NoSQL

Bases de Datos Avanzadas

Alejandro Osornio

2023-11-21

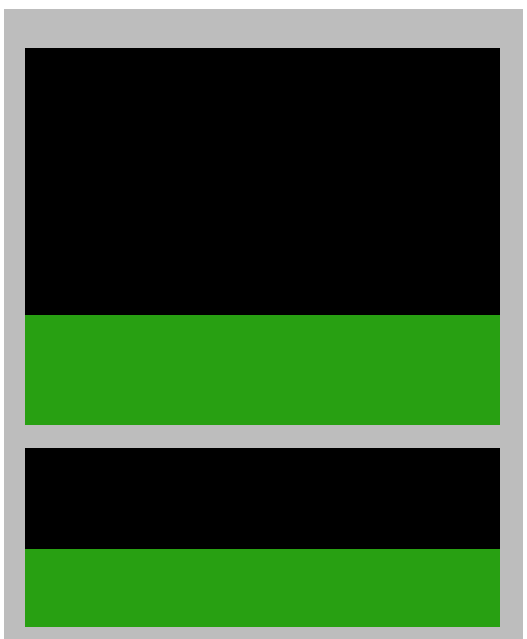
Escalabilidad

- La capacidad de un sistema para manejar una carga cada vez mayor de datos y trabajo.
- Ejemplo: Tienes un servidor, cuyo servicio ocupa el 80% ocupado del disco duro y 85% de RAM.

Quiz!!

¿Cómo harían para tener más recursos para atender a los usuarios?

Horizontal vs Vertical



Quiz!!

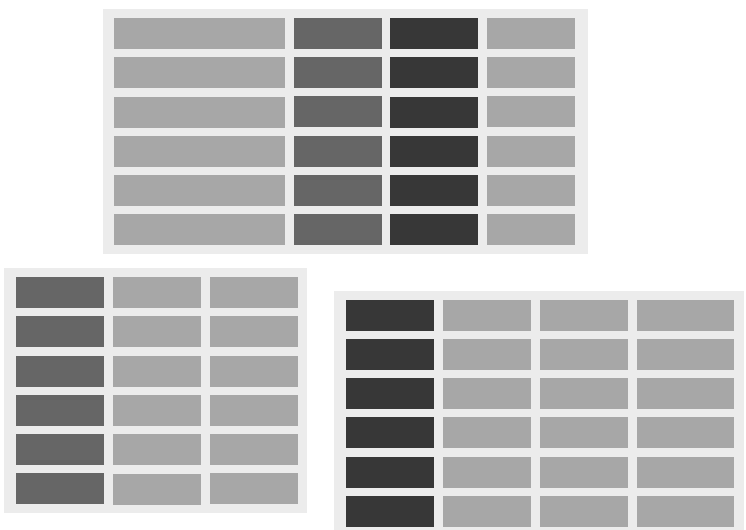
10 millones de usuarios, ¿escalar horizontal o vertical?

Modelo de datos

- Describe cómo se estructuran los datos y cómo se accede a ellos.
- El modelo de datos tiene implicaciones sobre las operaciones y representación de los datos
- Un modelo implica formas distintas:
  - De relaciones
  - De entidades
  - De atributos

Modelo relacional

- Propuesto por Edgar F. “Ted” Codd en 1970, de la IBM
- Basado en tablas, formadas por columnas (atributos) y filas (entidades)
- Se identifica las tuplas con llaves, y a la vez se usan para formar relaciones
- Noten que el modelo esta orientado a entidades, las relaciones son secundarias
- Las tablas deben ser pre definidas



"SQL"

- Desarrollado por Donald D. Chamberlin y Raymond F. Boyce, de la IBM
- Structured Query Language, antes Structured English Query Language (Sequel)
- Lenguaje utilizado para el manejo de información en RDBMS.
- Emplea algebra relacional y cálculo relacional de tuplas, muy innovador y sencillo
- Utilizado con datos estructurados

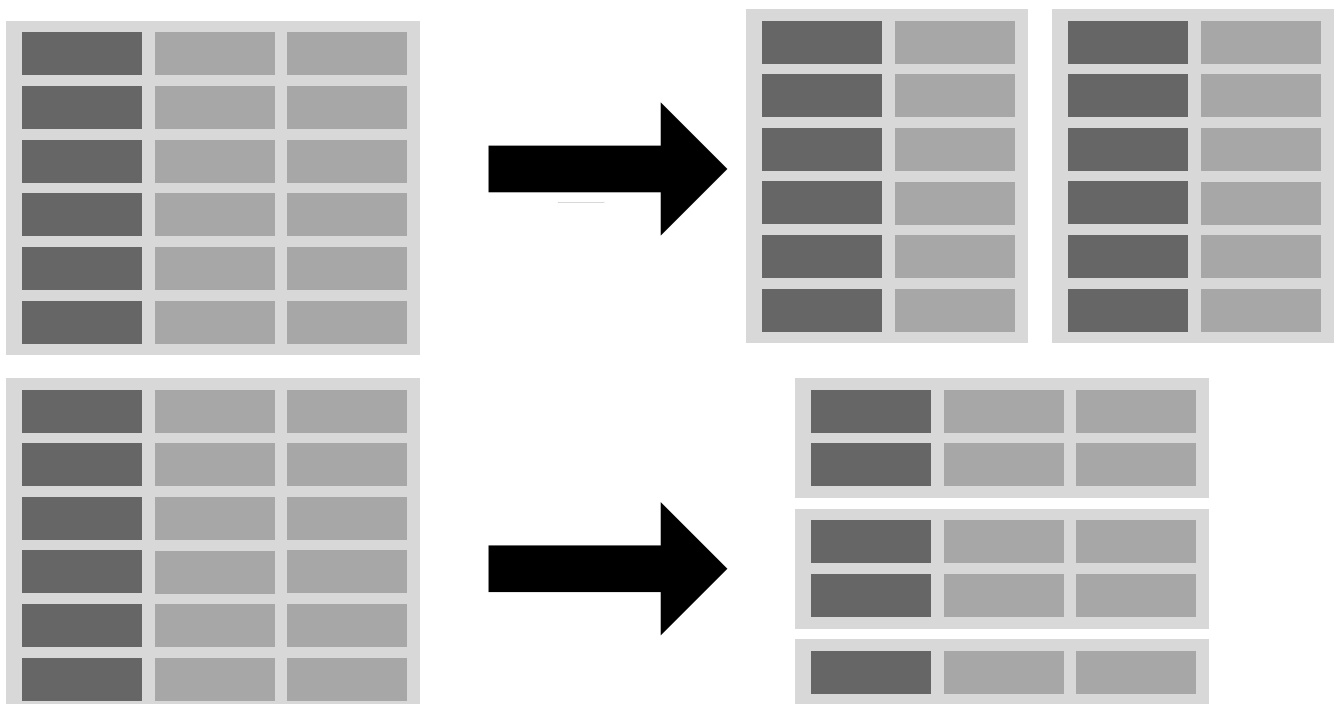
Quiz!!

¿Qué es una función hash?

RDBMS

- Sistema gestor de bases de datos relacionales
- Usa SQL como lenguaje para manejar la información
- Utiliza el modelo relacional
- Fuerte enfoque en ACID (Accesibilidad, Consistencia, *Isolation*, Durabilidad), que verifica para cada transacción.

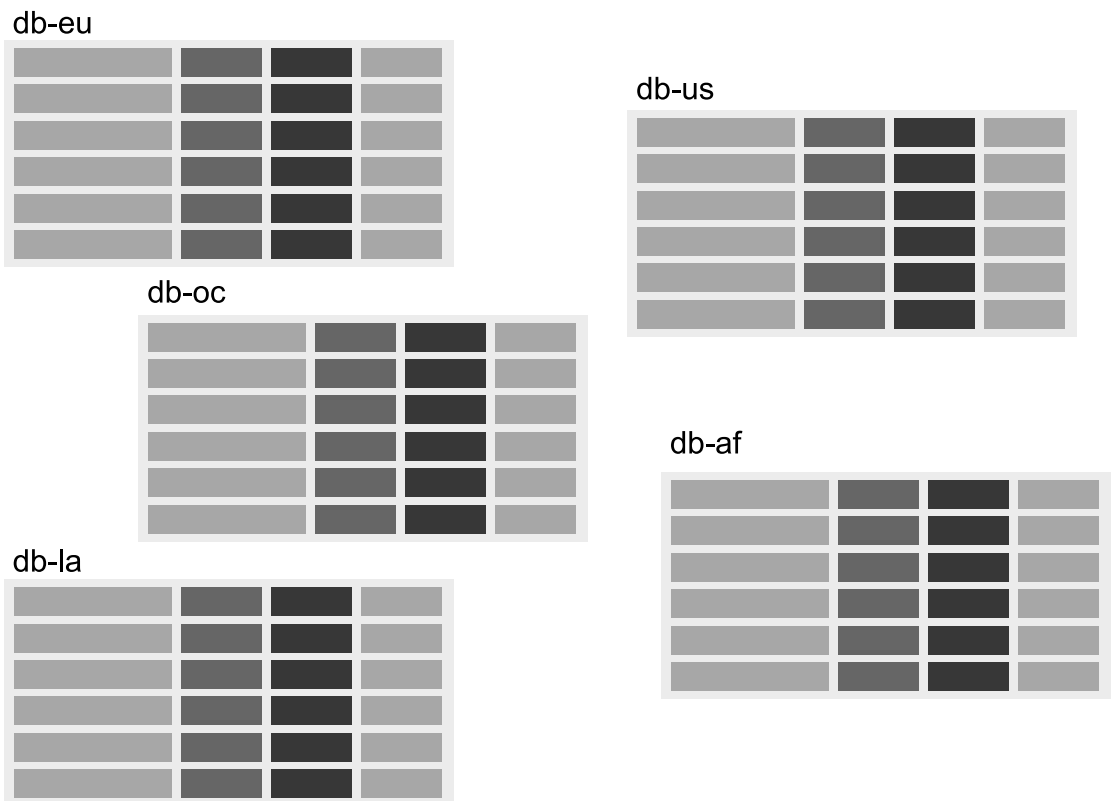
Sharding



Quiz!!

10 millones de usuarios, ¿cómo escalamos la base de datos?

Sharding



CAP

Los sistemas pueden cubrir solo 2 características de entre:

- Consistency: cómo se comporta el estado del sistema cuando se realiza una operación
- Availability: Se puede continuar el servicio aunque existan fallas en hardware u otros nodos
- Partition Tolerance: Cómo se desenvuelve el sistema cuando hay **islas** que no pueden conectarse entre si. Por ej, cuando agregamos/quitamos nodos

Quiz!!

¿Qué es lo que sacrifican los RDBMS de CAP?

NoSQL

- En general, todo lo que no es *SQL*. Es decir, que no sigue el modelo de datos relacional.
- Implica que no se use el mismo ANSI/ISO SQL
- Si damos menor prioridad a las retricciones que pone ACID se puede lograr escalabilidad y rendimiento.
- Menos consistencia e isolación para lograr disponibilidad y rendimiento.

NoSQL

Permite:

- Manejo de datos estructurados y no estructurados
- Facilita el desarrollo iterativo
- Mejora la experiencia del desarrollador
- Proporciona más herramientas para poder escalar mejor
- Uso inteligente de índices, hashes, y cache

Key-Value

- La información se representa de la forma (K, V)
- Si queremos un valor, simplemente lo buscamos usando su llave
- La llave K es un nombre de archivo, texto plano, hash, URI, etc.
- El valor V es un *blob* binario, puede almacenar lo que sea
- El soporte de operaciones basado en el valor es limitado (p. ej buscar los usuarios que se llamen Daniel)
- Multiples tipos: en memoria, en disco, híbrido
- `Get(key)`

Quiz!!

¿Cómo será que se particiona una base de datos *Key-Value*?

Wide-Column Stores

- Familia de columnas
- Similar a *Key-Value*, se ve de la forma $(K, (C, C, C, C, ...))$
- Donde cada columna se ve de la forma (K, V), es decir que se ve $(K, ((K_1, V_1), (K_2, V_2), ...))$
- Suelen dar soporte para almacenar versiones de columna con una tercera llave de tiempo `Get(key, col, time)`
- Algunos (p. ej Cassandra) permiten tener *agregados* (columnas anidadas en columnas)
- Partición eficiente en vertical y horizontal
- `Get(key, col)`

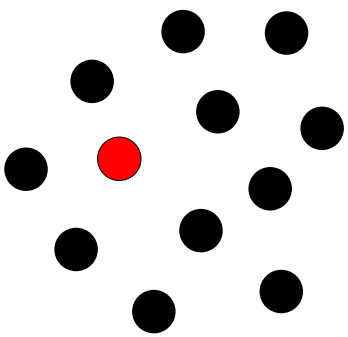
Document

- Similar a *Key-Value* de la forma (K, V) donde V es JSON, XML, o BSON.
- Permite cambios en la estructura de la información
- Se puede agregar soporte para más tipos de valores.
- Algunos (p. ej. MongoDB) permiten tener colecciones de documentos
- Se puede obtener atributos especificos del documento sin cargarlo todo
- Toda la información está contenida en el mismo documento
- Se puede relacionar otros documentos usando su id

```
{
  "_id": "123456789",
  "title": "Introduction to NoSQL Databases",
  "author": {
    "name": "John Doe",
    "email": "john.doe@example.com"
  },
  "content": "NoSQL databases provide a flexible and scalable approach to data storage, allowing developers
to...",
  "tags": ["NoSQL"],
  "date_published": "2023-01-15T08:00:00Z",
  "comments": [
    {
      "user": "Alice",
      "comment_text": "Great article! I learned a lot."
    },
    {
      "user": "Bob",
      "comment_text": "Could you elaborate on the sharding techniques mentioned?"
    }
  ]
}
```

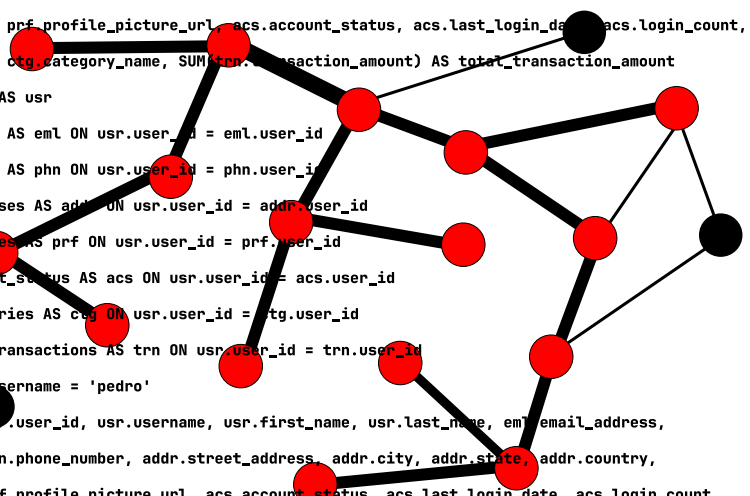
Quiz!!

¿En dónde usarían una base de datos con este modelo?



SELECT user:pedro;

```
SELECT usr.user_id, usr.username, usr.first_name, usr.last_name, eml.email_address,
           phn.phone_number, addr.street_address, addr.city, addr.state, addr.country,
           prf.profile_picture_url, acs.account_status, acs.last_login_date, acs.login_count,
           ctg.category_name, SUM(trn.transaction_amount) AS total_transaction_amount
FROM users AS usr
JOIN emails AS eml ON usr.user_id = eml.user_id
JOIN phones AS phn ON usr.user_id = phn.user_id
JOIN addresses AS addr ON usr.user_id = addr.user_id
JOIN profiles AS prf ON usr.user_id = prf.user_id
JOIN account_status AS acs ON usr.user_id = acs.user_id
JOIN categories AS ctg ON usr.user_id = ctg.user_id
LEFT JOIN transactions AS trn ON usr.user_id = trn.user_id
WHERE usr.username = 'pedro'
GROUP BY usr.user_id, usr.username, usr.first_name, usr.last_name, eml.email_address,
         phn.phone_number, addr.street_address, addr.city, addr.state, addr.country,
         prf.profile_picture_url, acs.account_status, acs.last_login_date, acs.login_count,
         ctg.category_name
HAVING total_transaction_amount > 1000
```

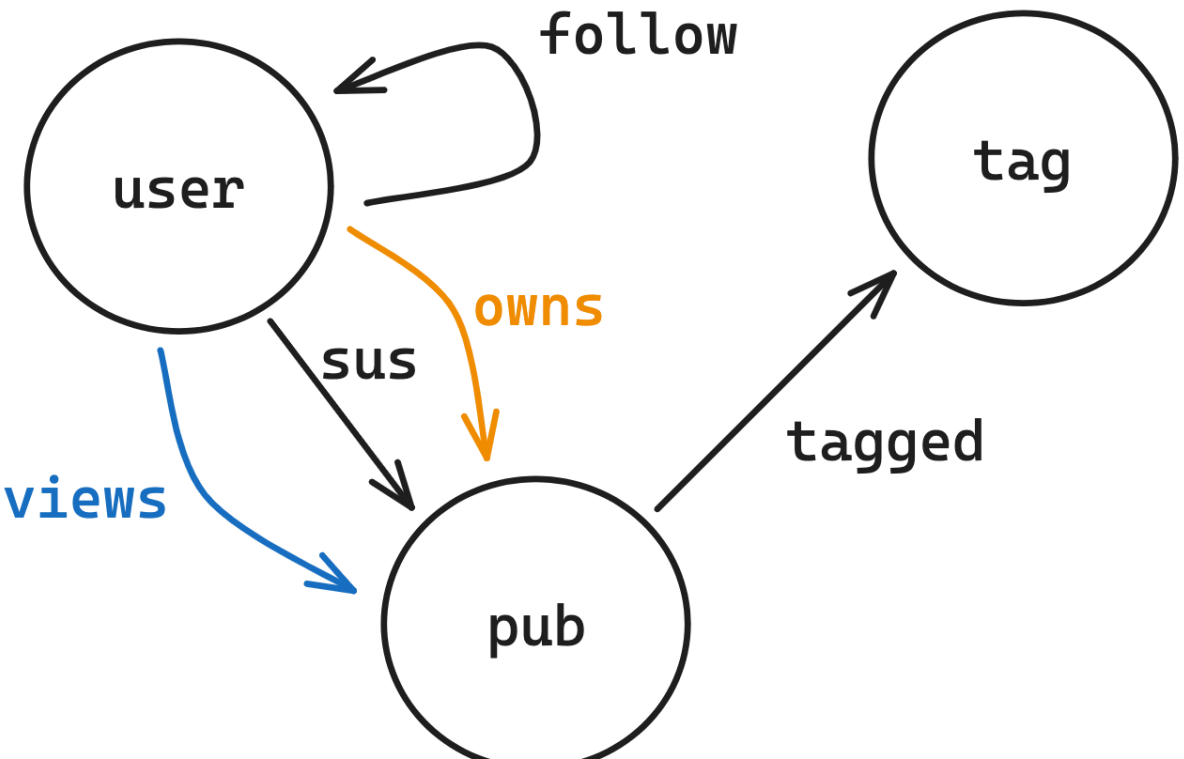


Graph

- Utilizan teoría de grafos para almacenar los datos y representar relaciones.
- Consiste en vértices que representan entidades y aristas que representan relaciones entre ellas
- Brindan lenguajes de *consulta* que facilitan el recorrido del grafo
- Permiten hacer consultas complejas sin necesidad de joins
- Por debajo usa los demás modelos para representar el grafo

Ejemplo!!

Vamos a crear un *engine* de recomendaciones, basado en las etiquetas de las publicaciones que hemos visto nosotros y lo usuarios que seguimos


follow
user
tag
owns
sus
views
tagged
pub

```
BEGIN;
    LET $seen = SELECT VALUE id
      FROM (SELECT out.id as id
          FROM ($auth.id)->view GROUP BY id LIMIT 100 );
    LET $seen_by_others = SELECT VALUE id
      FROM (SELECT out.id as id
          FROM ($auth.id)->follow->user->view GROUP BY id LIMIT 100);
    LET $seeng = array::union($seen, $seen_by_others);
    LET $a = SELECT * FROM $seeng->tagged->tag WHERE array::len(id) > 0;
    LET $tags = RETURN array::distinct(array::flatten($a));

    RETURN SELECT *, fn::is_sus(id)
      FROM array::distinct(array::flatten((SELECT VALUE in.* FROM $tags<-tagged)));
COMMIT;
```

Publicaciones vistas

```
LET $seen = SELECT VALUE id FROM (
  SELECT out.id as id FROM ($auth.id)->view
  GROUP BY id LIMIT 100
);

LET $seen_by_others = SELECT VALUE id FROM (
  SELECT out.id as id FROM ($auth.id)->follow->user->view
  GROUP BY id LIMIT 100
);

LET $seeng = array::union($seen, $seen_by_others);
```

Etiquetas de pub

```
LET $tags_por_pub = SELECT * FROM $seeng->tagged->tag WHERE
array::len(id) > 0;

LET $tags = RETURN
array::distinct(array::flatten($tags_por_pub));

RETURN SELECT *, fn::is_sus(id)
  FROM array::distinct(array::flatten((
          SELECT VALUE in.* FROM $tags<-tagged
  )));
```

Ventajas

- Pueden ofrecer mejor experiencia al desarrollar
- El modelo de datos es más simple de entender
- Flexibilidad con la información que se almacena
- Facilidad para realizar análisis sobre información profundamente relacionada
- Facilidad para escalar de forma horizontal

Desventajas

- Son menos confiables por ser menos estrictos con ACID en cada transacción
- No hay un lenguaje de manipulación de datos estándard
- Hay que aprender nuevos lenguajes conceptos y herramientas para utilizarlas
- Pocas herramientas para trabajar con ellas, como plataformas para *hostear*

Gracias!

[1] A. Davoudian, L. Chen, and M. Liu, “A Survey on NoSQL Stores,” ACM Computing Surveys, vol. 51, no. 2. Association for Computing Machinery (ACM), pp. 1-43, Apr. 17, 2018. doi: 10.1145/3158661.

[2] M. Besta et al., “Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries.” arXiv, 2019. doi: 10.48550/ARXIV.1910.09017.

[3] Jing Han, Haihong E, Guan Le, and Jian Du, “Survey on NoSQL database,” 2011 6th International Conference on Pervasive Computing and Applications. IEEE, Oct. 2011. doi: 10.1109/icpca.2011.6106531.

[4] R. Angles and C. Gutierrez, “Survey of graph database models,” ACM Computing Surveys, vol. 40, no. 1. Association for Computing Machinery (ACM), pp. 1-39, Feb. 2008. doi: 10.1145/1322432.1322433.

[5] ComputerWorld. “The Story So Far”. Consultado en Nov 2023 Disponible en https://www.computerworld.com/article/2588199/the-story-so-far.html

[6] Santhosh Kumar Gajendran. 2012. “A Survey on NoSQL Databases”. (2012).