

Bases de datos NoSQL

Introducción

- ▶ El término NoSQL es la combinación literal de No y SQL.
- ▶ Puede interpretarse como: *Not at all SQL*, *No relational DBMS* o *Not only SQL* (acepción más común).
- ▶ Cualquiera que sea la interpretación, el término se usa para abarcar a todos los DBMS que son no relacionales.
- ▶ Surgen, en principio, en el contexto del procesamiento masivo de datos en la Web.
- ▶ Después incorporaron otras fuentes de datos: sensores, GPS, celulares...; los cuales generan cantidades masivas de datos.
- ▶ Los RDBMS mostraron problemas relacionados con: eficiencia, paralelización, escalabilidad y costos.

Un poco de historia

- ▶ Google fue pionero en la solución a esos problemas de los RDBMS.
- ▶ Construyó (~2002) una infraestructura escalable para el procesamiento en paralelo de grandes cantidades de datos.
- ▶ Creó un ambiente consistente en:
 - Un sistema distribuido de archivos
 - Un almacenamiento de datos orientado a columnas
 - Un sistema de coordinación distribuido y
 - Un ambiente de ejecución de algoritmos paralelos basados en MapReduce.
- ▶ Posteriormente (~2006), entre los creadores de Lucene, una primera versión libre (open source) de la infraestructura de Google, y personal de Yahoo crearon Hadoop, un sistema que brinda todas las características del sistema de Google.

Un poco de historia (cont.)

- ▶ Después (~2007), Amazon presenta sus ideas sobre un sistema de almacenamiento de datos, Dynamo, distribuido, altamente disponible y eventualmente consistente.
- ▶ A partir de Google y Amazon, nuevos productos se han desarrollados tanto por grandes corporaciones como por empresas nuevas.
- ▶ Gigantes de la computación, como: Facebook, Yahoo, IBM, Oracle, etc., y de otros ámbitos, como: Netflix, Ebay, etc., han desarrollado productos y presentado casos de éxito relacionados con el procesamiento masivo de datos (Big Data).

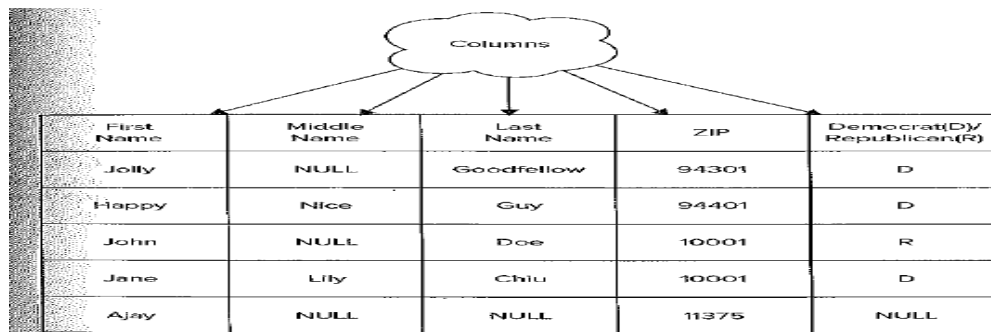
Tipos de bases de datos NoSQL

- ▶ Tres conceptos claves surgieron para el procesamiento masivo de datos:
 1. Los datos necesitan ser almacenados en un sistema de archivos en red que pueda ser expandido en múltiples máquinas.
 2. Los datos necesitan ser almacenados en una estructura que proporcione más flexibilidad que las estructuras normalizadas de una base de datos relacional.
 3. Los datos necesitan ser procesados de manera que los cálculos puedan ser realizados en subconjuntos aislados y luego combinados para generar los resultados deseados.

Tipos de bases de datos NoSQL (cont.)

► Orientadas a columnas ordenadas

- Pensando en términos relacionales se podría tener una tabla como la siguiente:



First Name	Middle Name	Last Name	ZIP	Democrat(D)/Republican(R)
Jolly	NULL	Goodfellow	94301	D
Happy	Nice	Guy	94401	D
John	NULL	Doe	10001	R
Jane	Lily	Chiu	10001	D
Ajay	NULL	NULL	11375	NULL

- ¿Qué pasa si la tabla evoluciona agregándole otros atributos como: Street Address y Veg/Non-veg?
La tabla podría contener muchos nulos en estos atributos para filas ya existentes.
- ¿Y qué pasa si se quieren guardar diferentes versiones de los datos?
El manejo de la tabla tendría que ser más complejo agregando una característica de tiempo para las distintas versiones.

Tipos de bases de datos NoSQL (cont.)

► Orientadas a columnas ordenadas

- La idea de estas bases de datos NoSQL es crear familias de columnas para manejar los datos. Ejemplo:

	name	location	preferences
	first name=>"...", last name=>"..."	zip=>"..."	d/r=>"...", veg/non-veg=>"..."

- Además de la flexibilidad de agregar columnas, sin nulos, se puede agregar el tiempo (para las versiones)

	time	name	location	preferences
	t9	first name=>"...", last name=>"..."	zip=>"..."	d/r=>"...", veg/non-veg=>"..."
	t8			
	t7			
	t5			

FIGURE 1.1

Tipos de bases de datos NoSQL (cont.)

► Orientadas a columnas ordenadas

- No es necesario que existan todos los datos para una misma “entidad”; algunas filas pueden existir y otras no (el row-key se usa para hacer el ordenamiento; normalmente lo genera el sistema):

row-key	time	name
	t9	
	t8	
	t7	
	t5	

row-key	time	location
	t9	
	t8	

row-key	time	preferences
	t9	
	t7	

Tipos de bases de datos NoSQL (cont.)

► Almacenamientos clave/valor

- Son bases de datos que almacenan los datos usando HashMap (arreglos asociativos) guardando pares clave/valor (se diferencian de las anteriores en que normalmente no hay ordenamiento, aunque pueden generarse índices para tal efecto).
- Son sistemas que tienen un uso intensivo y extensivo de memoria, principalmente cache, donde guardan los datos más utilizados.
- Utilizando el ejemplo anterior, los pares clave/valor podrían ser así, con los primeros dos niveles de claves (se usa notación JSON):

```
{ "row_key_1" : {  
  "name" : {  
    ... },  
  "location" : {  
    ... },  
  "preferences" : {  
    ... }  
  },  
  "row_key_3" : {  
    ...  
  }, ...  
}
```

```
"row_key_2": {  
  "name" : {  
    ... },  
  "location" : {  
    ... },  
  "preferences" : {  
    ... }  
},
```

Tipos de bases de datos NoSQL (cont.)

► Almacenamientos clave/valor

◦ Un tercer nivel de clave sería:

```
{ "row_key_1" : {  
  "name" : {  
    "first_name" : "Juan",  
    "last_name" : "Gómez" },  
  "location" : {  
    "zip" : "94301" },  
  "preferences" : {  
    "d/r" : "D" }  
},  
  "row_key_2" : {  
    "name" : {  
      "first_name" : "Rocío", "middle_name" : "R",  
      "last_name" : "Real" },  
    "location" : {  
      "zip" : "10001" },  
    "preferences" : {  
      "v/nv" : "V" }  
},  
  "row_key_3" : {  
    ...  
  }, ...  
}
```

Tipos de bases de datos NoSQL (cont.)

▶ Almacenamientos clave/valor

- Y un cuarto nivel de clave podría implicar al tiempo:

```
{ "row_key_1" : {  
  "name" : {  
    "first_name" : {  
      1 : "Juan" },  
    "last_name" : {  
      1 : "Gómez" }  
  },  
  "location" : {  
    "zip" : {  
      1 : "94301" }  
  },  
  "preferences" : {  
    "d/r" : {  
      1 : "D",  
      5 : "R" }  
    }  
  }, ...  
}
```