



PUCP

Diplomado de especialización de
desarrollo de aplicaciones con
Inteligencia Artificial

Almacenamiento de Datos No Estructurados

CURSO:

Análisis de sentimiento en redes
sociales.

PROFESOR :

MAG. ERASMO G. MONTOYA

Contenido

1. Introducción
2. Bases de Datos Relacionales
3. NoSQL (Not Only SQL)
4. Ejemplo: MongoDB

Introducción

Antes de BD -> Archivos

BASES DE DATOS (Databases)

Colecciones organizadas de datos

DBMS (Database Management System)

Sistemas que administran los datos, transacciones y otros problemas relacionados a la base de datos.

Bases de Datos Relacionales: Conceptos

Tablas (p.e. entidades)

Columnas (atributos)

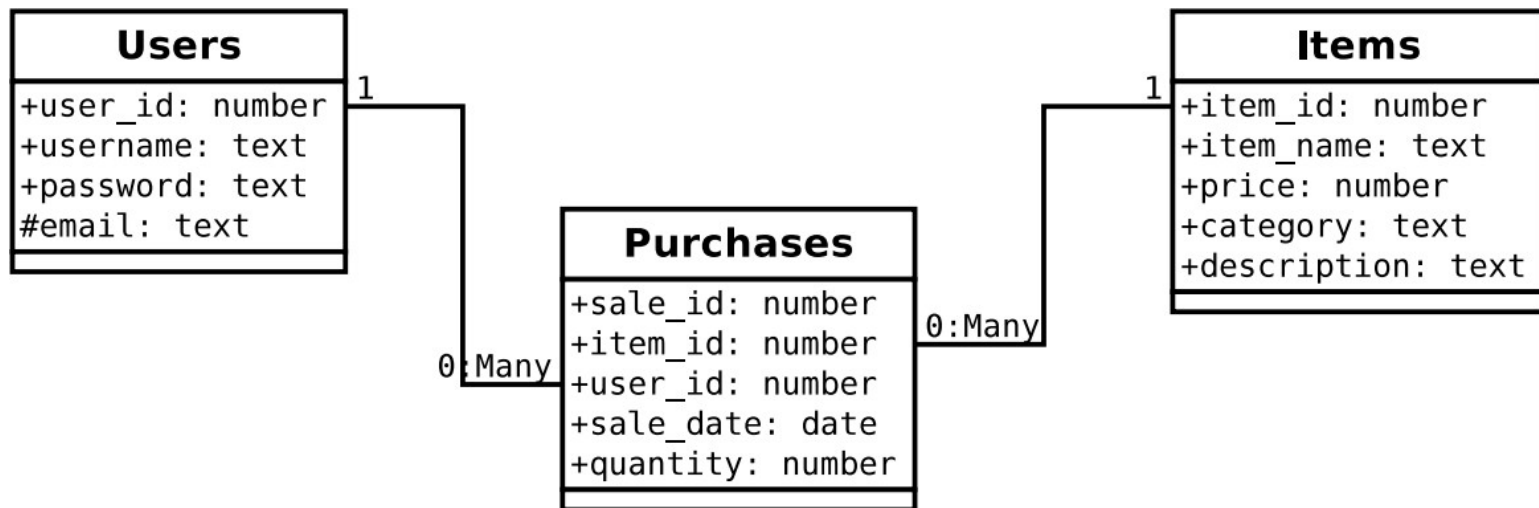
Filas (registro)

Relaciones (p.e. Foreign keys)

SQL (Structured Query Language)

Bases de Datos

Relacionales: Ejemplo



Bases de Datos Relacionales: Normalización

1. First Normal Form (1NF)

1. Second Normal Form (2NF)

1. Third Normal Form (3NF)

Bases de Datos Relaciones:

ACID

Propiedades para sus transacciones:

Atomicity

- Las transacciones se ejecutan todas en conjunto o ninguna

Consistency

- Se preserva la consistencia de datos en todas las transacciones.

Isolation

- Las transacciones se aíslan para que no afecte ni se vea afectada por otros movimientos.

Durability

- Los cambios realizados por una transacción se preservan (commit)

Bases de Datos Relaciones:

ACID

Principal **Ventaja**:

- Aplicación en sistemas bancarios y de transacciones críticas por su **seguridad**.

Principal **Desventaja**:

- La Atomicidad y el Aislamiento prácticamente fuerzan a realizar procesamientos **secuenciales**.

¿Qué alternativas tenemos?

*“Amazon, Facebook, and DARPA all recognized that when you **scale** systems large enough, you **can never put enough iron in one place** to get the job done (and you wouldn’t want to, to prevent a single point of failure).*

*Once you accept that you have a distributed system, you need to give up **consistency** or **availability**, which the fundamental transactionality of traditional RDBMSs cannot abide”*

Cedric Beust
[@cbeust](#)

NoSQL: Motivación

Los RBDMS no son suficientes para satisfacer todos los requerimientos (p.e. para manejar grandes cantidades de datos y de tipos no estructurados).

A partir de ello, se empezó a desarrollar alternativas de almacenamiento para satisfacer:

- Sistema **Distribuido**
- **Escalabilidad**
- Alta **Disponibilidad**
- Baja **Latencia**
- Bajo **Costo**

NoSQL: Motivación

Google -> BigTable

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., ... & Gruber, R. E. (2008). **Bigtable: A distributed storage system for structured data**. *ACM Transactions on Computer Systems (TOCS)*, 26(2), 4.

Amazon -> S3 Storage

<https://aws.amazon.com/es/s3/>

NoSQL: ¿Qué es?

Base de datos (naturalmente) **DISTRIBUIDAS** y por lo general no requieren **normalización**.

No considera las propiedades **ACID** para las transacciones

En su lugar, se propone **BASE**:

- **Basically Available** (disponibilidad sobre todo)
- **Soft State** (se pueden generar cambios en los estados sin necesidad de un query)
- **Eventually Consistent**(en algún momento el sistema tendrá consistencia en todos sus datos)

NoSQL: Eventual Consistency

En un modelo distribuido (cluster de servidores):



- Un servidor diferente puede responder un query
- Los servidores se comunican entre ellos con una programación determinada (“a su propio ritmo”)
- Un servidor sin el dato actualizado podría responder el query

NoSQL: Tipos

Columns Store

- Guardan la información por columnas (en vez de filas)
- Es útil para procesos de análisis de atributos específicos
- Ejemplos: Hbase, Cassandra

Documents Store

- Document  record
- Collection  table
- Ejemplos: MongoDB, CouchDB

Key-value Store

- Los datos son guardados con la lógica de un “hash” table

Graph Store

- Nodos (entidades) y aristas (relaciones) son los principales componentes

NoSQL vs SQL

SQL (Bueno)

Alto rendimiento para las transacciones (ACID)

Altamente estructurado y portable

Administra bien pocas cantidades de datos (~500GB)

Soporta tipos de datos variados en sus tablas

NoSQL vs SQL

SQL (Malo)

Queries complejos (JOINS) pueden tomar mucho tiempo

No es muy escalable naturalmente

El modelo relacional puede tener una curva de aprendizaje alta
(para el administrador de la BD)

Su implementación puede ser lenta y engorrosa

NoSQL vs SQL

NoSQL (Bueno)

Muy útil para datos volátiles o no relevantes (¿tweet?)

Alta disponibilidad para ejecuciones de lectura y escritura

En general es “más rápido” que SQL

Fácil de escalar

Su implementación puede resultar un poco más “sencilla”

NoSQL vs SQL

NoSQL (Malo)

La falta de relaciones puede afectar la conexión entre los datos

La seguridad y consistencia de la información

Concluyendo

SQL (RBDMS)

- Si tienes necesidad de ACID
- Si tus datos son muy importantes (finanzas)

NoSQL

- Si tienes problemas de disponibilidad
- Cuando es más importante tener datos rápidos que correctos
- Cuando tienes que escalar rápidamente tu almacenamiento

No hay una única opción, la decisión depende de sus requerimientos

Descanso 15 min

- Recomendación musical: [Willow Smith](#)

