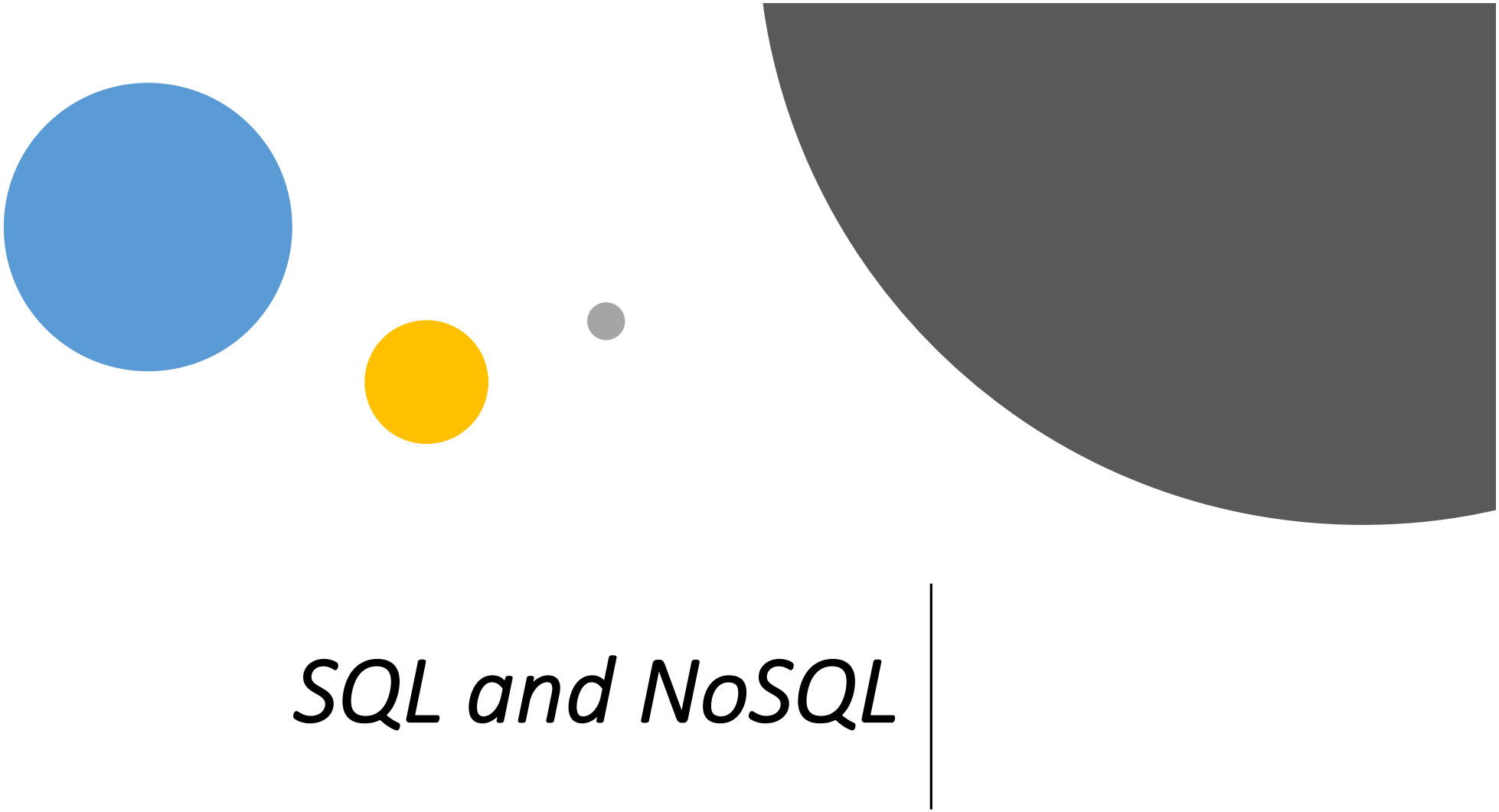
BIG DATA

Fuentes de Datos

MSc. Felipe Meza

Contenidos

- SQL / NoSQL



SQL and NoSQL

SQL / NoSQL

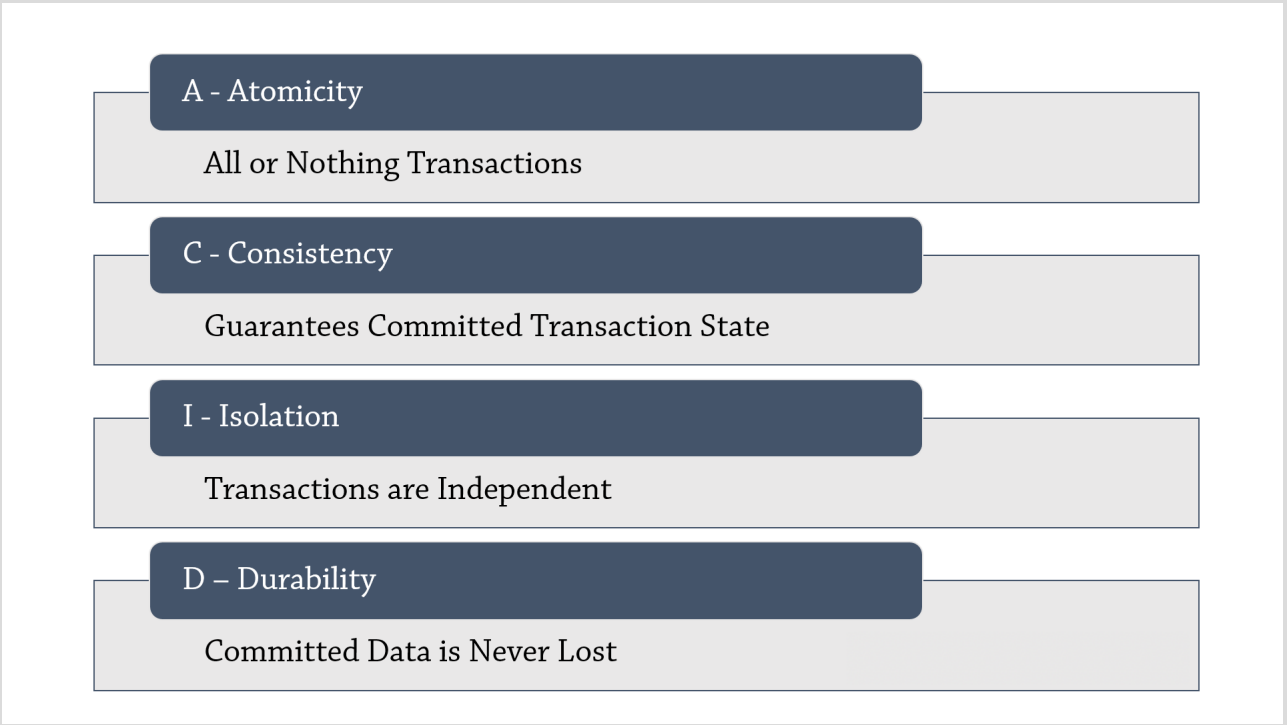
- SQL (Structured English as a Query Language, pronunciado "sequel") es un ***lenguaje de propósito específico*** creado para obtener (extraer) información estructurada en tablas, con ***schemas predeterminados*** y con ciertas ***reglas*** para unir diferentes tablas entre si.
- Por su parte NoSQL agrupa otros tipos de organización de datos con una estructura más flexible.

- Es necesario tener una **estructura definida** por **tipo de entidad**. Bajo condiciones normales entidades corresponden a **tablas**. Esto quiere decir que, a priori, ya conocemos todos los **atributos**. La calificación técnica en este sentido es que la Base de Datos es ***Relacional***.
- Orientado a tipos de ***datos básicos*** (numéricos y textuales). Conforme avanzan las bases de datos, eso si, van incorporando tipos de datos más complejos, lo que hace la diferencia más confusa. Ejemplos comunes de tipos de datos compuestos son **hileras *JSON*** y **listas de elementos**, bajo una sola columna. A esto se le llama ***normalización*** *de los datos*.

- **Normalización** de BD: Es el proceso de reestructurar una BD relacional con el fin de reducir los datos redundantes y mejorar la integridad de los datos.
- **Denormalización** de BD: Estrategia usada para tomar una base de datos normalizada, para extrae los datos de de una forma mas eficiente (en el menor tiempo posible).

- Existen **índices**, **llaves** y **restricciones** para resguardar la **integridad** de los datos.
- Existen **garantías** más generales que en NoSQL. La más común es *ACID: Atomicity, Consistency, Isolation and Durability*. También provee garantías **transaccionales**.
- SQL (IBM 70's), es un lenguaje **maduro** cuya infraestructura de generación y optimización de consultas es muy robusto.

SQL: Garantías ACID



SQL

itemid	orderid	item	amount
5	1	Chair	200.00
6	1	Table	200.00
7	1	Lamp	123.12

customerid	name	email
5	Rosalyn Rivera	rosalyn@adatum.com
6	Jayne Sargent	jayne@contoso.com
7	Dean Luong	dean@contoso.com

orderid	customerid	date	amount
1	4	11/1/17	523.12
2	3	11/15/17	32.99
3	1	11/21/17	23.99

NoSQL

- Se utiliza para referirse a bases de datos **no convencionales**, desde la concepción clásica SQL.
- Datos **no relacionales**.
- Datos que **crecen** muy aceleradamente.
- Situaciones en que SQL y BD relacionales son muy rígidas. Es posible que ***no*** se necesiten tantas garantías.
- Datos ***distribuídos***. Si bien es cierto las bases de datos relacionales no necesariamente son monolíticas, las consideraciones de máquinas que almacenan datos, réplicas y rendimiento son **más explícitas** en esquemas NoSQL.
- Poca ***normalización*** de los datos.
- En principio, no usa SQL para extraer los datos. ***Sin embargo***, existen tecnologías que abstraen la extracción de datos y permiten usar SQL.

Arquitecturas NoSQL

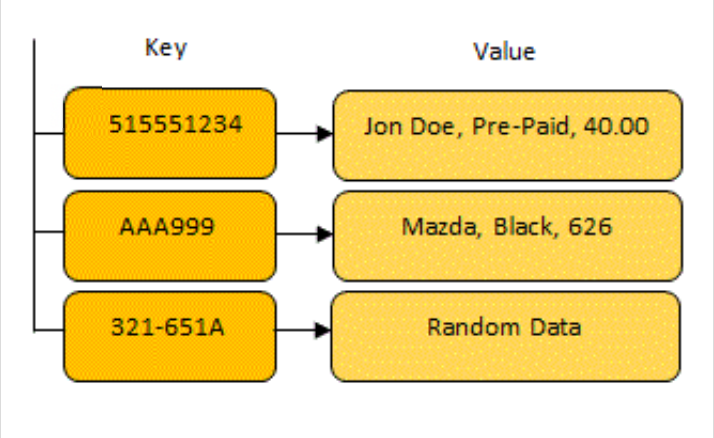
- Bajo las premisas anteriores se han desarrollado arquitecturas específicas. De las más comunes temenos:
  - **Diccionarios o tablas llave / valor:** La idea principal es que la llave del diccionario es un ***identificador directo a una entidad que debe ser accedida rápidamente*** y, por ende, es útil ***encapsular*** toda su información para su acceso posterior. Su uso predominante es en aplicaciones *CRUD* (Create, Read, Update, Delete). Esta idea de diccionarios se puede utilizar en múltiples niveles, teniendo bases de datos de primer orden que almacenan índices a otras bases de datos (a veces llamadas *Document Stores*).

ORACLE® NoSQL Database

redis

amazon DynamoDB

Arquitecturas NoSQL

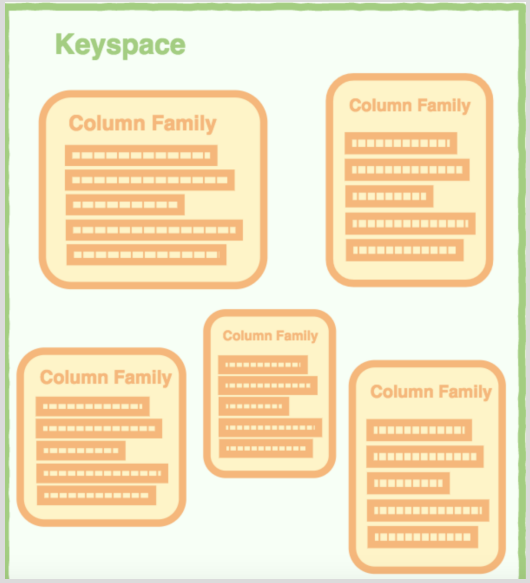


Diccionarios o tablas llave / valor

Arquitecturas NoSQL

- **Columnares:** *Bigtable* es un ejemplo. Los comentarios hechos en la clase anterior al respecto aplican. Un detalle adicional que cabe recalcar es que, contrario al nombre, la ***estructura puede ser por fila o por columna***. Esto quiere decir que cada arquitectura debe definir si su "primer nivel de almacenamiento" es una fila o columna, con sus respectivos pros y contras.

Arquitecturas NoSQL



Columnares

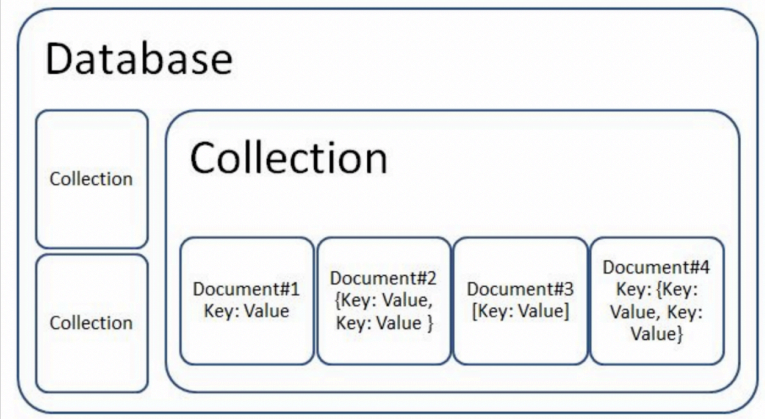
Se maneja el concepto de KEYSPACE, similar a SCHEMA, y contiene columnas por familias.

Ventajas en la compresión de los datos, alto desempeño en operaciones de agregación y escalabilidad.

La escritura puede ser algo lenta.

Arquitecturas NoSQL

- **Document-Based**: Los datos estan estructurados en la forma de documentos y colecciones, los documentos pueden ser PDF, WORD, XML, JSON file. Puede manejar datos estructurados, no-estructurados y semi-estructurados.



Couchbase

mongoDB

Arquitecturas NoSQL

- **Desventajas:** Estandarización, soporte y madurez (comparado con SQL). Herramientas analíticas son menos (comparado con SQL).

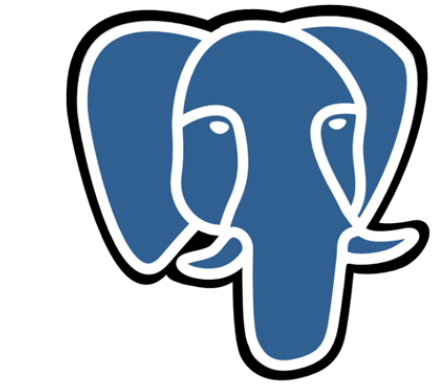
- **Ventajas:** Flexibilidad, alta escalabilidad a menor costo.

350 systems in ranking, December 2019

Rank			DBMS	Database Model	Score		
Dec 2019	Nov 2019	Dec 2018			Dec 2019	Nov 2019	Dec 2018
1.	1.	1.	Oracle	Relational, Multi-model	1346.39	+10.33	+63.17
2.	2.	2.	MySQL	Relational, Multi-model	1275.67	+9.38	+114.42
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1096.20	+14.29	+55.86
4.	4.	4.	PostgreSQL	Relational, Multi-model	503.37	+12.30	+42.74
5.	5.	5.	MongoDB	Document, Multi-model	421.12	+7.94	+42.50
6.	6.	6.	IBM Db2	Relational, Multi-model	171.35	-1.25	-9.40
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model	150.25	+1.85	+5.55
8.	8.	↓ 7.	Redis	Key-value, Multi-model	146.23	+1.00	-0.59
9.	9.	9.	Microsoft Access	Relational	129.47	-0.60	-10.04
10.	10.	↑ 11.	Cassandra	Wide column	120.71	-2.52	-1.10

https://db-engines.com/en/ranking

TAREA # 3



PostgreSQL