ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

Introduccion BBDD NoSQL

Alvaro Bravo

Índice

1. Historia sobre BBDD Relacionales

La base de datos relacional fue inventada por E.F. Codd en IBM en 1970.

Una base de datos relacional es una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas desde la que se puede acceder a los datos o volver a montarlos de muchas maneras diferentes sin tener que reorganizar las tablas de la base.

La interfaz estándar de programa de usuario y aplicación a una base de datos relacional es el lenguaje de consultas estructuradas (**SQL**). Los comandos de SQL se utilizan tanto para consultas interactivas para obtener información de una base de datos relacional y para la recopilación de datos para los informes.

- Una **base de datos** se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de campos (columnas) y registros (filas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y claves foráneas (o ajenas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y estas deben cumplir con la **integridad de datos**.
- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de estas se hacen las formas relacionales.

Ventajas

- Provee herramientas que garantizan evitar la duplicidad de registros.
- Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.

La base de datos se organiza en dos marcadas secciones; el esquema y los datos (o instancia).

El esquema es la definición de la estructura de la base de datos y principalmente almacena los siguientes datos:

- El nombre de cada **tabla**
- El nombre de cada **columna**
- El **tipo de dato** de cada columna
- La tabla a la que pertenece cada columna

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización de una base de datos, el resultado de dicho proceso es un esquema que permite que la base de datos sea usada de manera óptima.

Los datos o instancia es el contenido de la base de datos en un momento dado. Es en sí, el contenido de todos los registros.

2. Modelos de datos

CUBOS OLTP

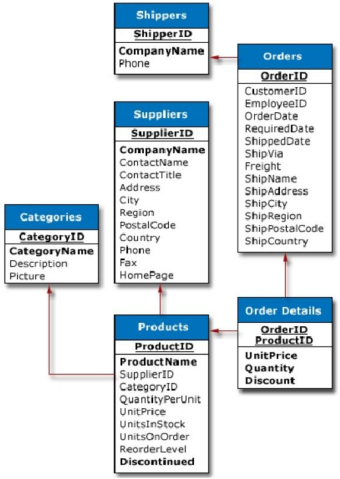
- Almacena datos actuales
- Almacena datos de detalle
- Los datos son dinámicos
- Las transacciones son repetitivas
- El número de transacciones es elevado
- Dedicado al procesamiento de transacciones
- Orientado a los procesos de la organización
- Soporta decisiones diarias
- Sirve a muchos usuarios administrativos

CUBOS OLAP

- Almacena datos históricos
- Almacena datos de detalle y datos agregados a distintos niveles
- Los datos son estáticos
- Los procesos no son previsibles
- El número de transacciones es bajo o medio
- Dedicado al análisis de datos
- Orientado a la información relevante
- Soporta decisiones estratégicas
- Sirve a técnicos de dirección

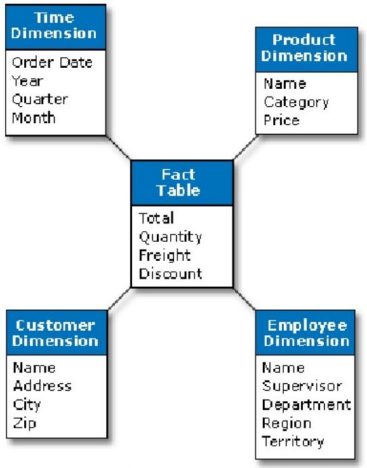
OLTP

- Ejemplos de OLTPs pueden incluir ERP, CRM, SCM, aplicaciones de Punto de Venta, Call Center.
- Están diseñados para una velocidad de transacción óptima.
- Los datos no se almacenan por un período prolongado en OLTPs, por costos de almacenamiento y las razones de velocidad de transacciones.



OLAP
Modelo Dimensional de Datos

- OLAPs están diseñados para ofrecer un análisis general de lo que sucedió.
- Por lo tanto el almacenamiento de datos (es decir, el modelado de datos) tiene que establecerse de manera diferente.
- El método más común es el diseño de la estrella.
- Son distintos a la tercera forma normal, utilizados en sistemas OLTP.



3. BBDD Relacionales VS NoSQL

NoSQL abarca una amplia variedad de tecnologías de bases de datos diferentes que se desarrollaron en respuesta a las demandas presentadas en la construcción de aplicaciones modernas:

Los desarrolladores están trabajando con aplicaciones que crean volúmenes masivos de nuevos tipos de datos que cambian rápidamente: **datos estructurados, semiestructurados, no estructurados y polimórficos**.

Se acabó el ciclo de desarrollo de la cascada de doce a dieciocho meses. Ahora los equipos pequeños trabajan en sprints ágiles, iterando rápidamente y presionando el código cada semana o dos, algunas incluso varias veces al día.

Las aplicaciones que una vez sirvieron a una audiencia limitada ahora se entregan como servicios que deben estar siempre encendidos, accesibles desde muchos dispositivos diferentes y escalados globalmente a millones de usuarios.

Las organizaciones ahora están recurriendo a arquitecturas de escalamiento horizontal que utilizan software de código abierto, servidores de productos básicos y computación en la nube en lugar de grandes servidores monolíticos e infraestructura de almacenamiento.

Las bases de datos relacionales no se diseñaron para hacer frente a los desafíos de escala y agilidad que enfrentan las aplicaciones modernas, ni se crearon para aprovechar el almacenamiento de productos básicos y la capacidad de procesamiento disponible en la actualidad.

ACID

- Fuerte Consistencia
- Aislamiento
- Enfocado en Commit
- Transacciones anidadas
- Pesimista
- Evolución Compleja

BASE

- Consistencia Eventual
- Disponibilidad Primero
- Mejor Rendimiento
- Optimista
- Respuestas Aproximadas
- Rapido y Simple
- Evolución Simple

4. Diferentes tipos de BBDD NoSQL

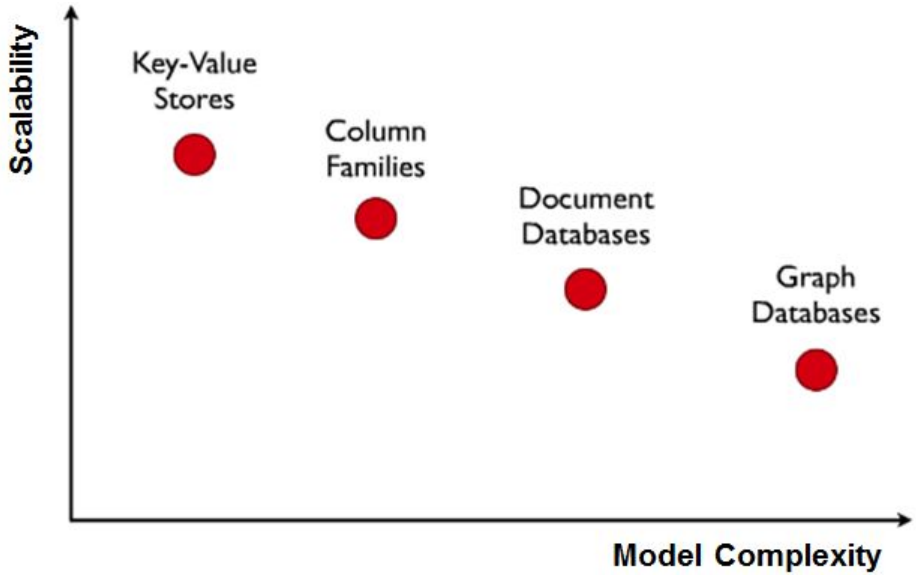
- **Orientadas a documentos**
  - Gestionan datos semi estructurados
  - JSON, XML, BSON
  - Muy versátiles
  - Gestión de contenido y manejo de datos de aplicaciones móviles.
  - **MongoDB, CouchDB**

- **Clave-Valor**
  - Más simples
  - Muy rápidas
  - Búsqueda por clave
  - Administración de sesiones y el almacenamiento en caché en aplicaciones web.
  - **Redis**

- **Column Family**
  - Realizar consultas y agregaciones sobre grandes conjuntos de datos
  - Almacenan en columnas en vez de registros
  - Motores de recomendación, catálogos, detección de fraudes
  - **Cassandra,HBase**

- **Orientadas a grafos**
  - Basada teoría de grafos (nodos, aristas).
  - sistemas que deben asignar relaciones, como sistemas de reserva o gestión de relaciones con clientes.
  - **Neo4j**

Rank Nov 2018	Rank Oct 2018	Rank Nov 2017	DBMS	Database Model	Score Nov 2018	Score Oct 2018	Score Nov 2017
1.	1.	1.	Oracle	Relational DBMS	1301.11	-18.16	-58.94
2.	2.	2.	MySQL	Relational DBMS	1159.89	-18.22	-162.14
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1051.55	-6.78	-163.53
4.	4.	4.	PostgreSQL	Relational DBMS	440.24	+20.85	+60.33
5.	5.	5.	MongoDB	Document store	369.48	+6.30	+39.01
6.	6.	6.	IBM Db2	Relational DBMS	179.87	+0.19	-14.19
7.	7.	↑ 9.	Redis	Key-value store	144.17	-1.12	+22.99
8.	8.	↑ 10.	Elasticsearch	Search engine	143.46	+1.13	+24.05
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	138.44	+1.64	+5.12
10.	↑ 11.	↑ 11.	SQLite	Relational DBMS	122.71	+5.96	+9.95
11.	↓ 10.	↓ 8.	Cassandra	Wide column store	121.74	-1.64	-2.47
12.	↑ 13.	↑ 15.	Splunk	Search engine	80.37	+3.48	+15.50
13.	↓ 12.	↓ 12.	Teradata	Relational DBMS	79.31	+0.67	+1.07
14.	14.	↑ 18.	MariaDB	Relational DBMS	73.25	+0.12	+17.96
15.	↑ 16.	↑ 19.	Hive	Relational DBMS	64.57	+3.47	+11.32
16.	↓ 15.	↓ 13.	Solr	Search engine	60.87	-0.44	-8.28
17.	17.	↓ 16.	HBase	Wide column store	60.41	-0.26	-3.15
18.	18.	↓ 14.	SAP Adaptive Server	Relational DBMS	56.57	-2.00	-10.47
19.	↑ 21.	↑ 20.	SAP HANA	Relational DBMS	55.88	+1.50	+6.70
20.	↓ 19.	↓ 17.	FileMaker	Relational DBMS	55.75	-0.29	-3.09
21.	↓ 20.	↑ 22.	Amazon DynamoDB	Multi-model	53.81	-0.65	+16.69
22.	22.	↓ 21.	Neo4j	Graph DBMS	43.12	+0.47	+4.67
23.	23.	23.	Couchbase	Document store	34.85	-1.06	+2.54



5. Teorema CAP

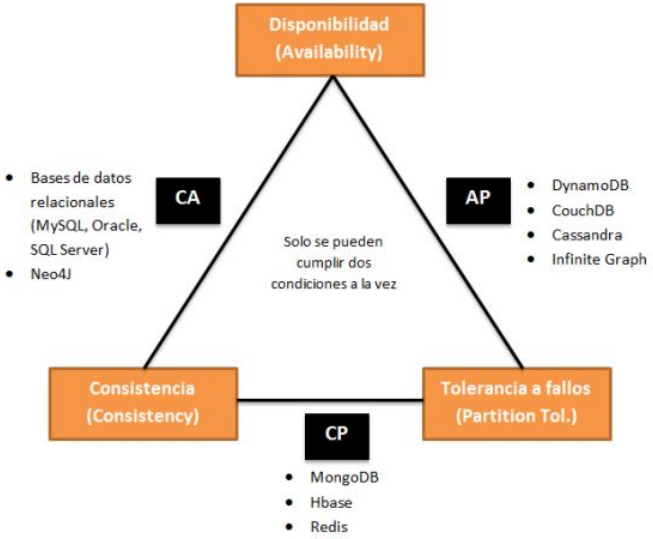
CAP

- **Consistencia**: cualquier lectura recibe como respuesta la escritura más reciente o un error.
- **Disponibilidad**: cualquier petición recibe una respuesta no errónea, pero sin la garantía de que contenga la escritura más reciente.
- **Tolerancia a particiones**: el sistema sigue funcionado incluso si un número arbitrario de mensajes son descartados (o retrasados) entre nodos de la red

Tipos

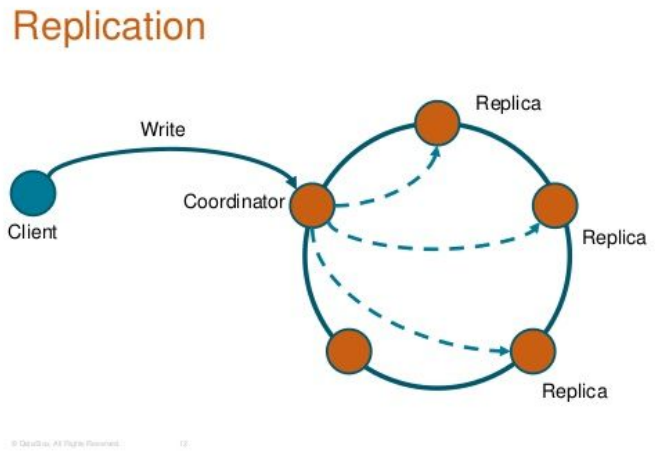
- **AP**: sistema siempre procesará la consulta e intentará devolver la versión disponible más reciente de la información, incluso si no puede garantizar que esté actualizada debido a la partición de la red.
- **CP**: sistema devolverá un error o un tiempo de espera si no se puede garantizar que la información particular esté actualizada debido a la partición de la red.
- **CA**: garantizan consistencia y disponibilidad, pero tienen problemas con la tolerancia a particiones. Este problema lo suelen gestionar replicando los datos.

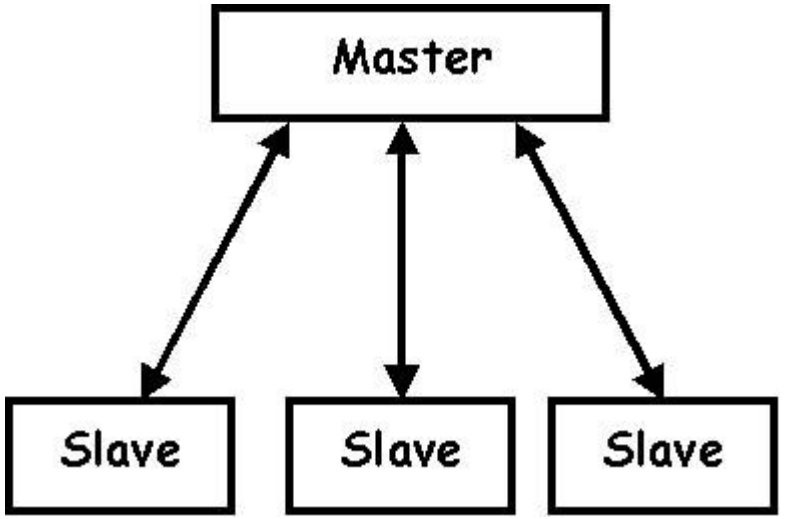
Nota. Cuando el sistema distribuido se está ejecutando normalmente, se puede satisfacer tanto la disponibilidad como la consistencia



MongoDB por defecto es CP pero puede cambiar a AP si permitimos leer de los nodos secundarios

6. MongoDB VS Cassandra





ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID