

Introducción a bases de datos no relacionales: NoSQL

Ampliación a Bases de Datos

Profesor: Pablo Ramos

pablo.ramos@u-tad.es

INTRODUCCIÓN: Tipos de datos

- No estructurados
 - Documentos de lenguaje natural
 - Audios y videos
- Semi-estructurados
 - Páginas web
 - XML/JSON (pueden estar estructurados)
- Estructurados
 - Bases de datos

INTRODUCCIÓN: Tipos de datos

- No estructurados
 - Necesitas procesos inteligentes para poder acceder a la información contenida.
 - Minería de datos y aprendizaje automático
- Estructurados
 - La información es accesible de forma rápida a través de un sistema o método de consulta
 - SQL

INTRODUCCIÓN: BBDD relacionales

- La importancia de las BBDD relacionales
 - Persistencia:
 - Problema:
 - Almacenaje en disco
 - ¿Archivos?
 - Solución:
 - BBDD facilita el acceso a datos específicos.
 - Método de acceso: Consultas

INTRODUCCIÓN: BBDD relacionales

- La importancia de las BBDD relacionales
 - Concurrencia
 - Problema:
 - Acceso a datos al mismo tiempo.
 - Se pueden corromper los datos.
 - Solución:
 - Transacciones
 - » Las consultas se encapsulan en transacciones atómicas.
 - » Las consultas se ejecutan por completo sin que otras consultas se ejecuten entre medias (**commit**) o se rechazan y se vuelve al estado inicial (**abort and rollback**).

INTRODUCCIÓN: BBDD relacionales

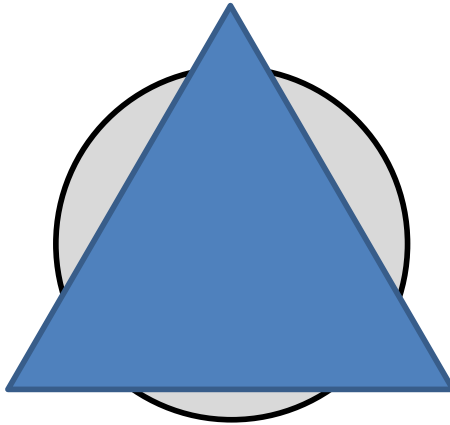
- La importancia de las BBDD relacionales
 - Concurrencia
 - Solución:
 - Transacciones: Propiedad **ACID**
 - » **Atomicidad**: Se ejecuta todo o nada.
 - » **Consistencia**: Solo se almacena información válida. La integridad de los datos debe mantenerse una vez finalizado.
 - » **Aislamiento**: Asegura que la ejecución de consultas concurrentes tengan como resultado el mismo que el de una ejecución secuencial.
 - » **Durabilidad**: Una vez finalizado, los datos permanecerán de forma indefinida, incluso después de un fallo del sistema.

INTRODUCCIÓN: BBDD relacionales

- La importancia de las BBDD relacionales
 - Integración
 - Problema:
 - Acceso a la información desde múltiples aplicaciones.
 - Solución:
 - Interfaz de acceso común para todas las aplicaciones y usuarios.
 - Modelo estandarizado
 - Problema:
 - Mismo problema para multitud situaciones.
 - Solución
 - Normalización del modelo relacional: Modelo y metodología de consulta altamente extendida.

INTRODUCCIÓN: Impedance Mismatch

- X Relational Mapping (XRM)
 - Proceso por el cual se crea una relación entre la estructura del modelo relacional y el tipo (X) de modelo de datos en memoria.
 - PROBLEMA



- Existen diferencias estructurales que dificultan encontrar una relación óptima entre una base de datos relacional normalizada y la forma de organizar los datos en memoria.
 - e.g. Jerarquía de clases en orientación a objetos (ORM)

INTRODUCCIÓN: Impedance Mismatch

- Object Relational Mapping (ORM)
 - PROBLEMA
 - Ejemplos de estructuras de datos no modelizables con la misma estructura que en memoria.
 - Entradas anidadas.
 - Listas de entradas.
 - Solución del modelo relacional:
 - Creación de tablas adicionales.

<u>Profesor</u>	
+ Nombre:	Pablo
+ Profesor:	Sí
+ Alumnos:	
	- Anita
	- Pepito
	- Juanito

relacional →

Profesores	
Nombre	Profesor
Pablo	Sí

Alumnos
Nombre
Anita
Pepito
Juanito

INTRODUCCIÓN: La caída del imperio relacional

- Aparición de capas intermedias para la comunicación con BBDD.
 - Problema:
 - “The impedance mismatch” es uno de los principales motivos del declive de las BBDD relacionales.
 - Solución:
 - Crear una interfaz que elude las restricciones intrínsecas de las BBDD relacionales.
 - Problema:
 - Acceso concurrente (e.g. Web services)
 - Solución:
 - Coordina el acceso concurrente.

iPARCHE!

INTRODUCCIÓN: La caída del imperio relacional

- El ataque de los clústeres.
 - Internet ha propiciado un nuevo modelo de negocio en el que toda la información recopilable es valiosa (Big Data)
 - Incremento del tamaño de las bases de datos:
 - **Escalado vertical:** Máquinas más grandes y potentes
 - Limitación del rendimiento y alto coste económico
 - **Escalado horizontal (clústeres):** Multitud de máquinas estándar.
 - Económico y alta redundancia y tolerancia a fallos.
 - Problema:
 - Las BBDD relacionales no están diseñadas para trabajar en clústeres.

INTRODUCCIÓN: La caída del imperio relacional

- El ataque de los clústeres.
 - Problema:
 - Las BBDD relacionales no están diseñadas para trabajar en clústeres.
 - Solución:
 - Escalado vertical. ¡Caro!
 - Oracle RAC y Microsoft SQL server. ¡Caro!
 - Virtualización de una máquina en un clúster.
 - » **Problema:** La caída de una máquina puede tirar el servicio.
 - Sharding: Datos distribuidos en diversas máquinas con su correspondiente BBDD.
 - » **Problema:** No se asegura ACID.
 - NoSQL

INTRODUCCIÓN: NoSQL

- NoSQL → Not only SQL → NoREL
- Bases de datos NO RELACIONALES
 - Desaparición de las restricciones intrínsecas de las BBDD relacionales.
 - Modelado de datos por tablas
 - Mismo tipo de datos para todos los elementos de una misma entidad
 - Relaciones entre entidades
 - No se garantiza ACID
 - Atomicidad
 - Consistencia
 - Aislamiento
 - Durabilidad

INTRODUCCIÓN: NoSQL

- No se garantiza la ACID
 - Modelado de datos agregado como alternativa.
 - Facilita el almacenamiento y acceso a datos en clústeres.
 - Solo es necesario una consulta para acceder a datos agregados

```
{_id: <Object1>,
  nombre: "Perico",
  contacto: [{
    telefono: ["123-456-789", "987-654-321"],
    email: "perico@correo.com"
  }],
  direccion: [{
    calle: "Calle principal",
    numero: 2
  }]}
```

Información agregada

INTRODUCCIÓN: Motivación

- Simplicidad
 - Simplificación de las bases de datos.
 - Simplificación del diseño de bases de datos.
 - Simplificación del uso y mantenimiento de bases de datos.
- Diseñado para modelar los datos como van a ser accedidos, no como van a ser almacenados.
- Escalabilidad horizontal.
- Desarrollo ágil

INTRODUCCIÓN: Motivación

- Cambios continuos en los modelos.
- Bases de datos no estructuradas (Schemaless).
- Modelado relacional no resuelve el problema
- Rendimiento:
 - Big data
 - Aplicaciones web en tiempo real
- Diseñado para los paradigmas de programación actuales.

INTRODUCCIÓN: The polyglot persistence

- Entonces...

¡Muerte a las BBDD relacionales!

- No tan rápido:
 - Las BBDD relacionales no van a desaparecer. Aún son útiles en una gran variedad de problemas.
- Nuevo punto de vista:
 - No uses las BBDD relacionales para todo.
 - No tires las BBDD relacionales a la basura.
 - Busca las BBDD óptima para cada problema.
 - Combina las BBDD óptimas en un sistema.

CLASIFICACIÓN

- Documentos
- Clave-Valor
- Columnas
- Grafos
- Orientado a objetos
- Multimodal
- Grid & Cloud
- XML
- Multi-dimensionales
- Multi-valor
- Eventos
- Red


CLASIFICACIÓN: Orientado a documentos

- Principal vertiente de BBDD NoSQL.
- Documento: Conjunto de datos pertenecientes a una entidad.
- Modelado de datos similar a json.
- Modelado de datos más accesible para los humanos frente a las BBDD relacionales.
 - Posibilidad de agrupar/anidar datos dentro de un mismo documento (**Modelo agregado**)
- Acceso a documentos por claves.
 - Identificador.
 - Variables indexadas.

CLASIFICACIÓN: Orientado a documentos

- Ejemplos: Mongo DB, Couch DB, RethinkDB, MarkLogic, etc.

```
{_id: <Object1>,
 nombre: "Perico",
 contacto: [{
   telefono: ["123-456-789", "987-654-321"],
   email: "perico@correo.com"
 }],
 direccion: [{
   calle: "Calle principal",
   numero: 2
 }]
}
```



Variables indexadas

CLASIFICACIÓN: Orientado a clave-valor

- Uso de arrays asociativos:
 - Clave → Valor
- Solo existe un valor para una misma clave.
- Tipo de implementación: hash maps, arboles binarios, etc.
- El valor puede contener todo tipo de datos (**Modelo agregado**)
 - Variables
 - Listas
 - Sets
 - Arrays asociativos

CLASIFICACIÓN: Orientado a clave-valor

- Ejemplos: Redis, MemCached, Riak, etc

clave	valor	expira
1	5	nunca
'variable'	'cadena'	2 horas
'lista'	[1,3,2,3]	mañana
'set'	(1,2,3)	nunca
'hash'	{'perro': 'guau', 'gato': 'miau'}	nunca

CLASIFICACIÓN: Orientado a columnas

- Cierta similitud frente a las BBDD relacionales tradicionales.
- Tupla:
 - Nombre de la columna (único).
 - Dato
 - Timestamp para verificar la validez del dato
- A diferencia de las BBDD relacionales los set de datos (filas) no tienen porque tener las mismas columnas:
 - Una columna puede existir para una fila, y en otra no.
 - Las columnas pueden variar a lo largo del tiempo de vida del set. Puede añadirse o quitarse columnas.
 - Lo datos se pueden **agregar**.

CLASIFICACIÓN: Orientado a columnas

- Ejemplos: Cassandra, Hbase, etc.

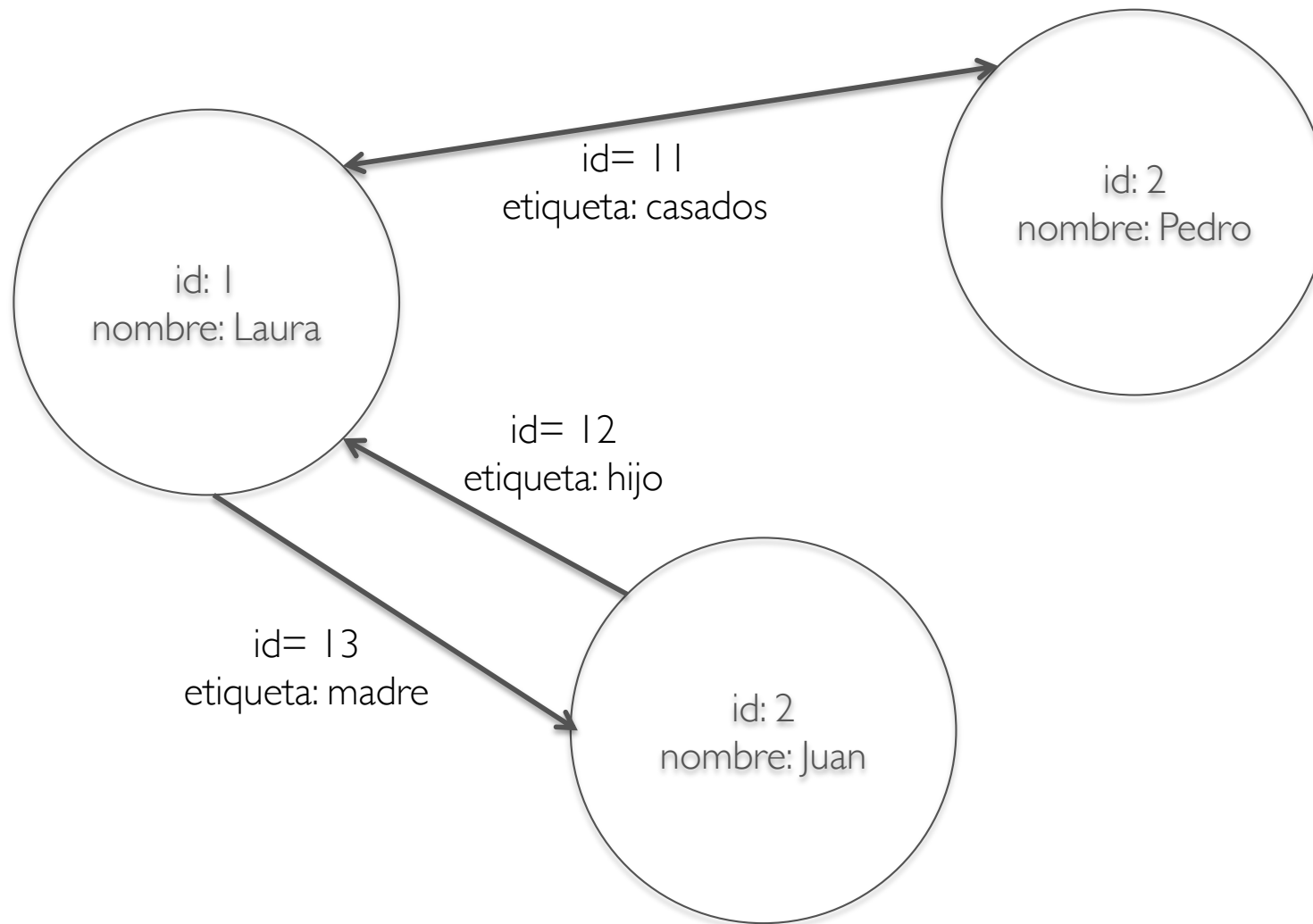
```
{
  nombre: {name: "nombre", value: "Perico", timestamp: 1},
  telefono: {name: "telefono", value: "123", timestamp: 1},
  email: {name: "email", value: "@", timestamp: 1}
},
{
  nombre: {name: "nombre", value: "Pepito", timestamp: 1},
  email: {name: "email", value: "a@b.com", timestamp: 1}
}
{
  nombre: {name: "nombre", value: "Juanito", timestamp: 1},
  trabajo: {name: "trabajo", value: "Paro", timestamp: 1}
}
```


CLASIFICACIÓN: Orientado a grafos

- Objetivo, modelar relaciones.
- Basada en teoría de grafos
 - Uso de nodos, aristas y propiedades para modelar la información
 - Nodos: representa entidades.
 - Aristas: representa relaciones.
 - Propiedades: representa tipo de relación.
- No existen búsquedas por índices. La información está interconectada.
- Muy eficiente para conjuntos de datos asociativos.

CLASIFICACIÓN: Orientado a grafos

- Ejemplos: Neo4j, Titan, etc.



CLASIFICACIÓN: Orientado a objetos

- Información almacenada en forma de objetos.
- Aproximación a bases de datos para programación orientada a objetos.
- Adecuado para modelar datos de alta complejidad.
- Relaciones representadas mediante punteros entre objetos.
- Ejemplos: Cache, DB4o, etc.

CLASIFICACIÓN: Multimodales

- Se combinan técnicas pertenecientes a varios de los tipos de modelado de datos vistos anteriormente.
- Ejemplos:
 - Documentos y clave-valor:
 - Amazon DynamoDB
 - Couchbase (CouchDB + Memcache)
 - Documentos y grafos:
 - OrientDB

Bibliografía

- Date, C. J. (2004). *An Introduction to Database Systems*. Pearson Education India.
- Sadalage, P. J., & Fowler, M. (2013). *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.
- Codd, E. F. (1990). *The relational model for database management: version 2*. Addison-Wesley Longman Publishing Co., Inc..
- Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL database. In *2011 6th international conference on pervasive computing and applications* (pp. 363-366). IEEE.
- Cattell, R. (2011). Scalable SQL and NoSQL data stores. *Acm Sigmod Record*, 39(4), 12-27.
- Moniruzzaman, A. B. M., & Hossain, S. A. (2013). Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*.
- Leavitt, N. (2010). Will NoSQL databases live up to their promise?. *Computer*, 43(2), 12-14.
- No-SQL databases. <http://nosql-database.org/>
- Databases ranking. <https://db-engines.com/en/ranking>