



# BASE DE DATOS - NOSQL

NO SOLO SQL

# AGENDA

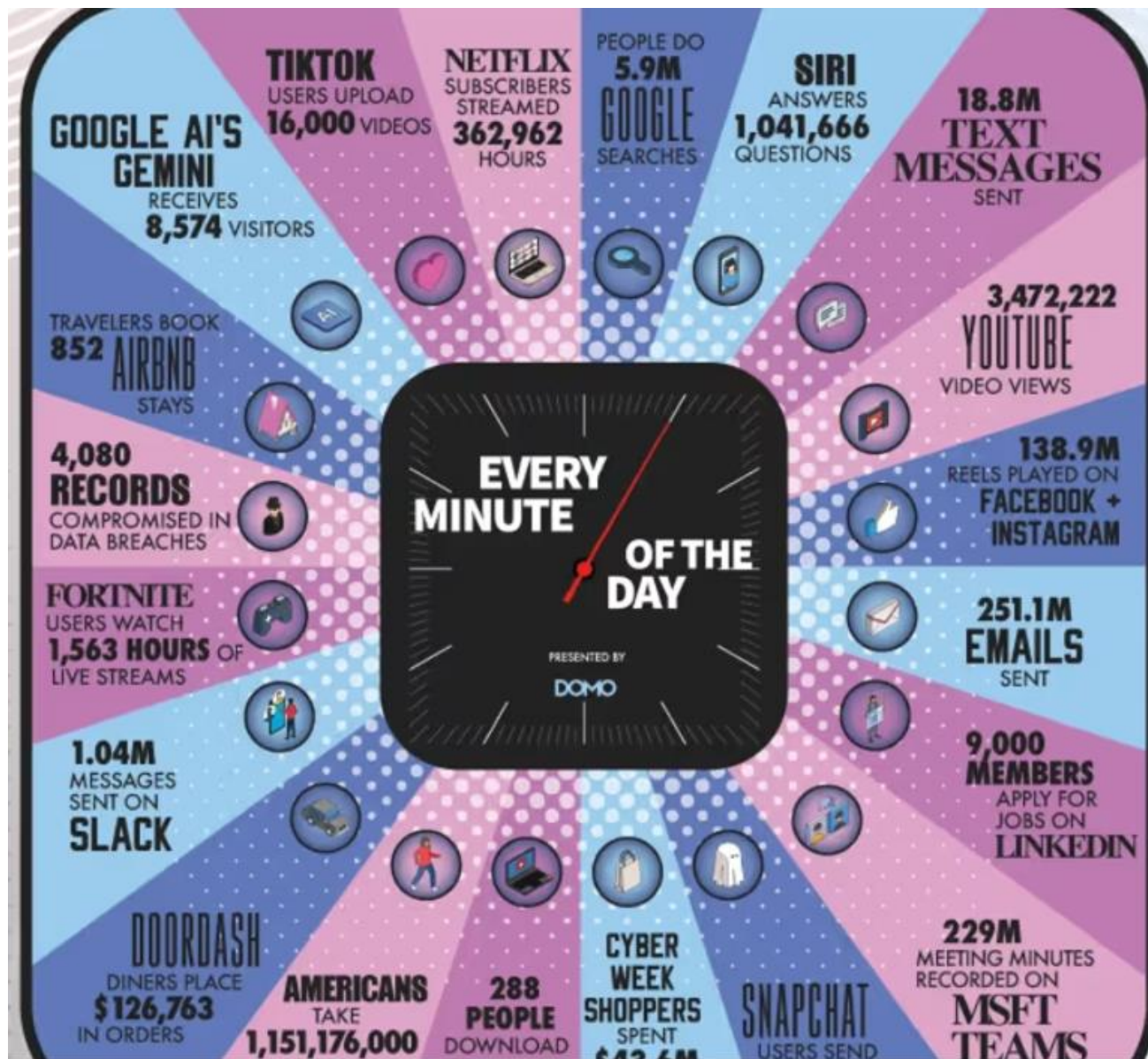
- Bases de datos relacionales
  - Panorama actual
  - Limitaciones
- Bases de datos NoSQL (no solo SQL)
  - Introducción
  - Propiedades ACID vs. BASE
  - Categorías de Bases de Datos NoSQL
    - Almacenamiento clave/valor
    - Almacenamiento en grafos
    - Almacenamiento documental
    - Almacenamiento en familia de columnas



# BASES DE DATOS - PANORAMA ACTUAL



- El panorama actual difiere del de hace algunas décadas.
  - En los 90 los manejadores de bases de datos relacionales se anunciaban como "escalables", pero en ese momento nunca imaginaron la cantidad de bytes que algunas empresas o entidades necesitan administrar día a día.
  - El auge de internet desde los 90 hasta la actualidad hace que nos encontremos en un mundo hiperconectados.
  - Se generan millones de datos por minuto, a través de diversas aplicaciones, redes sociales, blogs, foros, entre muchos otros sistemas.
- En este contexto, entender a las bases de datos solamente como un lugar de almacenamiento es un grave error.
- La dinámica ha cambiado en dirección a un sistema globalizado, en donde empresas, universidades e institutos incorporan las redes sociales para compartir su información, en donde los bancos se apoyan en el comercio electrónico y en el acceso a sus servicios a través de internet y en donde los objetos de la vida cotidiana se conectan entre sí (The Internet of Things).



# EL MUNDO DE HOY

[HTTPS://WWW.DOMO.COM/LEARN/INFOGRAPHIC/DATA-NEVER-SLEEPS-12](https://www.domo.com/learn/infographic/data-never-sleeps-12)

# BASES DE DATOS - LIMITACIONES



- Continuar utilizando bases de datos relacionales para todos los contextos de información es simplemente inviable.
- Existen algunos problemas con los RDBMS actuales que pueden suponer una seria limitante para la construcción de ciertos tipos de aplicaciones.
  - Transacciones para todo.
  - Escalabilidad costosa.
  - Todo se representa como un modelo relacional.
  - Operaciones y tipos de datos limitados.
  - Lecturas costosas.
  - Millones de usuarios.
  - Incompatibilidad con el modelo orientado a objetos.

# BASES DE DATOS NO RELACIONALES - INTRODUCCIÓN

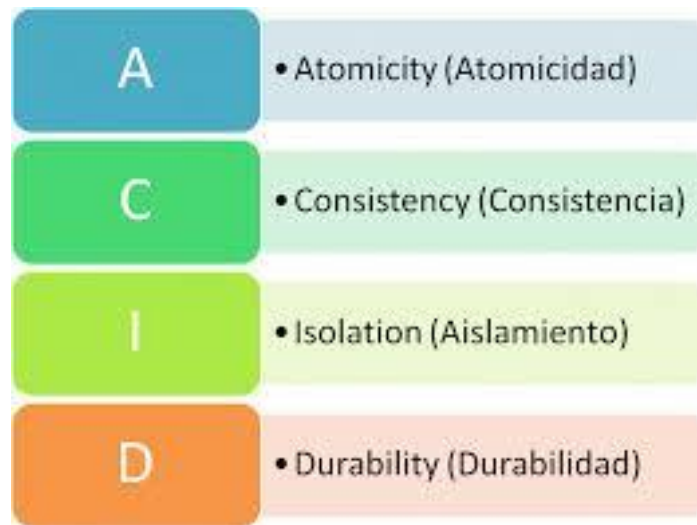
- En lo que se refiere a la gestión de los datos, los arquitectos y los desarrolladores de software se han visto en la necesidad de buscar soluciones que se adapten a distintos criterios para la consolidación (persistencia), análisis y explotación de los datos, con el fin de mejorar la escalabilidad, la partición y replicación de la información y la rigidez de las estructuras de dichos datos.
- NoSQL (también conocido como “no solo SQL”) comprende a un grupo de sistemas de bases de datos que se distingue de los tradicionales sistemas de gestión de bases de datos relacionales en diversos aspectos, siendo el más notorio, el de no poseer un lenguaje SQL como lenguaje principal de consultas.
- No requieren de una estructura fija y tabular, no siempre soportan las operaciones de JOIN.
- No siempre garantizan las propiedades de ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad)
- En general son adecuadas para escalar horizontalmente.

# BASES DE DATOS NOSQL

## INTRODUCCIÓN

- Las bases de datos NoSQL responden a la evolución de la información. No son un tipo de Bases de Datos, sino que son un conjunto de tipos de bases de datos.
- Aunque parece apresurado, todo apunta a que son el repositorio de datos del futuro.
- Sin embargo, aún hay problemas que retrasan este hecho. Uno de ellos es la resistencia al cambio. Hay que tener en cuenta que cuestiones como la consistencia o la integridad de los datos son conceptos que se encuentran muy apegados a las Bases de Datos Relacionales.
- No obstante, hay bases de datos NoSQL que incorporan propiedades de las bases de datos relacionales.

# ACID VS. BASE



- Las propiedades de ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), son muy importantes en un sistema de base de datos relacional.
- Son las propiedades necesarias para garantizar que un conjunto de instrucciones sea considerado como una transacción, es decir, una única unidad lógica de trabajo (“o de negocio”).
- Poner en práctica las propiedades de ACID en un sistema de bases de datos relacional no es una tarea sencilla, involucra un conjunto de acciones que consumen tiempo durante el procesamiento de la transacción. Mantener estas propiedades se vuelve aún más complicado para un ambiente de red.

ACID “relajado”: con el fin de proporcionar una gran escalabilidad, puede ser necesario relajarse o redefinir algunas de las cualidades de este modelo, en particular consistencia y durabilidad.

Para un sistema de base de datos NoSQL es necesario sacrificar algunas características de las bases de datos relacionales para obtener nuevas prestaciones o mejoras en algunos aspectos sin afectar el rendimiento del sistema.

NoSQL propone un sistema llamado “BASE (Basically Available, Soft State, Eventually Consistent)”.

BASE propone una visión más liviana y optimista del sistema, acepta que la consistencia en la base de datos estará en un estado de flujo o que la disponibilidad del sistema se logra mediante el apoyo a fallos parciales sin que el sistema completo se detenga.

## ACID VS. BASE

## PROPIEDADES BASE

- **Base Availability** (Disponibilidad básica): el sistema se encuentra disponible la mayoría del tiempo. El sistema puede estar temporalmente inconsistente para que las transacciones luego se puedan mezclar y habrá una respuesta a cualquier solicitud, aunque podría ser una respuesta de que la operación no ha podido ser realizada.
- **Soft State** (Estado suave): los datos en los diferentes nodos de la red no tienen que ser mutuamente consistentes en todo momento. Hay que aceptar cierta inexactitud temporal. Incluso durante los tiempos sin entrada puede haber cambios debido a la "consistencia final".
- **Eventual Consistency** (Consistencia eventual): el sistema eventualmente se volverá consistente una vez que deje de recibir información. Tarde o temprano, Los datos se propagan a todos los lugares donde debería. El sistema continuará recibiendo información y no verificará la consistencia de cada transacción antes de pasar a la siguiente.

### Basically Available

- Prioridad de la disponibilidad de los datos

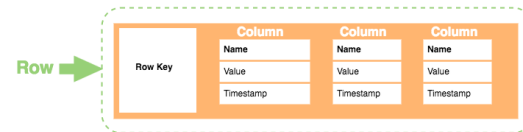
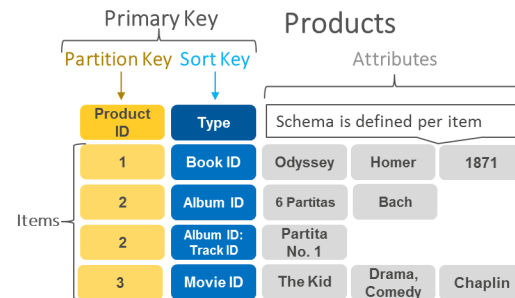
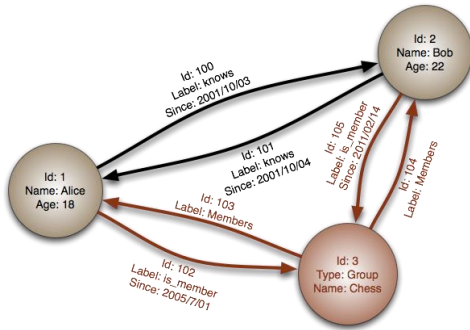
### Soft state

- Se prioriza la propagación de datos, delegando el control de inconsistencias a elementos externos

### Eventual consistency

- Se asume que inconsistencias temporales progresen a un estado final estable.

# BASES DE DATOS NOSQL - CATEGORÍAS



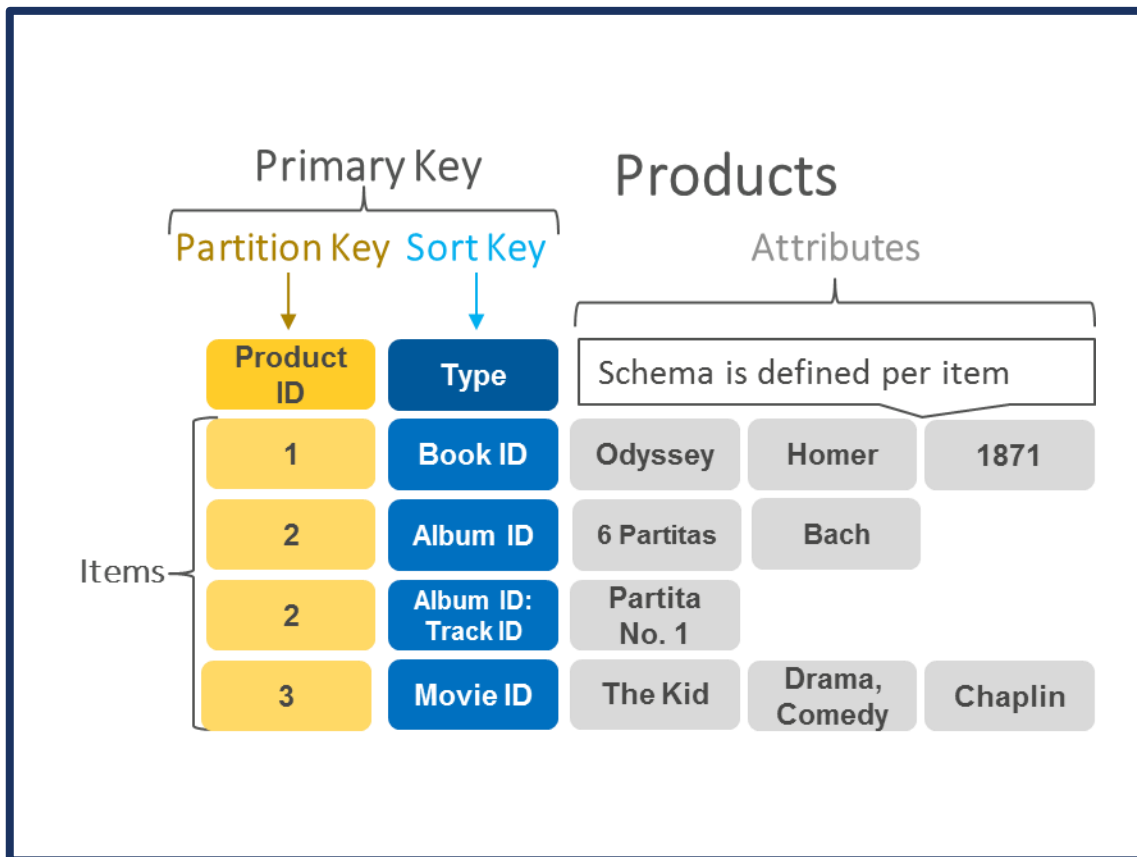
- Aunque existen muchas tecnologías en bases de datos NoSQL, se pueden destacar cuatro:
  - Almacenamiento Clave/Valor.
  - Almacenamiento de Documentos.
  - Almacenamiento de Familia de Columnas.
  - Almacenamiento de Grafos.

# BASES DE DATOS NOSQL – CLAVE/VALOR



- Son las más simples en cuanto a su implementación, almacenan datos como un conjunto de pares “clave/valor” (key-value), en los que la clave representa un identificador único, de modo que un valor simple dado, llamado clave, puede retornar un objeto complejo y arbitrario de información (valor). Tabla de hash.
- Uno de los mayores beneficios es no tener que especificar el formato a almacenar. Será responsabilidad de la aplicación saber con que trabajará, texto libre, un archivo XML o un archivo binario.

# BASES DE DATOS NOSQL – CLAVE/VALOR - RESTRICCIONES



- Una base de datos clave/valor poseen dos restricciones importantes:
  - **Claves distintas:** Nunca se van a poder almacenar dos filas con la misma clave, en otras palabras, las claves deben ser únicas.
  - **No hay consultas sobre los valores:** no se pueden realizar consultas sobre un rango de valores o alguna característica particular de los valores de la base. Esta es la principal complejidad a la hora de utilizar esta tecnología.

# BASES DE DATOS NOSQL – CLAVE/VALOR - CONTEXTOS

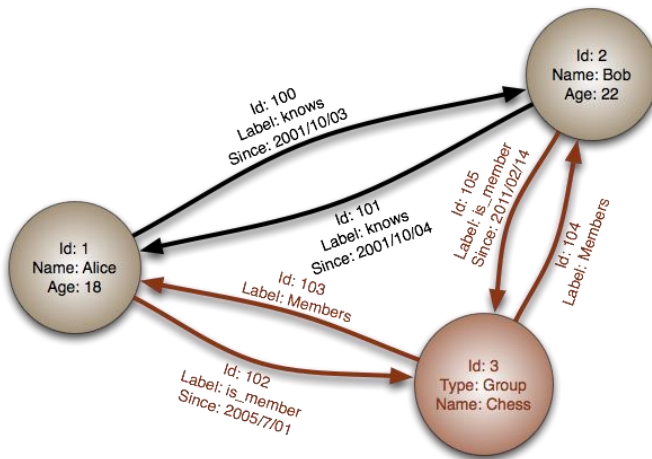
- Contextos en los que una base de datos NoSQL clave/valor se adapta de forma correcta o es beneficioso su utilización:
  - **Cuando se necesita guardar información de sesión:** dado que cada sesión web es única y se le asigna un id de sesión único.
  - **Perfiles de usuario y sus preferencias:** se almacena información por identificador de usuario y en ellas se guardan las preferencias del mismo.
  - **Carrito de compras:** se crea un carrito de compras con su información correspondiente.
- Además, hay contextos en los que no es beneficioso o no es posible la utilización de una base de datos NoSQL clave/valor:
  - **Relaciones entre la información:** cuando se tienen relaciones entre diferentes conjuntos de información.
  - **Consultas por valores:** no permiten consultar valores por un valor particular del conjunto.
  - **Operaciones por conjuntos:** dado que las operaciones están limitadas de una clave por vez, no se puede operar en muchas claves al mismo tiempo.

# BASES DE DATOS NOSQL – GRAFOS



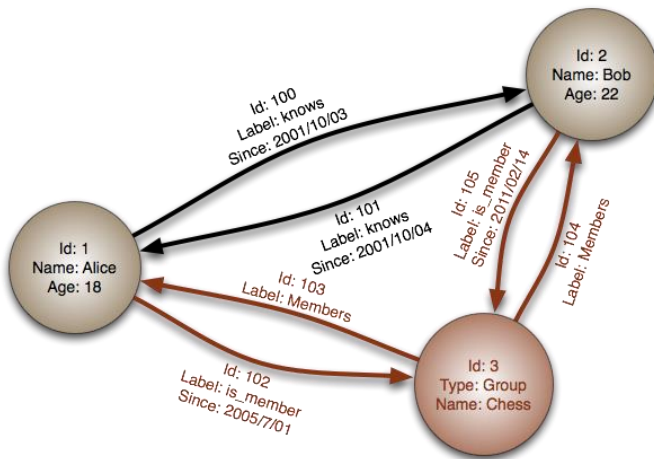
- Un grafo básicamente es un conjunto de puntos (vértices) en el espacio que están conectados por un conjunto de líneas (aristas).
- Una base de datos orientada a grafos (BDOG) es aquella que permite almacenar la información como nodos de un grafo y sus respectivas relaciones con otros nodos, permitiendo así aplicar la teoría de grafos para recorrer la base de datos; son muy útiles para guardar información en modelos con muchas relaciones entre distintas entidades o nodos
- La utilización de este tipo de base de datos depende del contexto de la información y la lógica de negocio, esto se debe a que aplicar este tipo de base de datos en un escenario no apropiado hará que no se aproveche todo su potencial.

# BASES DE DATOS NOSQL – GRAFOS – CONTEXTOS



- Escenarios adecuados para que una base de datos NoSQL se adapte de manera natural:
  - **Datos interconectados:** pueden ser utilizadas para implementar redes sociales.
  - **Servicios basados en la ubicación:** toda dirección o ubicación que tiene un envío puede ser considerada un nodo y todos los nodos que el envío tiene que hacer pueden ser representados por un grafo. Las relaciones tendrán propiedades como la distancia.
  - **Sistemas de recomendaciones:** gracias a que los nodos y las relaciones son cargadas en el sistema, ellas pueden ser utilizadas para realizar reconocimientos del estilo de “tus amigos también compraron este producto” o “cuando se compra este producto también se suele comprar”. También se puede analizar ciertos patrones de la transacción para analizar la posibilidad de estar ante un caso fraudulento.

# BASES DE DATOS NOSQL – GRAFOS – CONTEXTOS



- Cuando no es adecuado utilizar un tipo bases de datos orientadas a grafos:
  - Cuando se requiere actualizar todas o un conjunto de entidades al mismo tiempo. Las operaciones de modificar la estructura de los nodos implican mucho trabajo.
  - Son bases de datos de estructura compleja a causa de los grafos que utilizan. No deben ser tomadas a la ligera, si no se ven grandes beneficios en su aplicación no es recomendable utilizarlas.

# BASES DE DATOS NOSQL – DOCUMENTALES – CONTEXTOS



- El concepto central de este tipo de almacenamiento es el documento. Almacena, recupera y gestiona dato de documentos o datos de algún modo estructurado.
- El factor común en este tipo de bases de datos es que los documentos encapsulan y codifican datos bajo algún formato estándar (XML, YAML, JSON, BSON).
- El documento en una base de datos documental puede ser visto, como las tuplas de las bases datos relacionales, pero menos rígidos, debido a que no deben ajustarse a ningún esquema específico.
- Cada documento podría tener sus propias secciones y diferir uno de otro.

# BASES DE DATOS NOSQL – DOCUMENTALES - CONTEXTOS



## Relational data model

Highly-structured table organization with rigidly-defined data formats and record structure.



## Document data model

Collection of complex documents with arbitrary, nested data formats and varying "record" format.

- La forma de almacenamiento es muy similar al de las bases de datos clave/valor. Los documentos son direccionados mediante una clave o llave única.
- La clave, en general, es una simple cadena, pudiendo ser una URL o un camino para recuperar el documento almacenado.
- Una de las diferencias con las bases de datos clave/valor es la posibilidad de utilizar el contenido de los elementos almacenados, para poder indexarlos y procesarlos para lograr una recuperación rápida.

# BASES DE DATOS NOSQL – DOCUMENTALES – BENEFICIOS



- Los principales beneficios de utilizar este tipo de base de datos están dados por las facilidades y la flexibilidad de sus estructuras (esquemas libres).
- No todos los documentos de una misma colección tienen que tener la misma estructura, puede existir atributos en un documento que otro no tenga y viceversa.
- Libertad para agregar atributos a un documento dentro de una colección. No hace falta definirlos previamente, por lo que se le pueden agregar atributos en cualquier momento.

# BASES DE DATOS NOSQL – DOCUMENTALES – BENEFICIOS



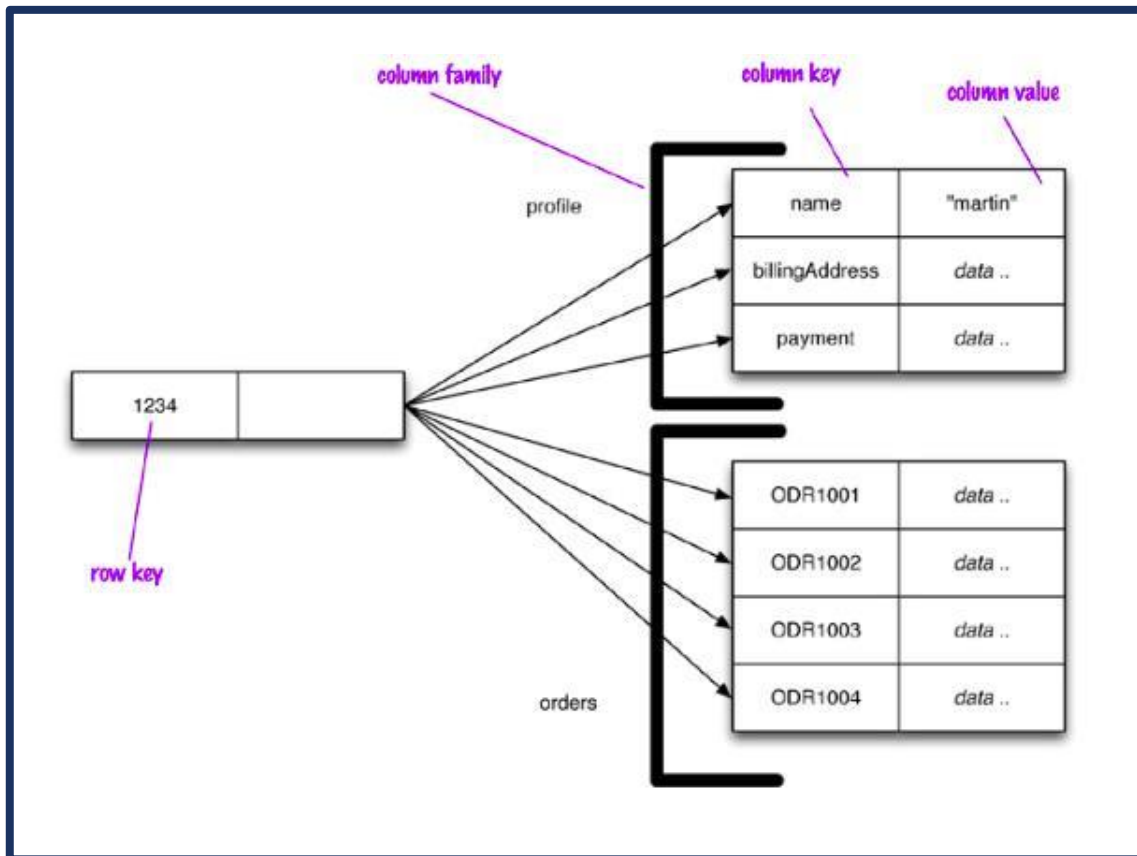
- Los documentos son unidades independientes y autocontenidos. No hay necesidad de normalizar la información para poder almacenarla.
- La lógica de la aplicación es más fácil de escribir. No hay traducciones entre los objetos del modelo y las consultas.
- Interacciones ágiles. En etapas tempranas de cualquier proyecto es difícil entender totalmente el modelo del negocio y el dominio puede ir cambiando a lo largo del tiempo de vida del proyecto.

# BASES DE DATOS NOSQL – DOCUMENTALES – USOS



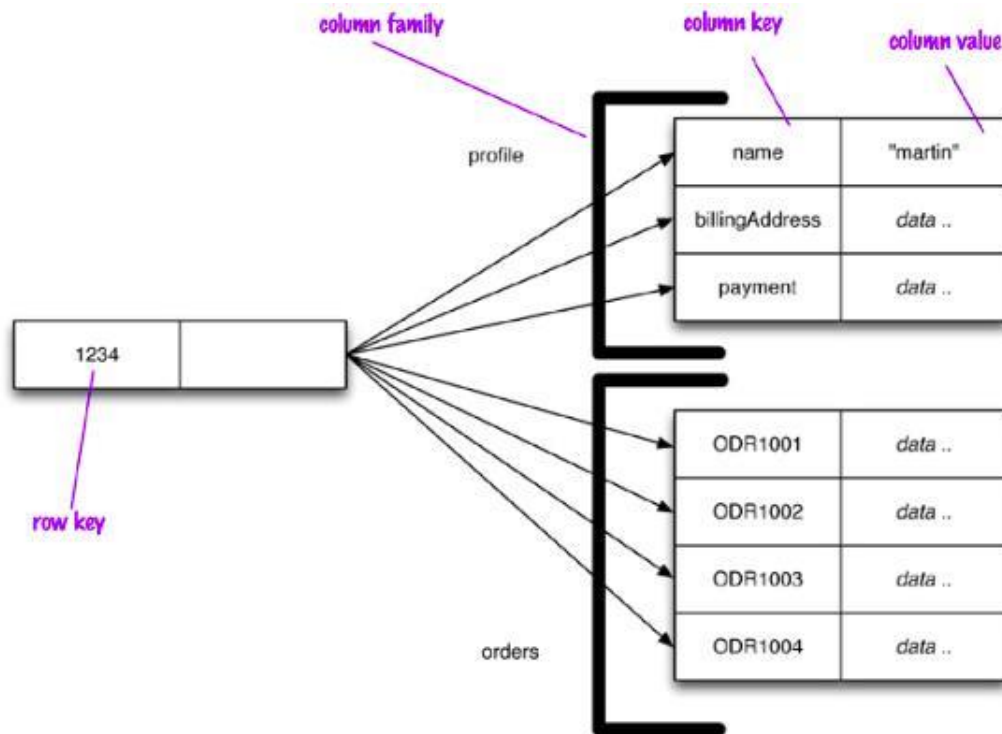
- Registros de eventos.
- Análisis en tiempo real.
- Aplicaciones de comercio electrónico.

# BASES DE DATOS NOSQL – FAMILIA DE COLUMNAS



- En este tipo de almacenamiento los datos se encuentran organizados por columnas, en lugar de por filas.
- Las bases de datos que utilizan esta forma de almacenamiento tienden a ser un híbrido entre las clásicas bases de datos relacionales y la tecnología orientada a columna.
- La columna es la base, es un elemento compuesto de un nombre, un valor y una marca de tiempo.

# BASES DE DATOS NOSQL – FAMILIA DE COLUMNAS



- El objetivo de una base de datos columnar es escribir y leer datos de manera eficiente, desde y hacia el almacenamiento en memoria secundaria, para acelerar el tiempo que se tarda en devolver el resultado de una consulta.
- La base de datos almacena sus datos (físicamente por familias de columna) de manera que pueden ser rápidamente agregados, con menos actividad de entrada y salida.
- Son especialmente eficaz en consultas que se necesitan leer pocos atributos, pero necesitamos ver todas las instancias de estos atributos

# BASES DE DATOS NOSQL – FAMILIA DE COLUMNAS - USOS



***cassandra***

- Registro de eventos.
- Sistemas de manejo de contenidos.
- Contadores.



GRACIAS