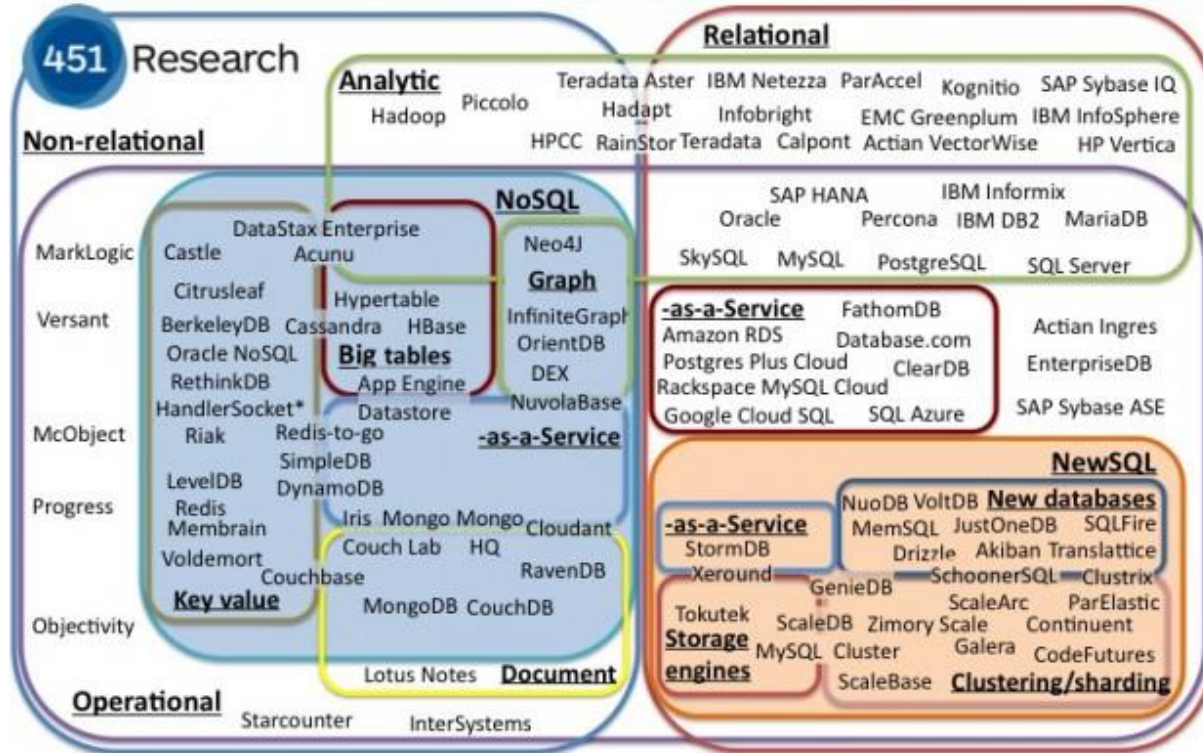


Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.

## The NoSQL Ecosystem



## The evolving database landscape



© 2012 by The 451 Group. All rights reserved



Diego Alberto Rincón Yáñez MSc.

Twitter: @d1egoprog.



# Métricas de Diseño



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprogram

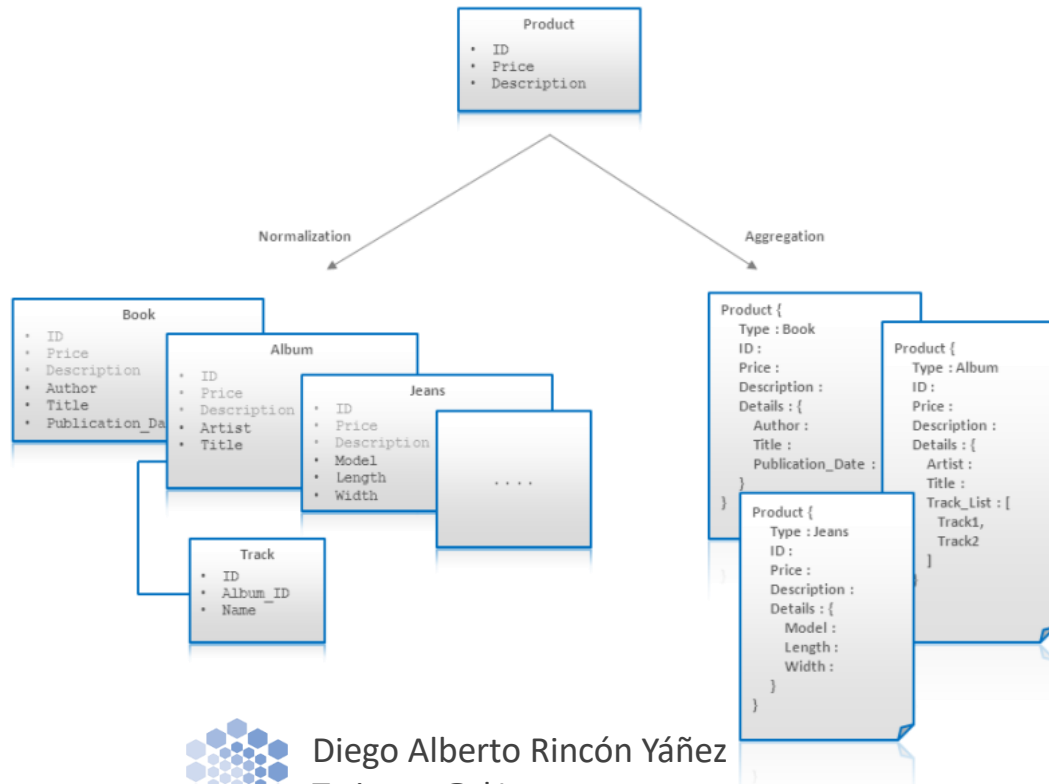


# Técnicas Conceptuales

- Denormalization
- Aggregates
- Application Side Joins
- Atomic Aggregates
- Index Table
- Otras 10.....

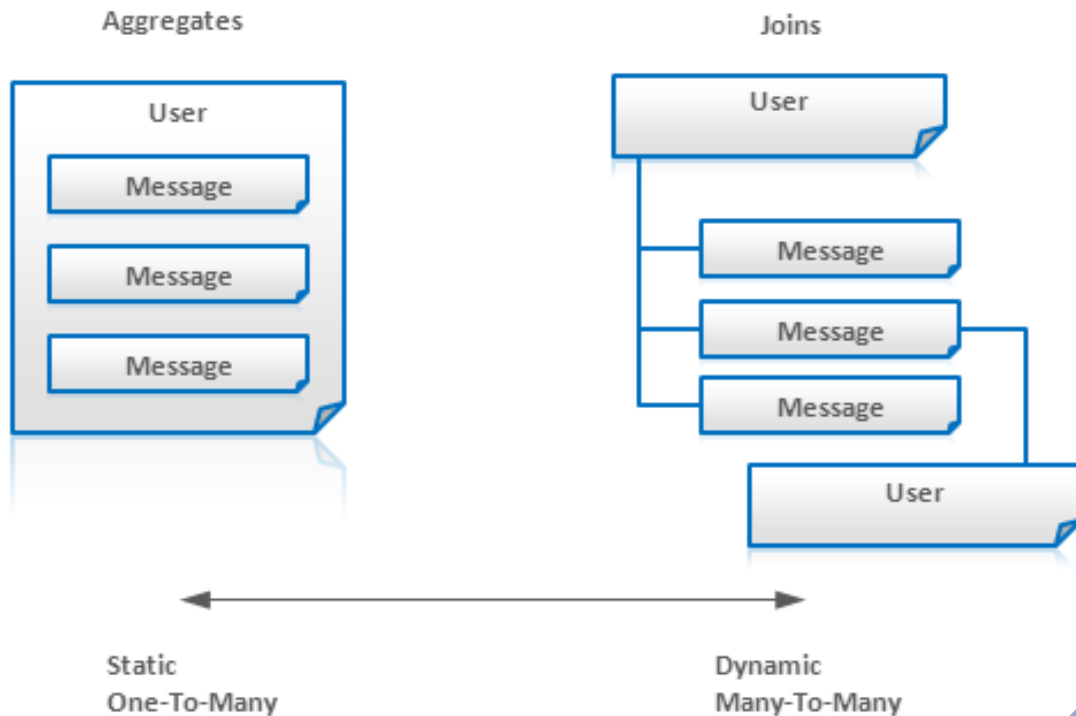


# Agregación



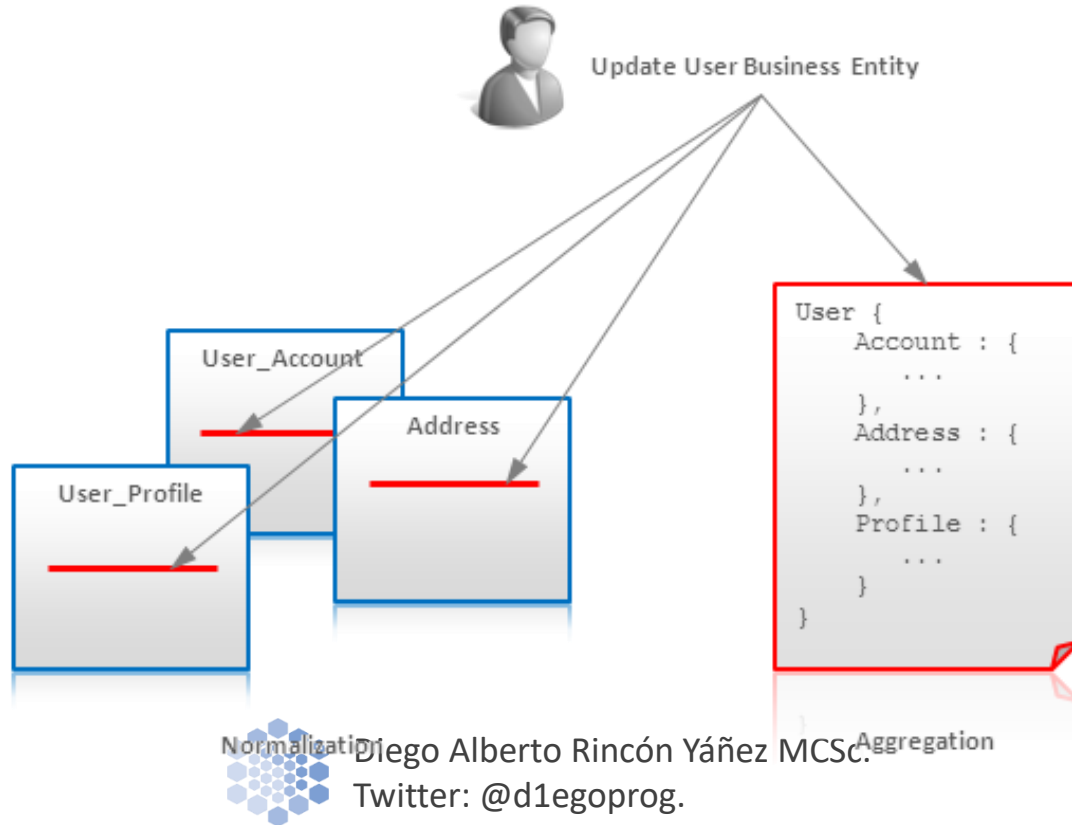
Diego Alberto Rincón Yáñez  
Twitter: @d1egoprog.

# “Joins” Laterales



Diego Alberto Rincón Yáñez MCSc.  
Twitter: @d1egoprog.

# Agregados Atómicos



# Tabla de Indices

The diagram illustrates a database structure with a main table and an index table. The **Users** table contains user information, and the **Cities** index table maps cities to their corresponding user IDs.

UserID	Info
7734	City:San Francisco, email:alef@gmail.com
4667	City:New York, email:jsample@yahoo.com
6578	City:Seattle, email:knovoselic@gmail.com
1263	City:San Francisco, email:jgray@yahoo.com

City	UserIDs
San Francisco	7734, 1263, ...
New York	4667, ...
Seattle	6578, ...



# Agenda

- **Introducción al NoSQL**
- **Técnicas de Modelado**
- MongoDB
  - Arquitectura
  - Componentes
  - Alta Disponibilidad
- Taller Práctico



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.



# MongoDB



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.

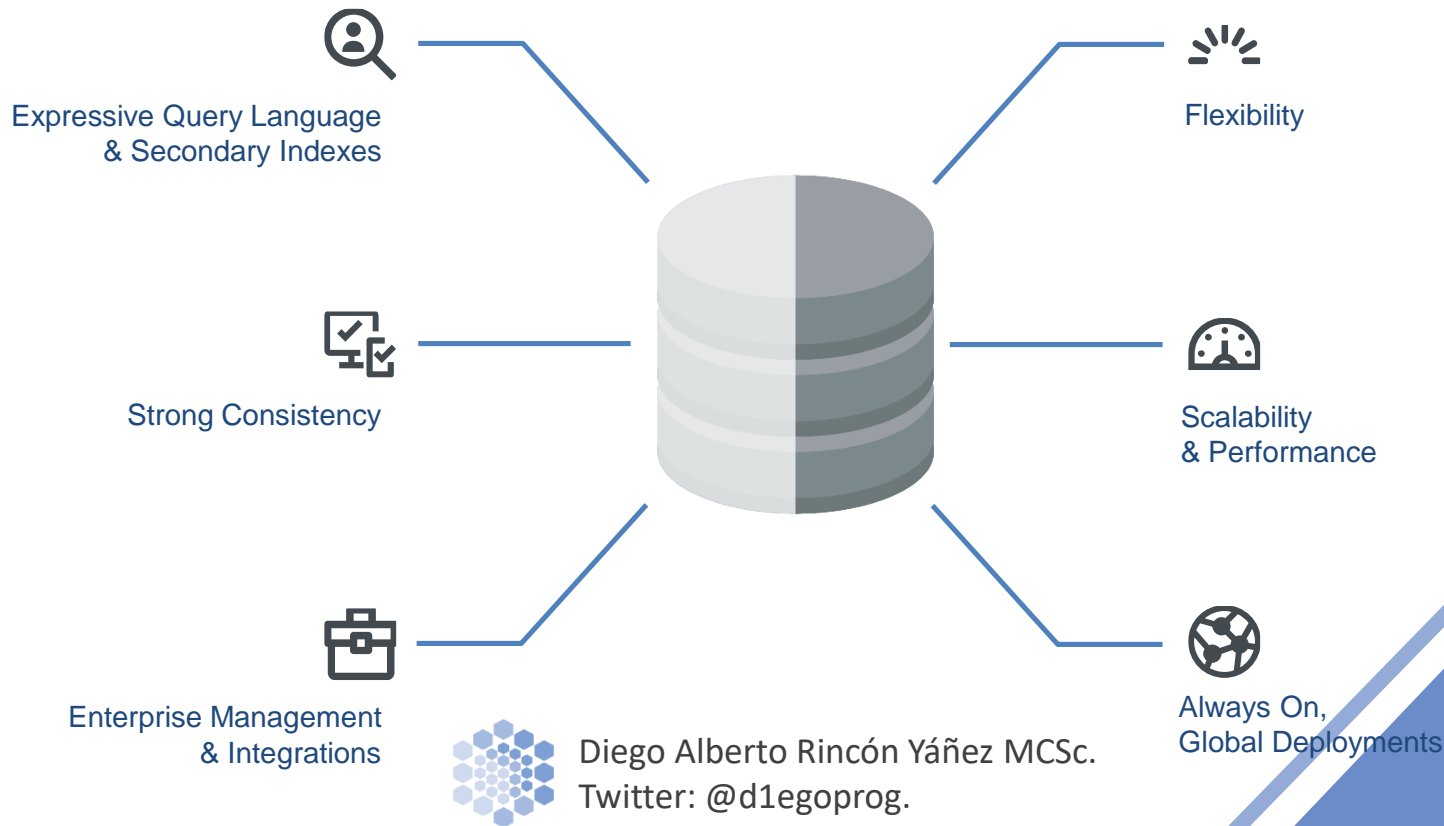
# Arquitectura Nexus



Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.



# Arquitectura Nexus



# Arquitectura Nexus

- **Expressive query language & secondary Indexes:** Los usuarios deben poder acceder a sus datos, tanto operacional como analítica, soporte de índices.
- **Strong consistency:** Acceso inmediato a actualizaciones realizadas, bajar el nivel de los modelos de consistencia (Nivel Operacional).
- **Enterprise Management and Integrations:** Integraciones a nivel empresarial: seguridad, automatización, procesos, operaciones y Equipos (DBA, Ingenieros).

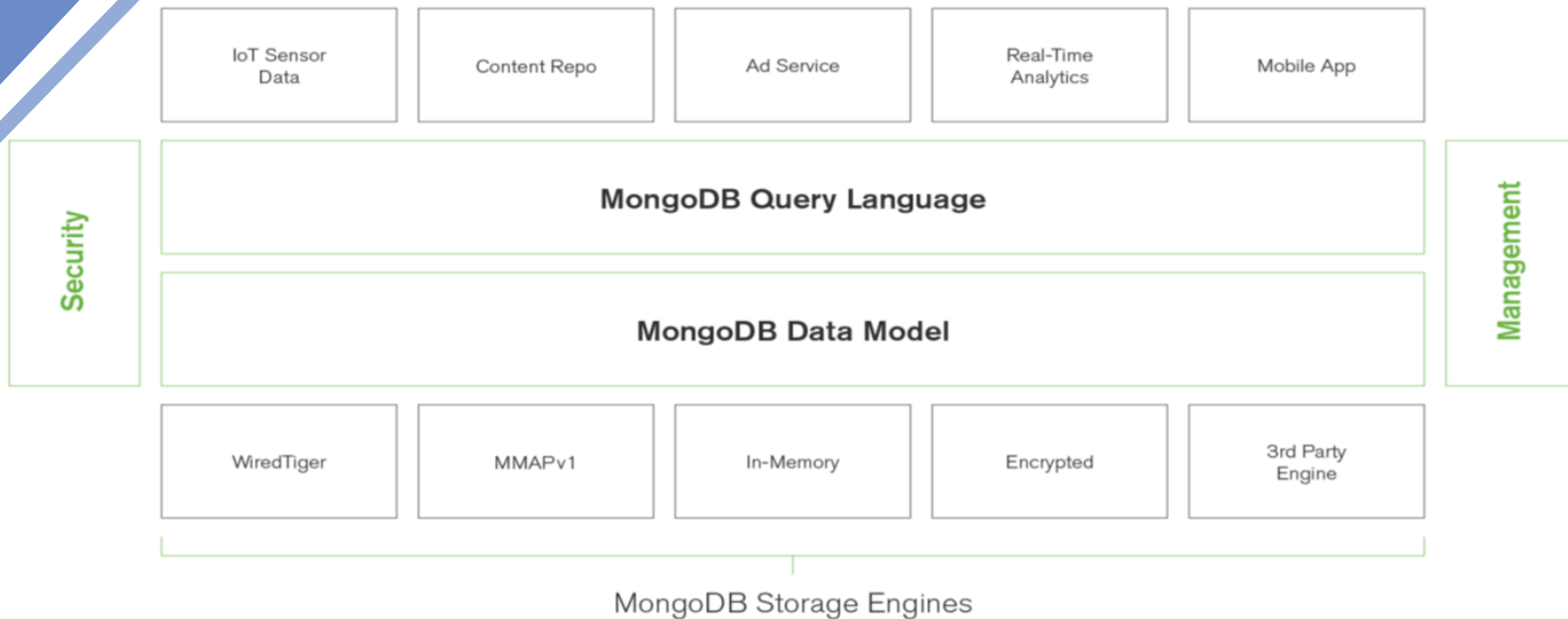


# Arquitectura Nexus

- **Flexible Data Model:** Permite combinar datos de cualquier estructura, permitiendo dinamismo, sin tiempos de indisponibilidad.
- **Scalability and Performance:** Despliegue en hardware de tipo “commodity”, crecimiento horizontal ilimitado.
- **Always-On Global Deployments:** permite el despliegue inter organización o data center, geográficamente distribuido, manteniendo los anteriores



# Arquitectura Nexus



Diego Alberto Rincón Yáñez MCSc.  
Twitter: @d1egoprog.

# Tipos de Consulta

- **Key-value:** Devuelve el resultado de una consulta básica según el contenido del valor .
- **Range:** Devuelve una colección resultado de un límite entre dos valores
- **Geospatial:** Retorna resultados basados en proximidad de un punto a una línea.
- **Search:** Retorna datos orientados al grupo de resultados, pueden utilizar operadores booleanos clásicos como AND, OR, NOT.
- **Aggregation Framework:** Retornan conteos, agregaciones o transformaciones, similares a las sentencias SQL.
- **JOINS y graph traversals:** Por medio de \$lookup las colecciones pueden ser combinadas como una operación left outer JOIN.
- **MapReduce:** Consultas ejecutadas expresadas en JavaScript que se procesan en toda la base de datos

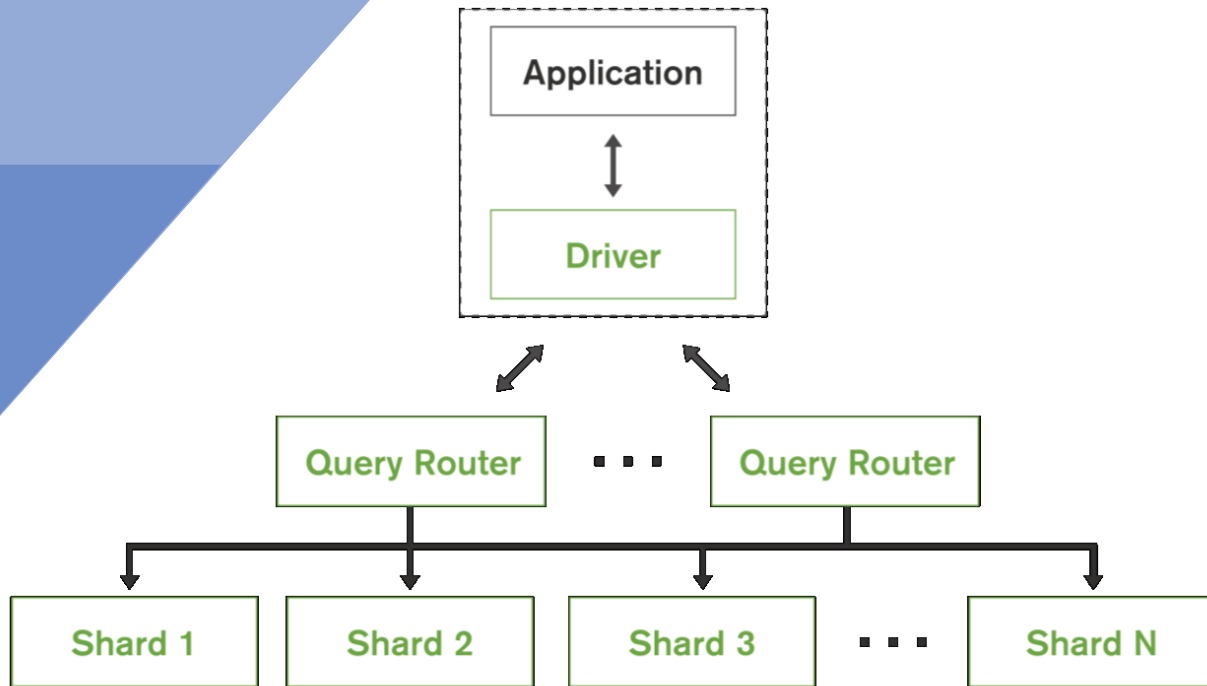


# Tipos de Alta Disponibilidad

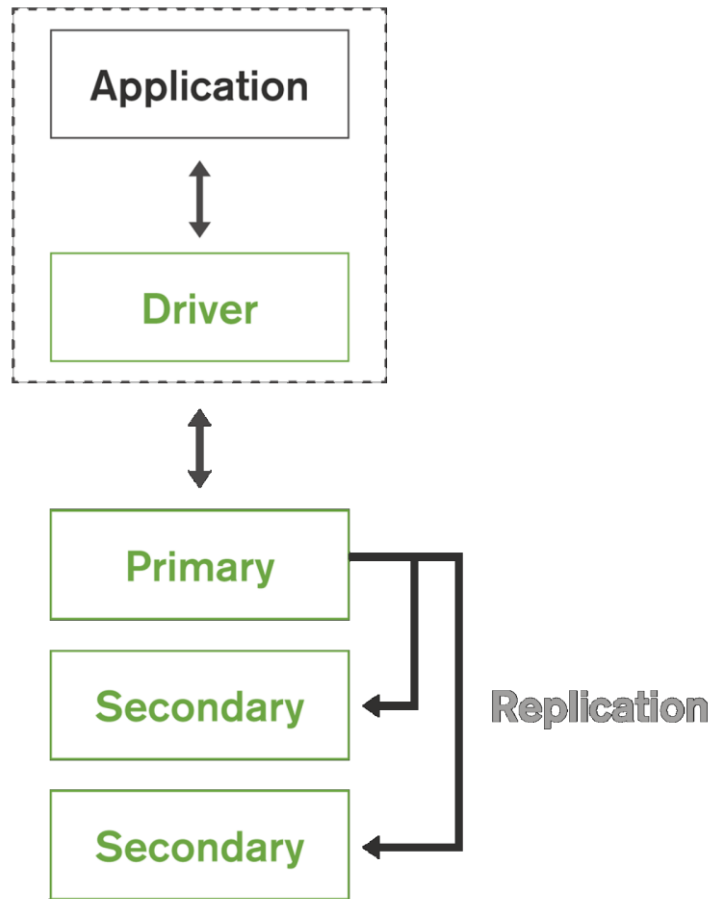
- Sharding
  - Range
  - Hash
  - Zone
- Replication
  - Ft. Replica Set Oplog



# Sharding



# Replication



Diego Alberto Rincón Yáñez MCSc.  
Twitter: @d1egoprog.



# Herramientas

- Compass
- Ops Manager
- Stich

<https://docs.mongodb.com/ecosystem/>  
<http://mongodb-tools.com/>

**Proj Tip:**

<https://docs.mongodb.com/ecosystem/use-cases/storing-log-data/>



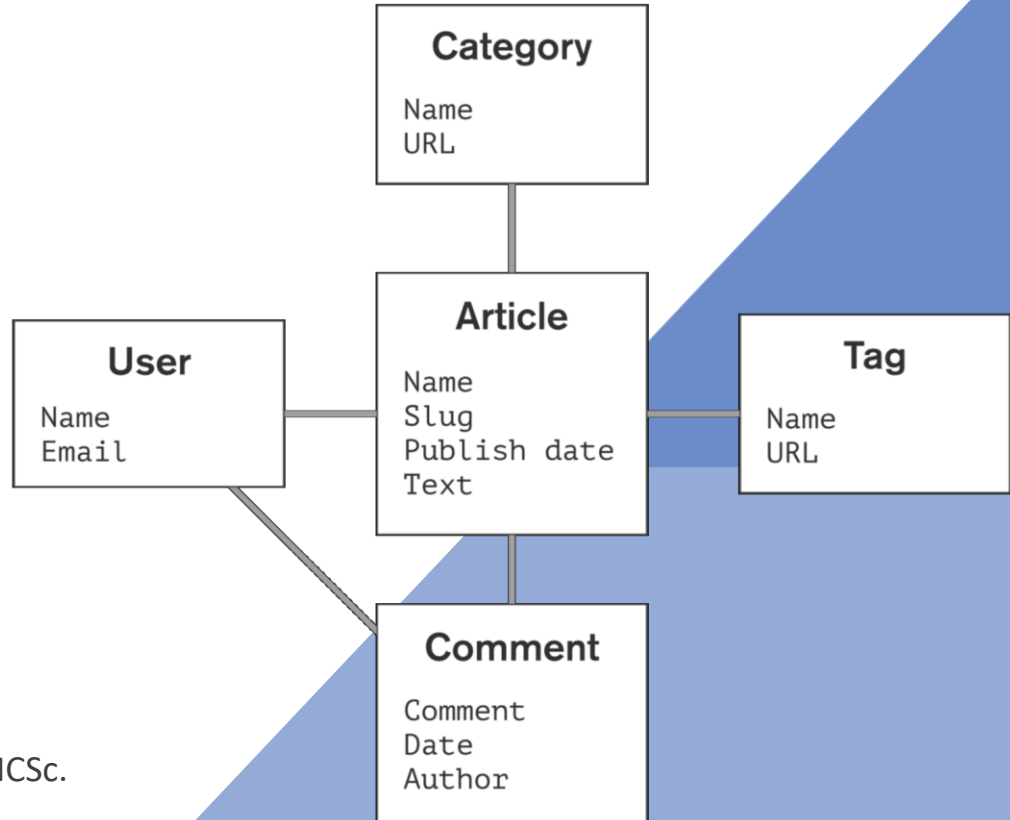
Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.

# Diferencias 101

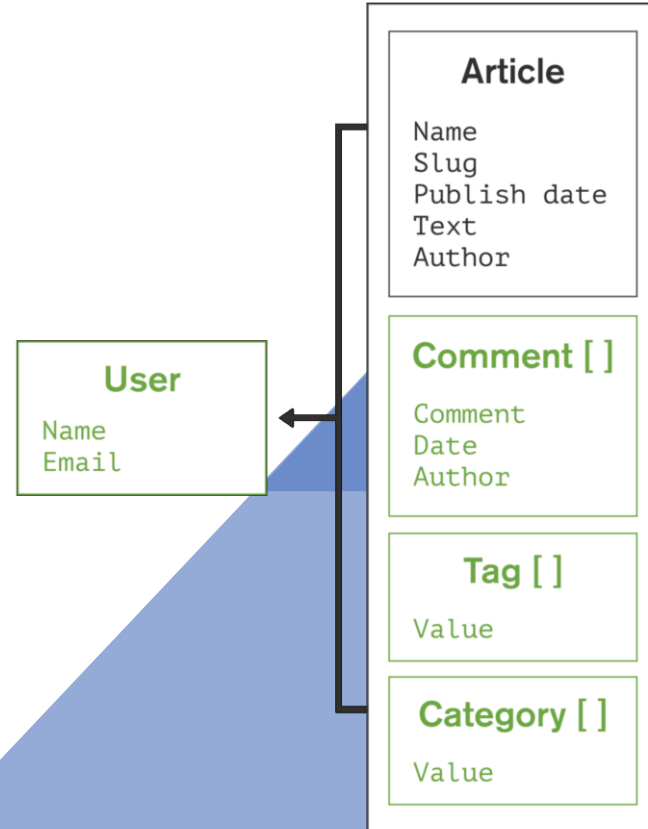
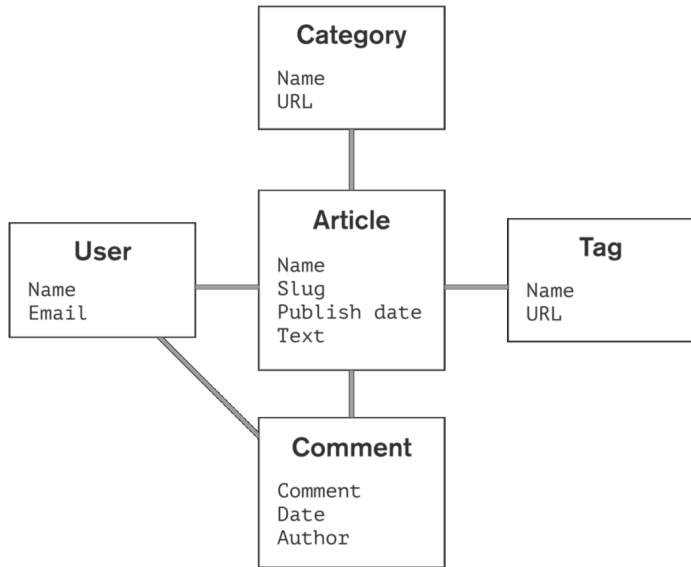
RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
Column	Field
Join	Embedded Documents
Primary Key	Default key “_id”



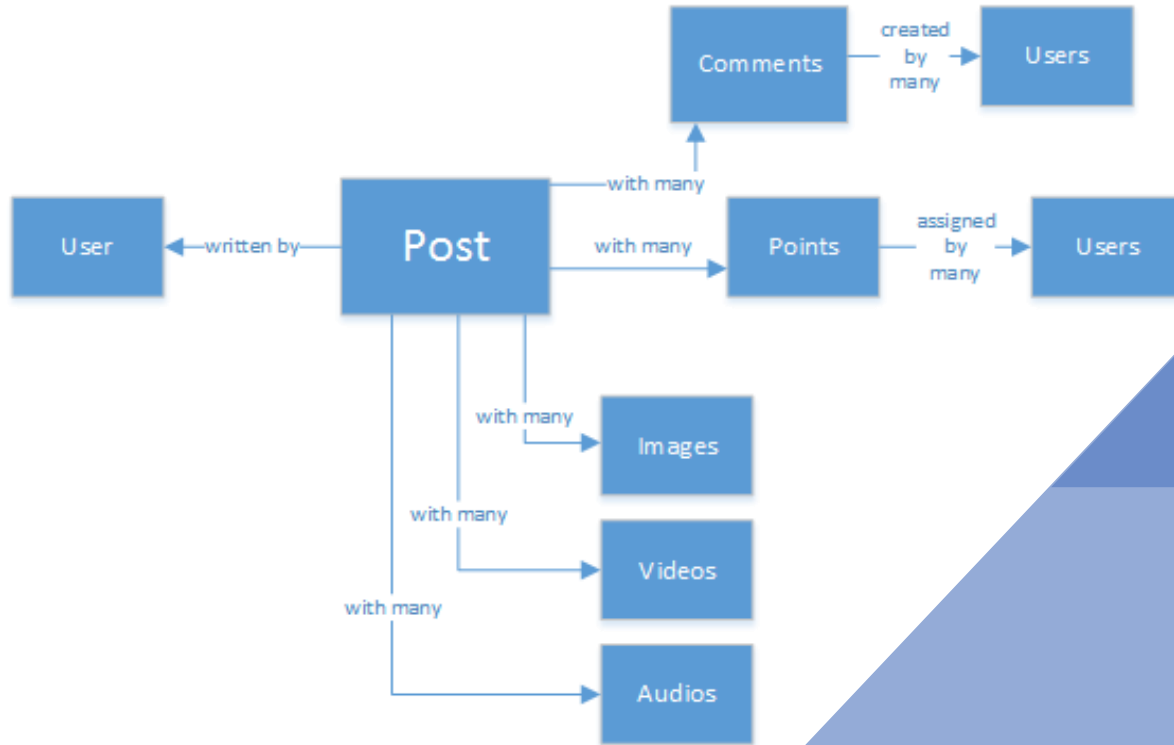
# Ejemplo 1 (MER Tradicional)



# Ejemplo 1 NoSQL



# Ejemplo 2 (Social)



# Ejemplo 2

## JSON

```
{  "id": "ew12-res2-234e-544f",
  "title": "post title",
  "date": "2016-01-01",
  "body": "this is an awesome post stored on NoSQL",
  "createdBy": "User",
  "images": [
    "http://myfirstimage.png",
    "http://mysecondimage.png"
  ],
  "videos": [
    { "url": "http://myfirstvideo.mp4", "title": "The first video" },
    { "url": "http://mysecondvideo.mp4", "title": "The second video" }
  ],
  "audios": [
    { "url": "http://myfirstaudio.mp3", "title": "The first audio" },
    { "url": "http://mysecondaudio.mp3", "title": "The second audio" }
  ]
}
```





Questions  
are  
guaranteed in  
life;  
Answers  
aren't.

¿Preguntas?



Diego Alberto Rincón Yáñez MCSc.  
Twitter: @d1egoprog.

# Taller

- ✓ Descargar e Instalar
  - MongoDB
  - Compass
- ✓ Probar el código JAVA y ver rendimiento
  - Simple, muchos y paralelo
- ? Modificar el código para insertar por bloques de 10k de documentos y verificar rendimiento.
- ? Extra: Preguntar por el reto.



Diego Alberto Rincón Yáñez MCSc.  
Twitter: @d1egoprog.

