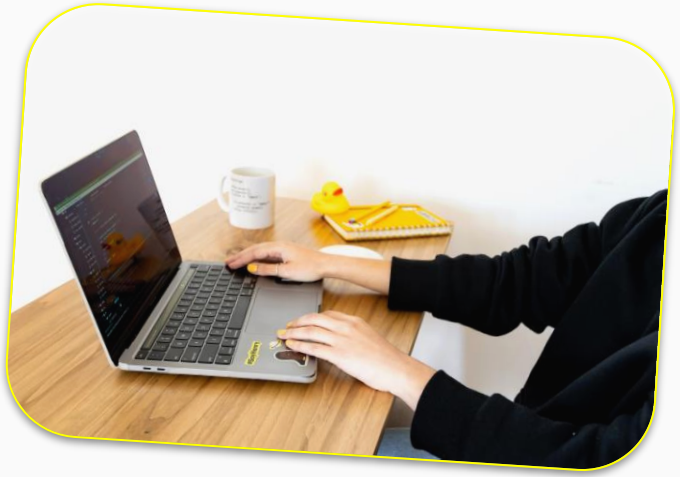
HENRY

SQL
NoSQL

OBJETIVOS DE CLASE

- **Diferenciar** las principales características de los motores de bases de datos SQL y NoSQL
- **Conocer** las características ACID y BASE
- **Comprender** el Teorema CAP y los conceptos de Tolerancia a Particiones, Consistencia y Disponibilidad.



AGENDA

- SQL
- NoSQL
- ACID & BASE
- Teorema CAP
- Tolerancia a Particiones
- Consistencia
- Disponibilidad

Motores de Bases de datos

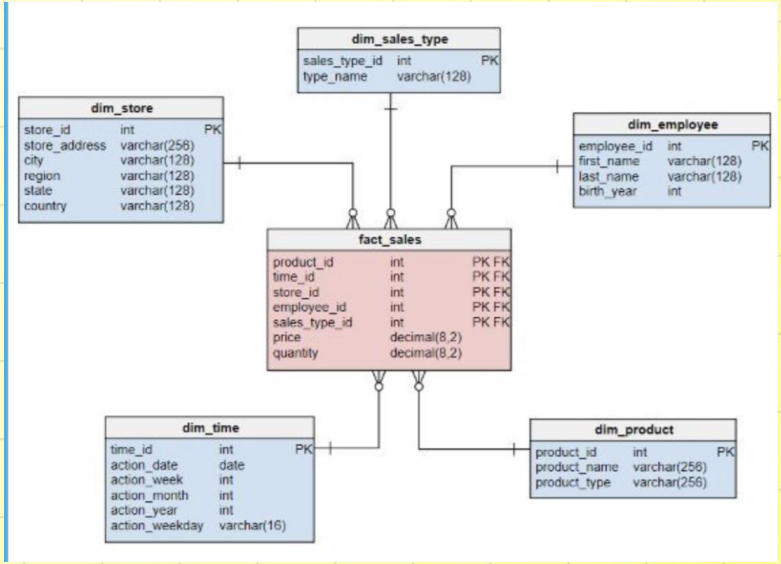
SQL

¿Qué son?

Las bases de **datos relacionales**, RDBMS son universalmente utilizadas prácticamente por todas las empresas, están estructuradas y optimizadas para asegurar la consistencia y exactitud de los datos, mientras también eliminan Redundancia. Estas bases se **almacenan en grandes y confiables servidores** para asegurar que los datos estén siempre disponibles.

Características

- Formato tabular
- Tipos de datos
- Relaciones (JOINS)
- Restricciones (PK, FK)



NoSQL

¿Qué son?

Una base de datos **NoSQL** (“**Not Only SQL**”) es una manera efectiva de organizar gran cantidad de datos heterogéneos con acceso y actualizaciones de datos eficientes. Esto se logra flexibilizando algunas restricciones de bases de datos relacionales. Son muy útiles para **analizar y utilizar masivas cantidades de datos estructurados o no estructurados** almacenados en **servidores remotos.**

Características

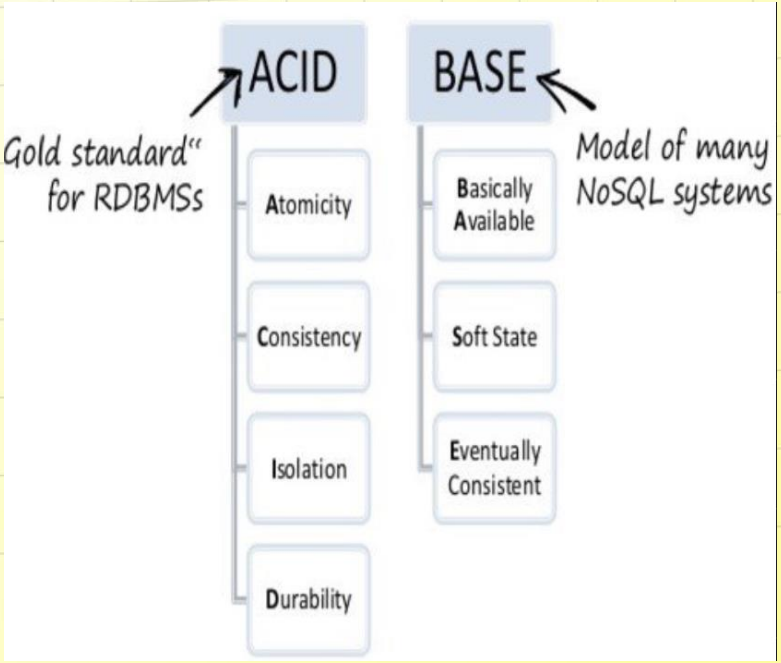
- Casos de uso específicos
- Esquema flexible
- Escalabilidad horizontal
- No utilizan SQL ni JOINS

ACID & BASE

ACID

NoSQL no soporta el esquema relacional y las propiedades asociadas con los procesos transaccionales propios de RDBMS, ACID (“Atomicity, Consistency, Isolation, Durability”).

En cambio soporta propiedades **BASE** (“Basically Available, Soft State, Eventually Consistent”). Las **bases NoSQL** son aproximadamente precisas en cualquier punto del tiempo y eventualmente consistentes.



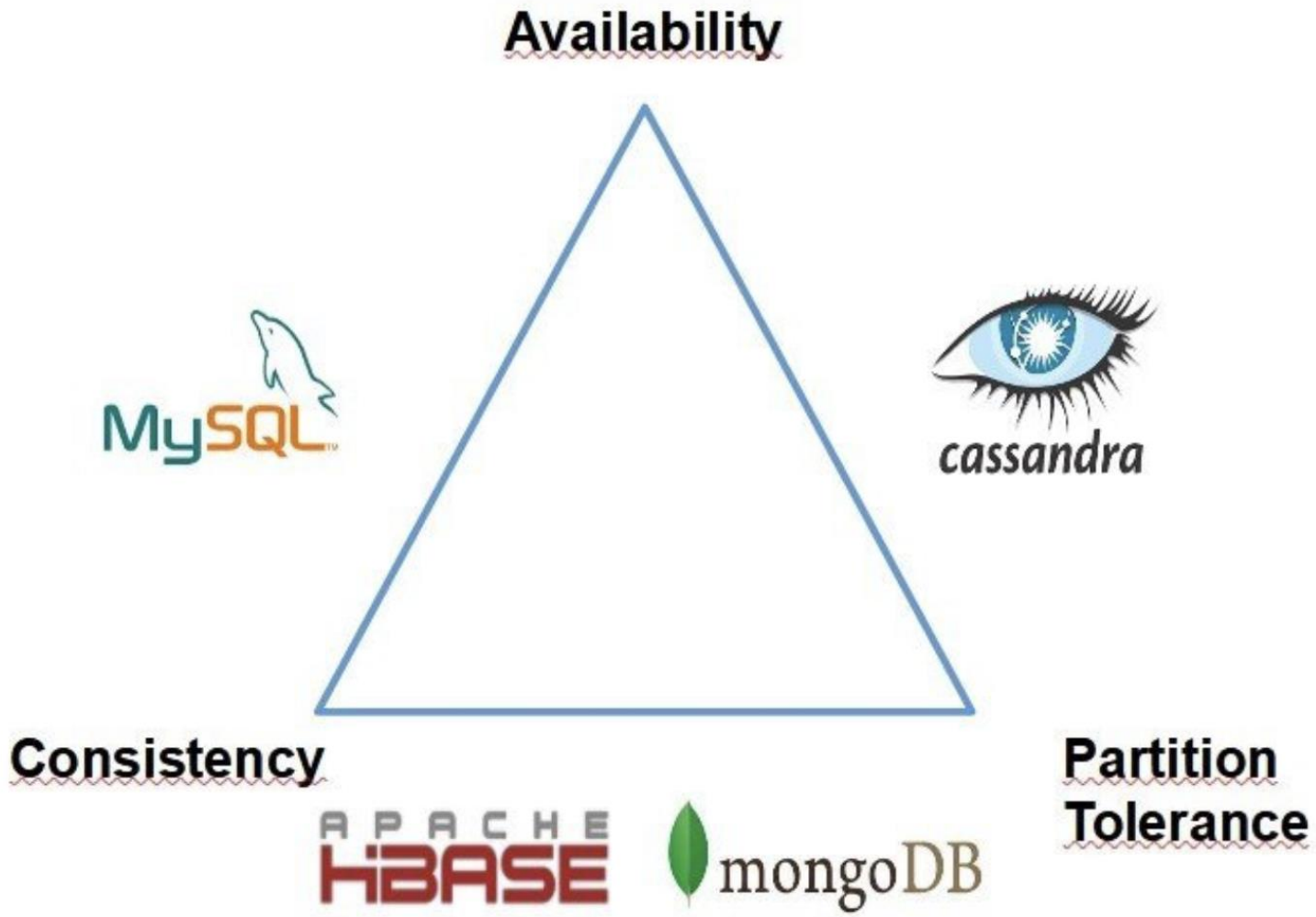
- **Atomicidad**: la ejecución de cada transacción es atómica, se realizan todas las acciones o ninguna.
- **Consistencia**: Cada transacción preserva la integridad.
- **Aislamiento**: Una transacción no puede afectar a la otra.
- **Durabilidad**: Persistencia de los cambios luego de un “COMMIT”.

BASE

- **Básicamente disponible:** garantiza la disponibilidad de los datos, responde a cada solicitud.
- **Soft State:** El estado del sistema puede cambiar en el tiempo, (consistencia eventual).
- **Consistencia eventual:** el sistema “**eventualmente**” es consistente una vez que termina de recibir datos, los datos se propagan y el sistema no chequea la consistencia de cada transacción antes de mover a la siguiente.

Teorema CAP

- Un sistema es considerado **Consistente** (“Consistency”) si todas las réplicas contienen el mismo valor.
- Un sistema es considerado **Disponible** (“Availability”) si los datos están disponibles en todo momento.
- Un sistema tiene **tolerancia a Particiones** (“Partition Tolerance”) si el procesamiento puede continuar en ambas particiones en dos o más “islas”.
- Es importante también que la **data sea consistente y esté disponible** aún ante un fallo en la red que deje “particionada” la base en “islas”.
- De acuerdo con el **Teorema CAP** no es posible proveer simultáneamente más de dos de las tres propiedades (Consistencia, Disponibilidad, Particionado).

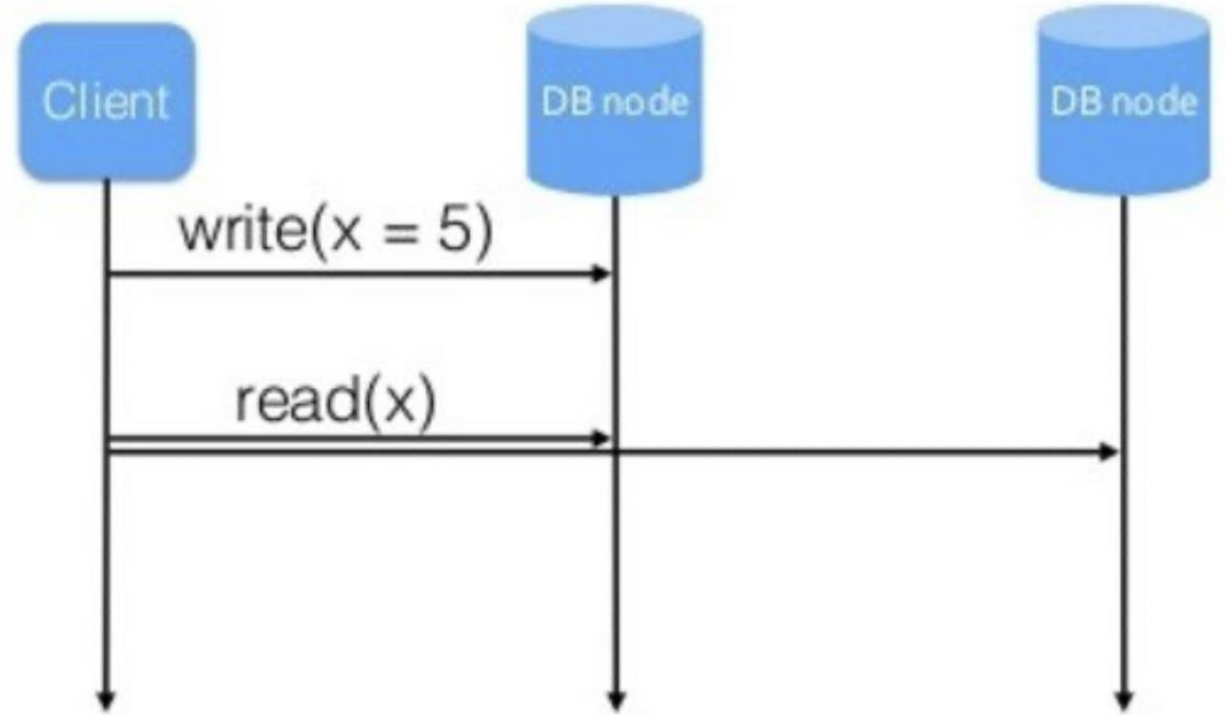

Availability
MySQL
cassandra
Consistency
Partition
Tolerance
APACHE
HBASE
mongoDB

Tolerancia a Particiones

¿Qué es?

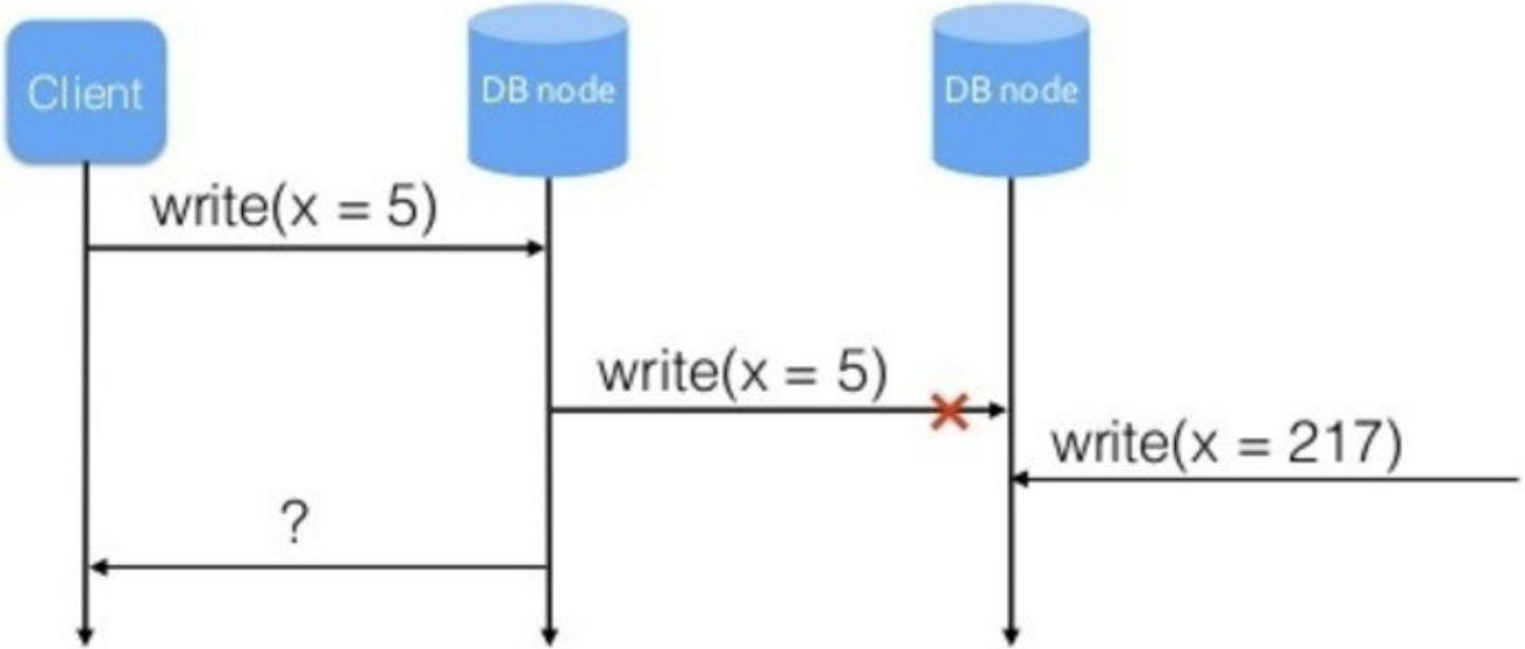
Con frecuencia, CAP se malinterpreta como si uno tuviera que elegir abandonar una de las tres garantías en todo momento. De hecho, la elección entre la consistencia y la disponibilidad se da solo cuando ocurre una partición de red o falla; en cualquier otro momento, no hay que hacer concesiones. Los sistemas de base de datos diseñados teniendo en cuenta las garantías tradicionales de ACID, como RDBMS, eligen la consistencia sobre la disponibilidad; mientras que los sistemas diseñados en torno a la filosofía BASE, comunes en el marco NoSQL, elige la disponibilidad sobre la consistencia.

consistencia



Consistency = these two reads are guaranteed to return the same value (which need not be 5)

Disponibilidad



If the database accepts
the write, you have availability
(the other node may be accepting
other writes, even to x)

RESUMEN DE LA CLASE

- ✓ SQL
- ✓ NoSQL
- ✓ Acid & Base
- ✓ Tolerancia a particiones

¿PREGUNTAS?

¡Muchas gracias!