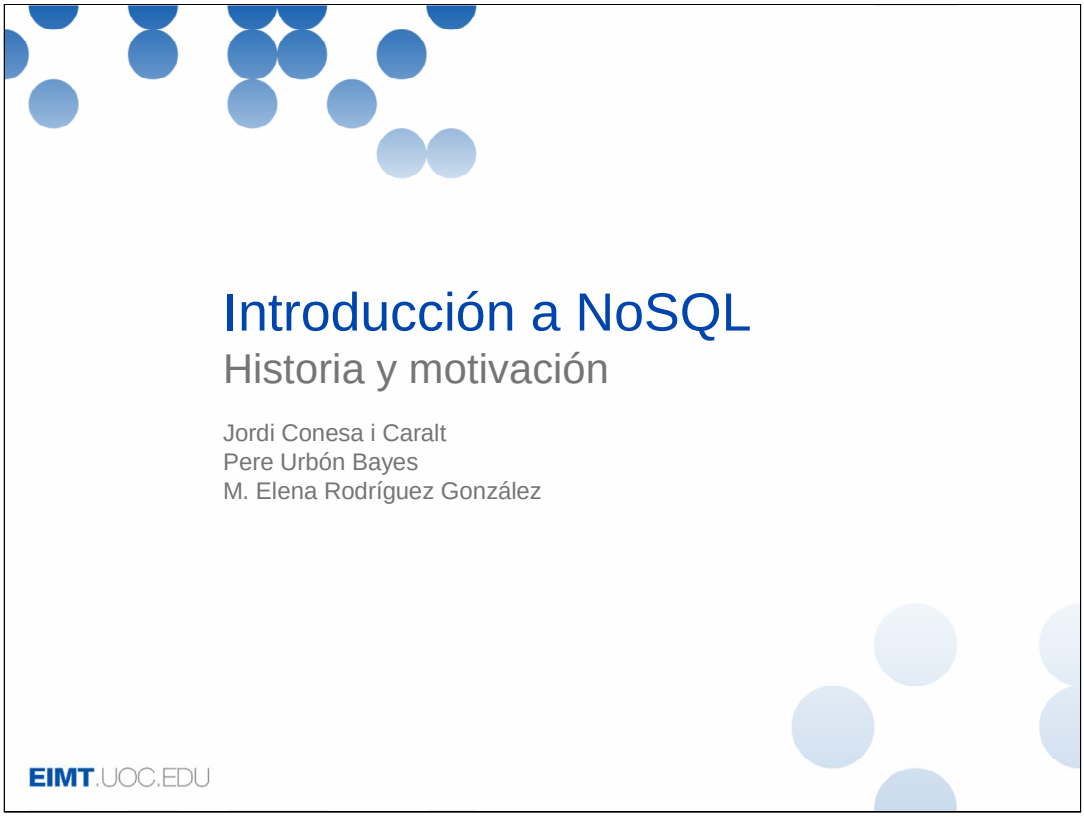


Bienvenidos a la presentación de introducción a NoSQL, donde hablaremos sobre este nuevo tipo de bases de datos tan presentes en la actualidad, haciendo hincapié en su definición, su motivación y su contextualización en el mundo de las bases de datos.

Introducción a NoSQL

- Los inicios de NoSQL
- ¿Qué es NoSQL?
- Historia de las bases de datos

EIMT.UOC.EDU

Esta presentación se divide en tres secciones principales:

En primer lugar, empezaremos hablando de los hechos que motivaron la creación de las bases de datos NoSQL, para luego responder a la pregunta, ¿Qué es NoSQL?

Acabaremos con un pequeño repaso a la historia de las bases de datos, desde los primeros sistemas diseñados para almacenar y consultar datos, hasta los utilizados actualmente.

Introducción a NoSQL

- Los inicios de NoSQL
- ¿Qué es NoSQL?
- Historia de las bases de datos

EIMT.UOC.EDU

Vamos a empezar, pues, enumerando los hechos que llevaron al desarrollo de las tecnologías NoSQL.

El origen del término

- En 1998 aparece por primera vez la referencia NoSQL, usada por Carlo Strozzi para nombrar su SGBD que no usaba el lenguaje SQL.
- En 2009 reaparece el término, que se utilizará de forma generalizada a partir de ese año, para referirse a los nuevos SGBD, no relacionales y distribuidos, que aparecen durante la primera década del 2000.
- Ese mismo año se crea una comunidad internacional, y muy activa, alrededor de estos nuevos proyectos:
  - El año 2009 con no:sql(east) y el 2010 con no:sql(eu), entre otras, hasta nuestros días.

EIMT.UOC.EDU

El término NoSQL aparece por primera vez en 1998 para denominar la base de datos creada por Carlo Strozzi que no usaba el lenguaje SQL, y que no tiene nada que ver con las bases de datos NoSQL actuales. A partir de 2009 se populariza esta denominación para referirse a una nueva generación de bases de datos no relacionales y altamente distribuidas que comienzan aparecer durante la primera década del 2000.

Aunque muchas de las empresas que desarrollaron y popularizaron esta nueva tecnología son ahora grandes corporaciones (por ejemplo Google o Facebook), la tecnología NoSQL ha estado siempre ligada al auge de las *startup* y del movimiento del código abierto. Existe una comunidad internacional muy activa que incluye conferencias, proyectos, etc.

Es importante destacar que la denominación NoSQL es polémica. Si revisamos los hechos, y con ello los problemas que llevaron al desarrollo de las bases de datos NoSQL, estos no están directamente relacionados con el lenguaje SQL, tal y como el nombre NoSQL puede sugerir. El hecho de que los nuevos modelos de bases de datos creados a partir del 2000 se llamen NoSQL es accidental. Por tanto, siendo rigurosos, debemos tomar el término NoSQL simplemente como una etiqueta que identifica un conjunto de bases de datos de nueva aparición, y no como un acrónimo que signifique “No se usa SQL”, “No SQL”, “No sólo SQL”, ni nada parecido.

Una vez claro el origen y significado del término NoSQL, vamos a hablar sobre los hechos que motivaron su aparición.

Hechos relevantes

- El año 2000 el profesor Eric A. Brewer, durante el Symposium on Principles of Distributed Computing, presenta teorema CAP.
- En el año 2002 se prueba formalmente su validez por los profesores S. Gilbert y N. A. Lynch del MIT.
- Aparecen nuevas necesidades en las empresas y con ellas nuevos proyectos para soportarlas.
- Entre el año 2003 (Google) y 2005 (Marklogic) aparecen los primeros proyectos NoSQL, aunque ya en 1989 hay bases de datos con características similares.

EIMT.UOC.EDU

Durante la primera década del siglo 21 una serie de situaciones propician que se desarrolle una nueva generación de bases de datos. Vamos a intentar describir las principales.

Nos encontramos en pleno auge de Internet. Aparece Google y las primeras empresas de comercio electrónico como, por ejemplo, Amazon o eBay. La irrupción de Internet cambia las reglas del juego apareciendo nuevos modelos de negocio y dando a los datos una importancia aun mayor.

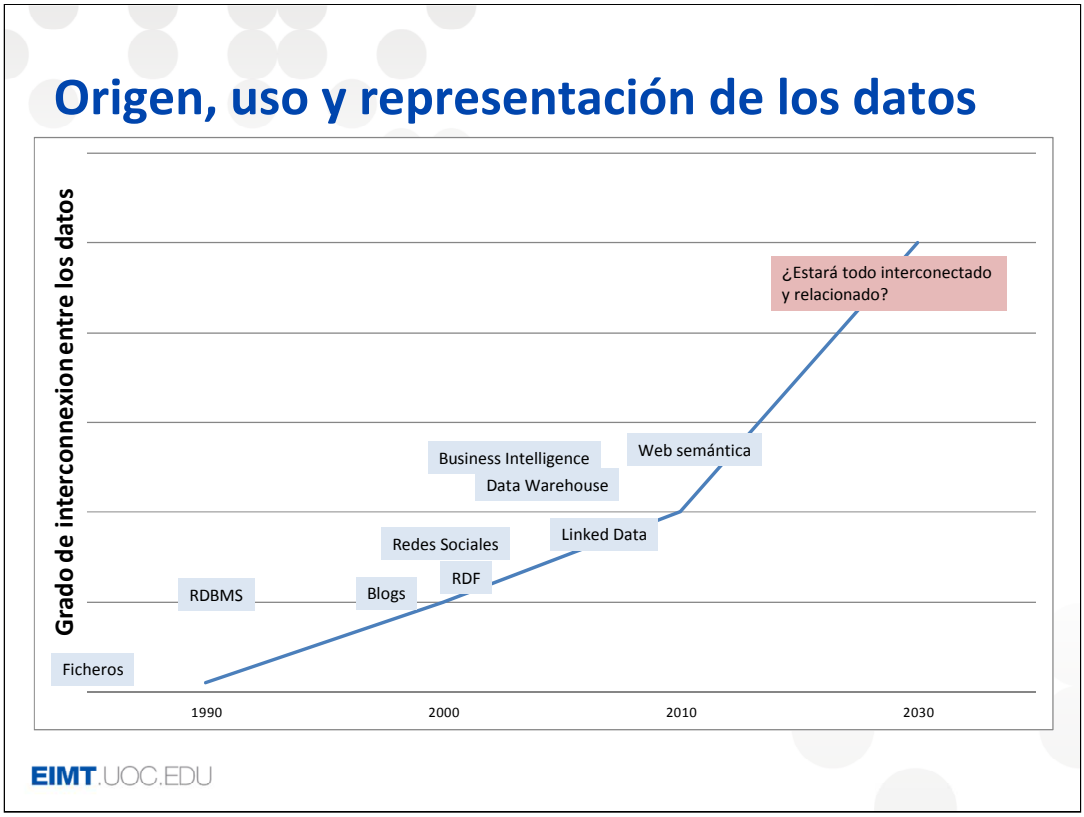
Cada día que pasa, más personas y empresas se conectan a Internet, incrementando los usuarios de la red, los servicios ofrecidos y, en consecuencia, el volumen de datos generado. La gestión de este gran volumen de datos impulsa importantes avances en el desarrollo de sistemas altamente distribuidos como, por ejemplo, el *framework* MapReduce. Para el proceso de este gran volumen de datos, la visión de datos centralizados que ofrecen las bases de datos tradicionales deja de ser eficiente y son necesarios modelos que soporten la distribución de datos de forma masiva.

Otro hecho motivador es la creación de aplicaciones (o servicios) en Internet. Estas aplicaciones ofrecen servicios a un número indeterminado y difícilmente previsible de usuarios. Y no sólo eso, sino que deben responder rápidamente, con independencia de la ubicación del usuario, del número de usuarios conectados al servicio, o de la hora del día que sea. Además, deben proveer de alta disponibilidad para evitar que un fallo del sistema deje sin servicio a un gran número potencial de usuarios.

En este contexto de cambio, el profesor Eric Brewer introdujo el teorema CAP en una charla invitada de un congreso de computación distribuida. Brewer comentó que las propiedades de un sistema distribuido son la Consistencia, la Disponibilidad y la Tolerancia a particiones (precísamente viene de ahí el acrónimo CAP). Y que, desafortunadamente, en un sistema distribuido sólo dos de estas propiedades pueden ser garantizadas de forma simultánea. Es decir, es imposible garantizar las tres a la vez.

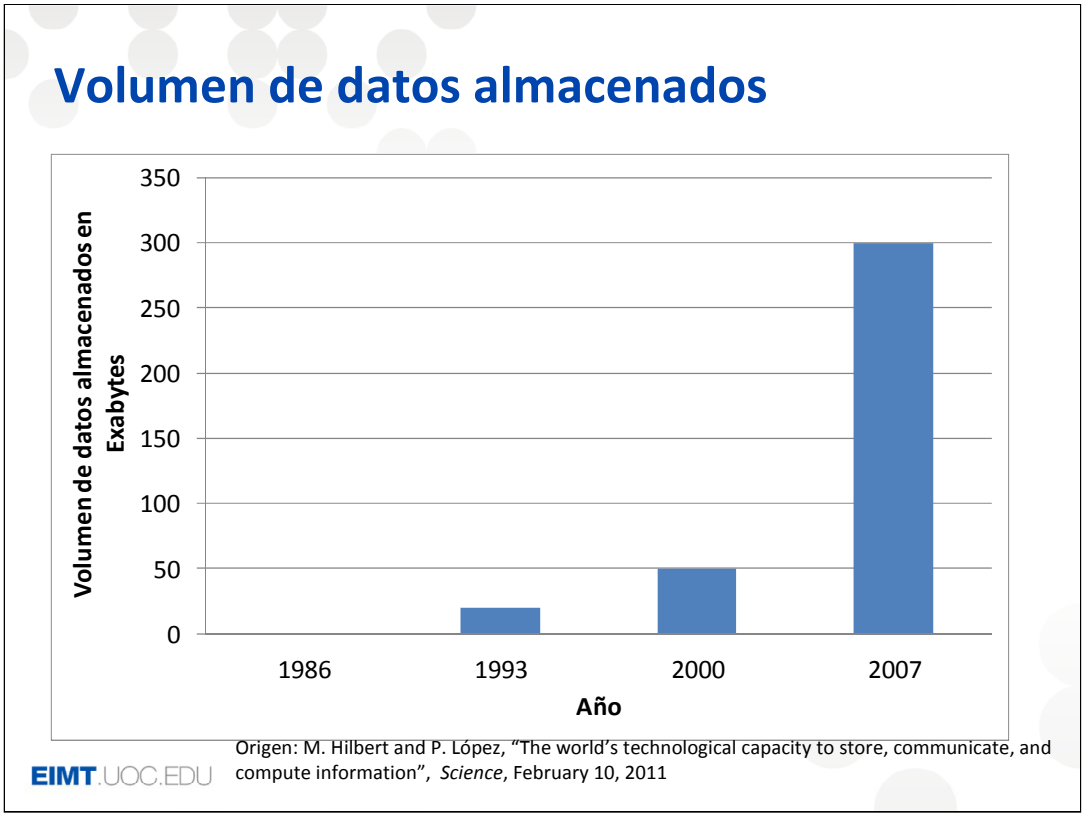
Un par de años después, dos profesores del MIT demostraron la validez del teorema CAP para ciertos modelos de sistemas distribuidos. Este teorema dio pie e inspiró a una nueva generación de bases de datos.

Las primeras bases de datos NoSQL que se conocen son BigTable de Google en 2003 y Marklogic en 2005, aunque ya en 1989 existían bases de datos con características parecidas. Esto nos lleva a una primera característica de las bases de datos NoSQL, son bases de datos creadas a partir del año 2000.



En el nuevo contexto, el tipo y estructura de datos que manejamos ha cambiado.

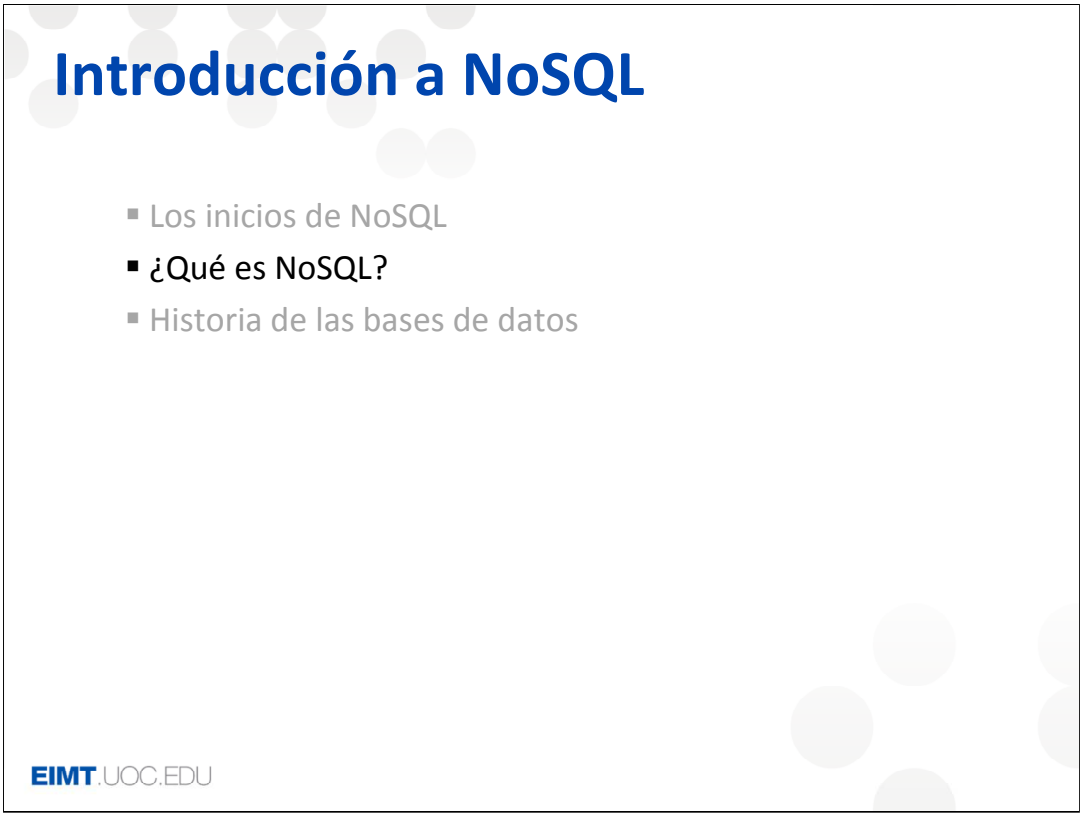
Tal como podemos ver en este gráfico, los datos han evolucionando desde datos que se guardaban en ficheros planos y que acostumbraban a estar poco relacionados, hasta situaciones donde los datos están plagados de relaciones complejas. Este es el caso, por ejemplo, de los datos utilizados en las redes sociales, en la Web semántica o en los sistemas de inteligencia de negocio (*business intelligence*).



Pero no únicamente se incrementan las relaciones entre los datos, sino también su volumen. El número de datos que se deben almacenar y procesar es cada vez mayor. Los motivos son conocidos: aumento de los dispositivos capaces de conectarse a Internet como serían las redes de sensores (Internet of Things), el auge de los teléfonos móviles inteligentes (*smart phones*), etc.

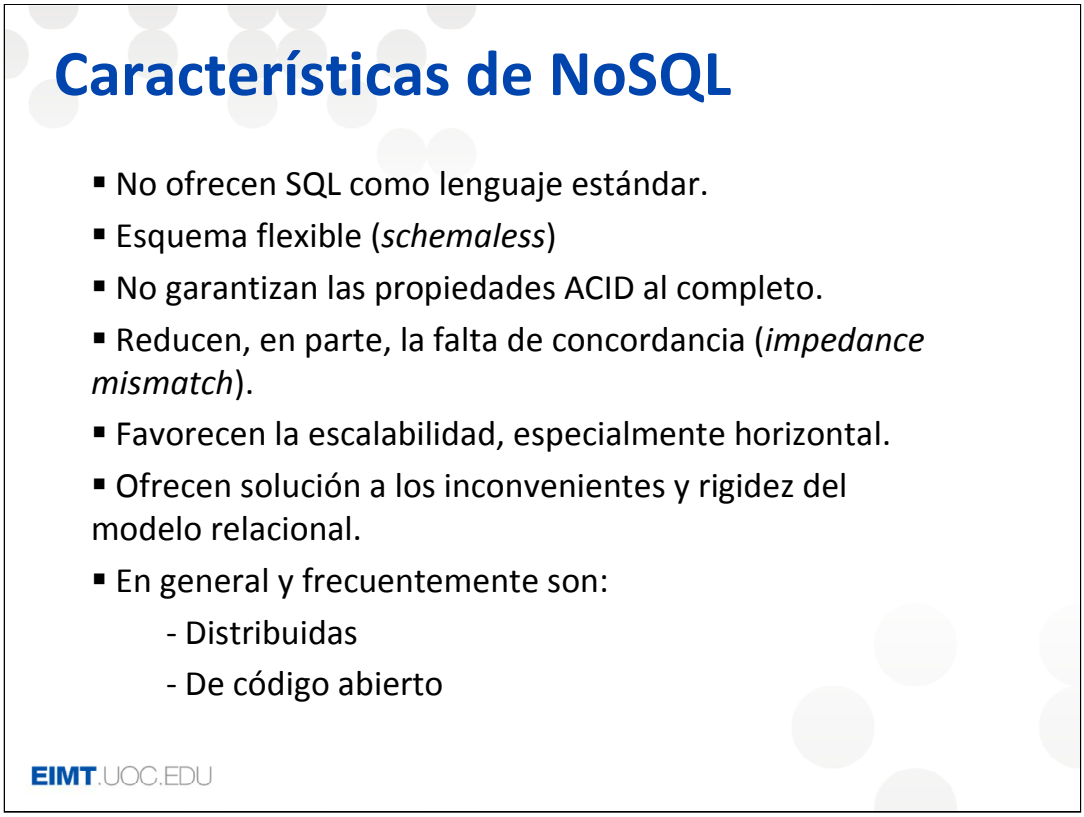
Si analizamos estos dos últimos gráficos, nos damos cuenta de que la situación ha cambiado desde que el modelo relacional se propuso. Esto nos obliga a hacernos la siguiente pregunta ¿Son las bases de datos relacionales los mecanismos más adecuados para procesar este nuevo tipo de datos en el nuevo contexto de alta distribución?

En resumen, la necesidad de almacenar y gestionar grandes volúmenes de datos, altamente relacionados entre sí, en entornos distribuidos y en un ecosistema donde las aplicaciones tienen que responder rápidamente, de forma continuada y para todo el mundo, ha causado que nazca una nueva generación de bases de datos.



Una vez analizados los hechos que favorecieron la creación de las bases de datos de NoSQL, daremos respuesta a la pregunta ¿Qué es NoSQL?.

Esto lo haremos haciendo un repaso de las características, aplicaciones y casos de uso, para acabar con las diferencias más significativas entre el modelo relacional y NoSQL.



Las principales características de NoSQL son:

•No ofrecen SQL como lenguaje estándar: las bases de datos NoSQL engloban a varios tipos de bases de datos, cada uno con su lenguaje de consulta específico. En algunos casos el lenguaje utilizado es SQL, incluyendo algunas extensiones específicas para NoSQL. Eso facilita el acceso a estas bases de datos a gente que sepa utilizar SQL. La gran mayoría de las bases de datos NoSQL se pueden utilizar mediante una API particular de programación, siendo éste el modo de uso generalizado.

•El esquema de datos es flexible o no tienen un esquema predefinido: eso quiere decir que se puede empezar a añadir datos sin definir previamente el esquema de los mismos, como se hace en el modelo relacional. Eso permite mucha más flexibilidad para tratar datos heterogéneos pero dificulta su programación, haciendo que la gestión del esquema (interpretación de los datos) se haga explícitamente en el código de los programas que acceden a la base de datos. El término *schemaless* es un poco abusivo en este contexto, porque aunque es cierto que no existe un esquema explicito, siempre hay un esquema implícito que se usa y condiciona cómo se distribuirán los datos en las bases de datos NoSQL como veremos más adelante. Uno de los riesgos de que el esquema quede implícito es que se compromete la independencia de los datos, uno de los pilares sobre los que se fundamenta el desarrollo de las bases de datos relacionales.

•Las propiedades ACID de las transacciones (Atomicidad, Consistencia, Aislamiento y Definitividad) no siempre se garantizan por completo, esto se hace con el objetivo de mejorar el rendimiento y aumentar la disponibilidad. De hecho, como veremos más adelante, las bases de datos NoSQL acostumbran a seguir otro modelo transaccional denominado BASE.

•Permiten reducir, en parte, los problemas de falta de concordancia (*impedance mismatch*) entre las estructuras de datos usadas en los programas y las bases de datos. Es decir, el formato en que se guarda la información en las bases de datos es cercano al formato utilizado en los programas que acceden a ella.

•Están especialmente diseñadas para crecer, generalmente de forma horizontal (mediante la fragmentación de los datos/esquema y la existencia de copias idénticas de los datos en múltiples servidores).

•Son generalmente distribuidas —con diferentes modelos de distribución (*peer-to-peer*, *master-slave*, etc.)— y de código abierto (con licencias dobles o basadas en soporte).

Las aplicaciones y los datos

- Datos con estructura variable
  - Múltiples orígenes de datos, con diferente formato
  - Altamente relacionados
  - Los datos fluyen en tiempo real.
  - Introducción masiva de BI y *data warehouse*
  - Grandes volúmenes de datos (aka 30 PB).
  - Flexibilidad
  - Rendimiento
  - El proceso de los datos se debe hacer en “tiempo real”.

EIMT.UOC.EDU

Por otro lado, las características de las aplicaciones que usan bases de datos NoSQL suelen ser las siguientes:

•El esquema de los datos es variable, siendo costoso de gestionar en las bases de datos relacionales. Éste es el caso de aplicaciones que gestionan datos con múltiples orígenes y formatos, por ejemplo, un comparador de precios *online*.

•Los datos están altamente relacionados, siendo el ejemplo más clásico las redes sociales, como Facebook.

•Los datos deben procesarse en tiempo real. Por ejemplo, en el caso de Twitter, los sistemas reciben mensajes de los usuarios cuando estos desean enviarlos, pudiendo existir horas punta en situaciones y eventos determinados, pero ello no debe afectar a su rendimiento.

•Los sistemas de *data warehousing* tienen requisitos distintos a los de las aplicaciones transaccionales clásicas. Si a esto le añadimos una capacidad analítica más potente, donde se realizan más análisis y utilizando más datos (los de la competencia, por ejemplo, o los provenientes de las redes sociales), nos lleva a sistemas que pueden ser difícilmente gestionables bajo un entorno relacional.

•Trabajan con grandes volúmenes de datos, hablamos de magnitudes superiores.

•El rendimiento y flexibilidad son cruciales. Una tienda *online* fuera de línea o que tarde demasiado en atender al cliente es una tienda que se percibirá como cerrada y que no realizará la venta. Esto es muy peligroso teniendo en cuenta que en Internet la competencia sólo está a un *click* de distancia.

•Los datos se deben procesar en tiempo casi real, ejecutando las operaciones necesarias rápidamente y de esta manera garantizar una experiencia óptima al usuario. Por ejemplo, en aplicaciones de bolsa, los *brokers* deben tener información fidedigna rápidamente que les permita tomar las mejores decisiones.

Diferencias entre NoSQL y bases de datos relacionales

- No hay un modelo de datos único:

  NoSQL ofrece diferentes modelos de datos, mientras que las bases de datos relacionales sólo proporcionan el modelo relacional.

- Esquema flexible (*schemaless*):

  A diferencia del modelo relacional en NoSQL no es necesario un esquema predefinido que defina como se estructuran y relacionan los datos.

- Falta de estandares:

  Mientras que detrás de las bases de datos relacionales existen estándares, éste no es el caso de las BD NoSQL (de momento).

EIMT.UOC.EDU

Las bases de datos relacionales se diferencian de las bases de datos NoSQL principalmente en los siguientes puntos:

• No hay un modelo de datos único: El modelo relacional ofrece una visión uniforme de los datos, la relación, mientras que las bases de datos NoSQL engloban a muchos modelos de datos como, por ejemplo, el modelo de agregación, el de grafos, etc. La existencia de diferentes modelos de datos causa la existencia de diferentes familias de bases de datos NoSQL.

•En el modelo relacional es necesario definir, a priori, un esquema conceptual que indique qué datos hay, cómo se estructuran y relacionan. Eso no es necesario en la mayoría de modelos NoSQL (lo que se denomina *schemaless*). Las implicaciones de ello es que los modelos NoSQL no disponen de independencia de datos y la dificultad de definir restricciones de integridad en la base de datos. Es decir, el usuario/programa de la base de datos es el encargado de interpretar y gestionar los datos. Por otro lado, la ventaja es que el proceso de datos heterogéneos es más simple en las bases de datos NoSQL.

•Las bases de datos relacionales están entre nosotros desde hace muchos años, habiendo dado tiempo a la creación de estándares, por ejemplo, el lenguaje SQL. Conviene tener en cuenta que esto aún no ha sido así para las bases de datos NoSQL. A pesar de ello, sí que encontramos estándares de facto como, por ejemplo, el uso del formato JSON.

Aunque estas diferencias son ciertas para la mayoría de bases NoSQL, hay multitud de bases de datos y cada una de ellas tiene funcionalidades distintas. Por tanto, el oyente debería tomar lo dicho hasta ahora simplemente como una guía general, que puede ser más o menos cierta dependiendo del producto NoSQL a considerar.

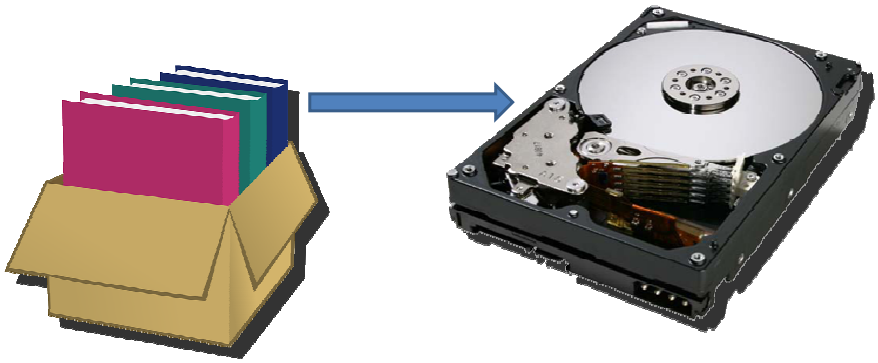
Introducción a NoSQL

- Los inicios de NoSQL
- ¿Qué es NoSQL?
- Historia de las bases de datos

EIMT.UOC.EDU

Ahora que ya sabemos qué es NoSQL y porque apareció echaremos la vista atrás para recordar cuál ha sido la evolución de la tecnología de bases de datos. Descubriremos que las ideas básicas de NoSQL no son del todo nuevas, ya éstas han estado presentes, de un modo u otro, desde hace tiempo.

De los ficheros a las bases de datos



Todo empieza utilizando ficheros de texto o binarios en el disco duro, hasta que aparece la tecnología de bases de datos para organizar los datos y facilitar el acceso a los mismos.

EIMT.UOC.EDU

Desde los inicios de la computación, ha existido la necesidad de organizar y almacenar los datos de forma permanente para su manipulación posterior. Para ello, las primeras computadoras (que ya no usaban tarjetas perforadas) empezaron a utilizar ficheros de texto o binarios, pero luego la consulta de los datos y la gestión de las relaciones entre los datos era compleja.

Como ya se intuye, la idea no escala. Si los datos aumentan y las relaciones entre los mismos son más complejas, la gestión de los ficheros se hace más difícil también. Por ello se creó software específico para la gestión de los datos. De esta manera aparecen las primeras bases de datos, que ofrecen estructuras de datos (índices) y/o lenguajes (por ejemplo, SQL) permiten almacenar y gestionar los datos en ficheros de forma más eficiente.

Primeros pasos

- En 1960 aparecen las primeras bases de datos jerárquicas y en red.
- En 1970 el profesor Edgar F. Codd propone el modelo de datos relacional y el álgebra relacional, que darían lugar a las bases de datos relacionales.
- A finales de la década de 1970 aparecen los primeros sistemas gestores de bases de datos (SGBD) relacionales: DB2, Ingres, Sybase, entre otros.

EIMT.UOC.EDU

En 1960 aparecen las primeras bases de datos, las jerárquicas y en red. En ellas los datos se organizan en forma de árboles y grafos, respectivamente. Estos primeros sistemas fueron introducidos por Charles Bachman, líder del Database Task Group en el CODASYL. Estos sistemas eran muy complejos, por lo que en 1969 IBM crea IMS, el que se considera el primer sistema gestor de bases de datos de propósito general. IMS fue usado desde aplicaciones ligadas al sector de la aviación, hasta su uso masivo en el sector de banca.

Dado que el desarrollo de aplicaciones que usaban estas bases de datos era muy costoso, en el año 1970 el profesor Edgar F. Codd introduce el modelo de datos relacional, junto al álgebra relacional, que darán lugar a las bases de datos relacionales. A diferencia de los modelos anteriores, se trata de un modelo sencillo pero completamente formalizado, con claras influencias de la lógica matemática y el álgebra de conjuntos. Por estos motivos, el modelo relacional despertó el interés de la comunidad científica (y de la industria), cambiando gradualmente el paisaje comercial que hasta entonces existía, pasando a ser el paradigma predominante.

Las primeras bases de datos que implementan el modelo relacional aparecen a finales de la década de los 70, por ejemplo, este es el caso de DB2, Ingres y Sybase, entre otros. Con sus posteriores evoluciones, algunas de estas bases de datos han llegado hasta nuestros días (por ejemplo, DB2 y Oracle).

Los años 80 y 90

- A finales de la década de los 80, y principios de los 90, junto con el auge de la programación orientada a objetos, aparecen las bases de datos orientadas a objetos.
- La discordancia entre el modelo de programación y el modelo relacional (*impedance mismatch*) favorecen su popularidad.
- Con el mismo objetivo aparecen los ORM (herramientas de mapeo objeto-relacional).
- Aunque poco utilizadas, durante la década de los 90, algunas bases de datos relacionales ofrecen extensiones orientadas a objetos.

EIMT.UOC.EDU

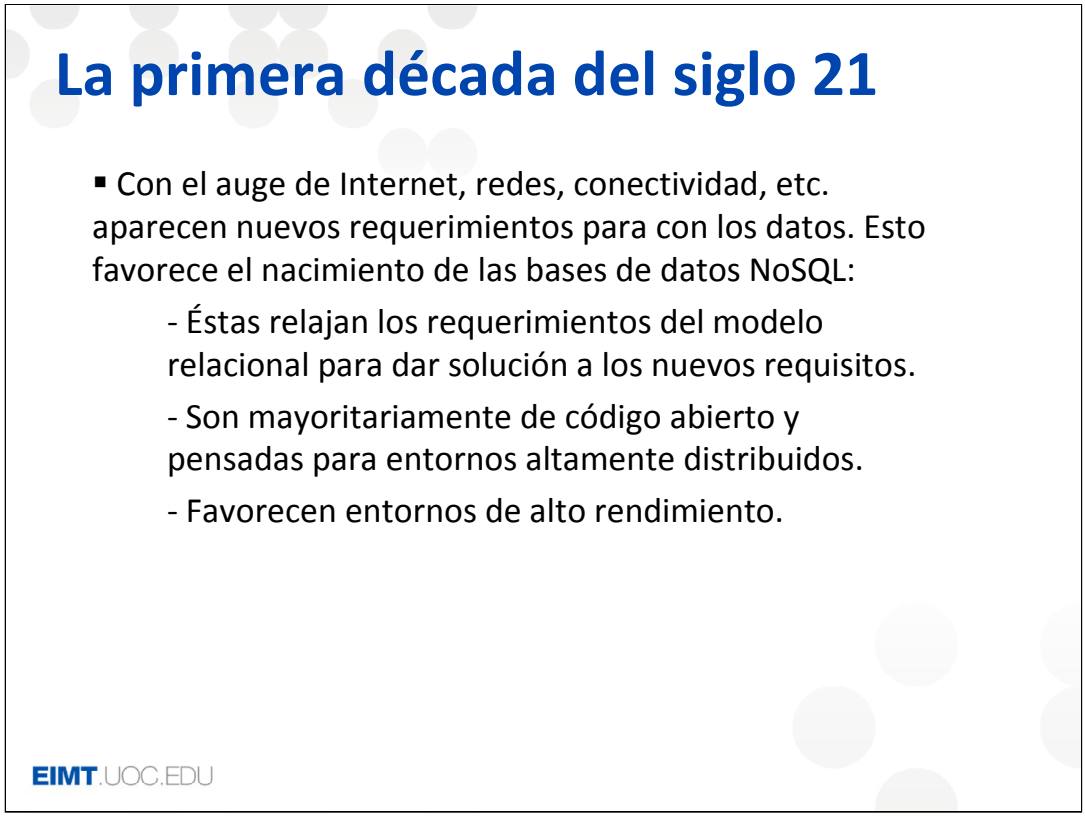
Aunque el modelo de base de datos predominante es el relacional, durante la década de los 80 y 90 la evolución de las bases de datos no se para y se introducen las bases de datos orientadas a objetos y las objeto-relacional.

Estas bases de datos aparecen debido al auge de la programación orientada a objetos y los problemas para almacenar objetos (objetos y clases complejas, relaciones de herencia etc.) en una base de datos relacional. Este problema de diferente representación entre los datos procesados por los programas y por las bases de datos es lo que se conoce como *impedance mismatch*.

Algunas alternativas para tratar de evitar los problemas de no concordancia son las herramientas ORM, de mapeo objeto-relacional. Una herramienta ORM es una librería creada con el objetivo de crear el mejor modelo relacional para almacenar un modelo orientado a objetos. Estas herramientas facilitan la creación y gestión del modelo, pero también la consulta de los datos de forma integrada con el lenguaje de programación utilizado.

Aunque muy poco utilizadas, durante la década de los 90, algunas bases de datos relacionales también ofrecen extensiones especificas de orientación a objetos para mejorar la expresividad semántica del modelo de datos relacional (por ejemplo, entre estas extensiones está la posibilidad de definir relaciones de herencia entre relaciones). Son las llamadas bases de datos objeto-relacional.

Esta idea de proporcionar extensiones al modelo relacional se utiliza también para dar solución a otros tipos de problemas, provocando la creación de las BD geográficas, los sistemas OLAP, etc. Estas extensiones son ampliamente utilizadas en la actualidad.



Durante la primera década del siglo 21 con el auge de Internet, las redes sociales, la conectividad y los teléfonos móviles inteligentes, aparecen nuevos requerimientos para los que las bases de datos relacionales ya no necesariamente ofrecen la mejor solución. Con este objetivo, nacen las bases de datos NoSQL. Éstas se caracterizan por ser variadas, versátiles y ligeras, y surgen con el objetivo de ofrecen un mejor rendimiento, disponibilidad y escalabilidad. A menudo prescinden de algunas de las funcionalidades clásicas de una base de datos relacional.

Tal como ya hemos visto, las bases de datos NoSQL no son en general una idea del todo nueva. Desde el principio de las bases de datos ya existen bases de datos con gran parecido a las actuales NoSQL, aunque ahora se han dado las circunstancias que han propiciado su desarrollo y posterior popularización.

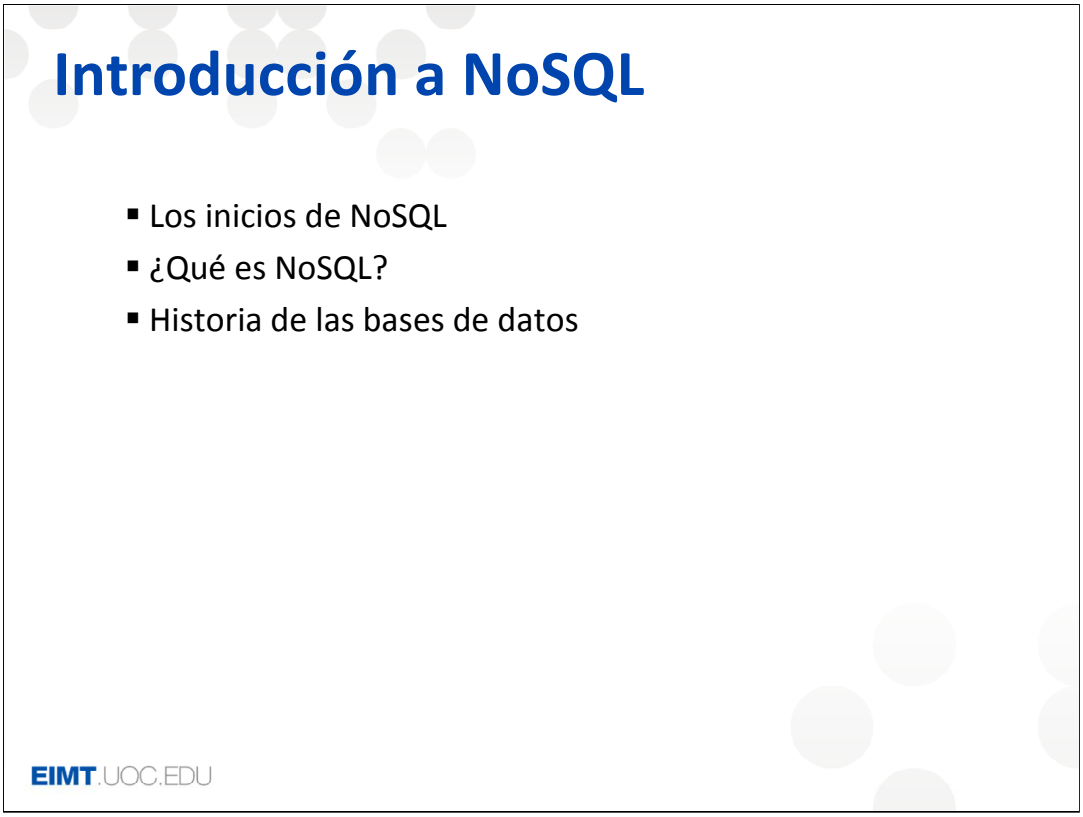
La actualidad

- La tecnología de bases de datos, junto con el del procesamiento de los datos, están en continuo desarrollo. Nuevos productos y estrategias se crean constantemente.
- Existe un debate abierto en la comunidad sobre las ventajas y desventajas de las diferentes bases de datos:
  - Bases de datos relacionales versus NoSQL
  - Entre los diferentes tipos de NoSQL

EIMT.UOC.EDU

Si una cosa nos ha enseñado la historia es que la tecnología está en continua evolución y éste es también el caso de la tecnología de bases de datos. Esta evolución está guiada generalmente por los requerimientos y necesidades de cada momento.

Pero como sucede con cualquier tecnología nueva, existe un debate abierto en la comunidad sobre sus ventajas e inconvenientes. Lo único seguro es que gracias a este debate continuo, en un futuro gozaremos de un mayor conocimiento, pudiendo elegir con más criterio que herramienta es la que mejor se adapta a nuestras necesidades.



Hasta aquí esta pequeña introducción a NoSQL. En este video hemos visto los cambios en el entorno que han motivado la aparición de las bases de datos NoSQL, hemos presentado las principales características de las bases de datos NoSQL y sus principales diferencias con las relacionales y hemos contextualizado su aparición en relación con la historia de las bases de datos.

Referencias

C. Coronel, S. Morris & P. Rob (2013). *Database Systems: Design, Implementation and Management 10e*. Course Technology, Cengage Learning.

B. Grad & T.J. Bergin (2009). "History of Database Management Systems", *IEEE Annals of the History of Computing*, 31(4), pp 3-5. (http://bit.ly/18gmGTX).

A. Popescu. *NoSQL and Polyglot Persistence*. (http://bit.ly/1ggTT4k).

P.J. Sadalage & M. Fowler (2013). *NoSQL Distilled. A brief Guide to the Emerging World of Polyglot Persistence*, Pearson Education. (http://bit.ly/1koKhBZ).

P. Yang (2011). "Moving an Elephant: Large Scale Hadoop Data Migration at Facebook" (https://www.facebook.com/notes/paul-yang/moving-an-elephant-large-scale-hadoop-data-migration-at-facebook/10150246275318920).

EIMT.UOC.EDU

Esperamos que hayáis disfrutado y aprendido con este vídeo introductorio.

A continuación encontraréis algunas referencias que os permitirán profundizar más en el origen de esta apasionante tecnología.

Que tengáis un buen día.