EJEMPLOS DE BASES NO SQL REMANENTE

ÍNDICE

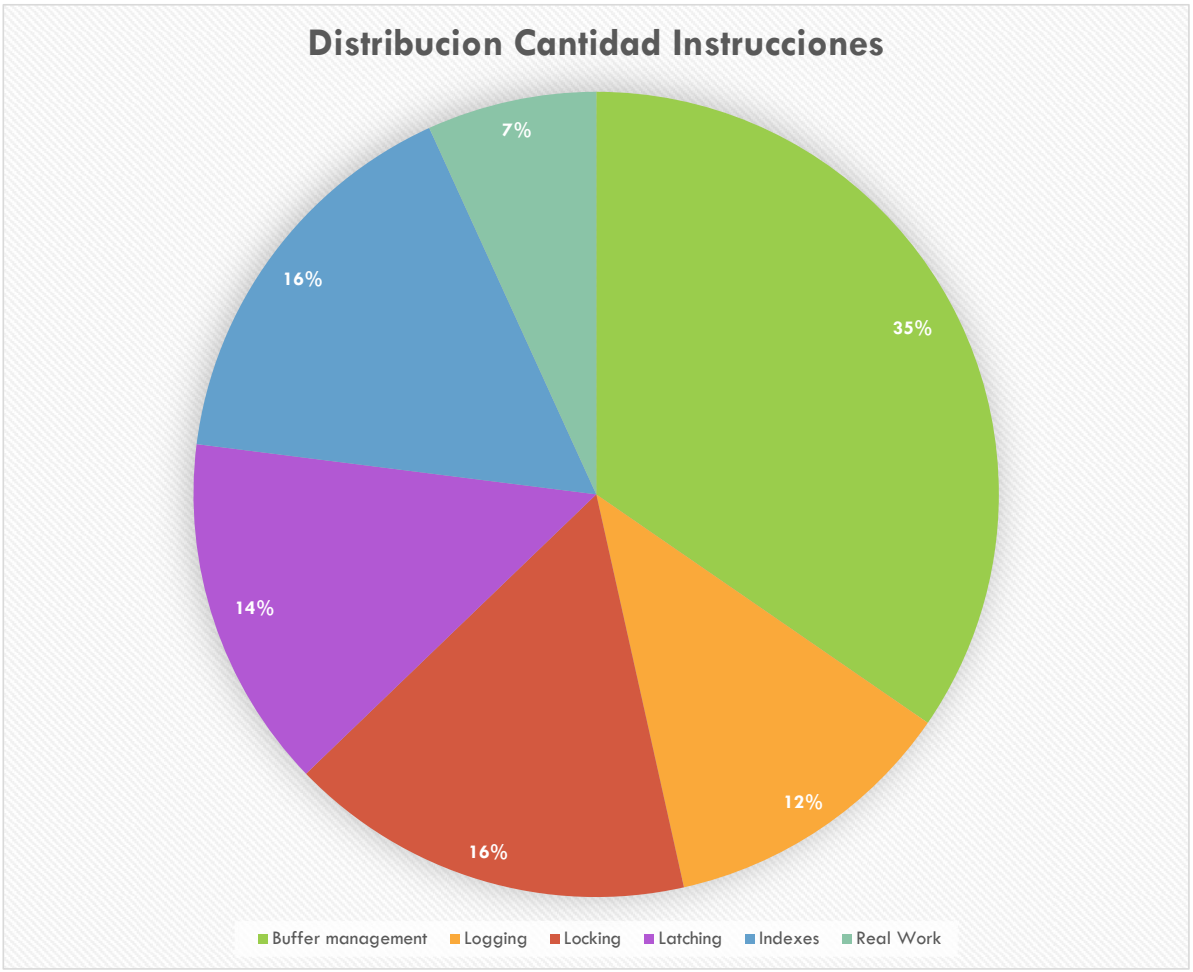
- In memory databases
- Cloud databases
- Bases de datos espaciales
- Stream database
- Vector Databases
- RDF
- Neo4j
- Multitenant

IN MEMORY DATABASE

- Hasta ahora asumimos que los datos que usa la base de datos están en disco, lo que implica que hay una comunicación entre el disco y la memoria
- El abaratamiento de la memoria dio lugar a empezar a pensar en tener todos los datos en memoria.
- En esta situación es más conveniente almacenar los datos de control de concurrencia junto con los datos que en estructuras separadas
- Es necesario mantener el log para permitir recuperar los datos si se produce un crash

¿¿QUIÉN CONSUME LOS RECURSOS??

- Fuente: OLTP through the looking glass, and what we found there



CONTROL DE CONCURRENCIA CON MÁS DE 100 CORES

- En el artículo “Staring into the abyss: an evaluation of concurrency control with one thousand cores” se muestran los problemas de los protocolos de control de concurrencia cuando existen más de 100 cores para bases de datos que están completamente en memoria.
- Utiliza diferentes pruebas TCP-P para evaluar la performance

EJEMPLOS DE INMEMORY DATABASE

- VOLT DB
  - Es una de las creaciones de Michael Stonebreaker
  - Es una base particionada
  - Single thread por CPU
  - Asegura la durabilidad por medio de command logging y las réplicas
  - Es ACID
  - Usa SQL
  - Los Stored Procedures se escriben en JAVA
  - En realidad, forma parte de las llamadas “NEW SQL DATABASES”

EJEMPLOS DE IN MEMORY DATABASE

- REDIS
  - Es una "**in memory database**", los cambios se guardan en disco en forma asincrónica
  - Tiene un servicio cliente y un servicio servidor, que lo hacen ideales para las instalaciones del tipo MASTER - SLAVE
  - Para manejar transacciones arma una cola con todos los comandos que la conforman y los ejecuta todos juntos.
  - Tiene un tipo de control de concurrencia que permite "enterarse" si se cambió un valor en el medio de la ejecución. El comando es WATCH
  - Maneja Streams y datos geoespaciales

ORACLE IN MEMORY DATABASE

- Tiene formato dual, guarda cada tabla en formato ROW y en formato COLUMN
- La tabla en formato ROW sigue estando en disco
- Los datos en formato columna tienen el ROWID que se refiere a la posición en disco de la tabla real.
- Los datos se guardan comprimidos
- Al resolver las consultas de analítica en memoria se necesitan menos índices.

CLOUD DATABASE

- CDBMS es una base de datos distribuida que provee computación como un servicio.
- Se paga por tiempo y espacio físico
- Los datos se guardan encriptados
- Hay bases de datos relacionales y no relacionales
- Puede contratarse
  - una máquina virtual o
  - una “base de datos virtual”

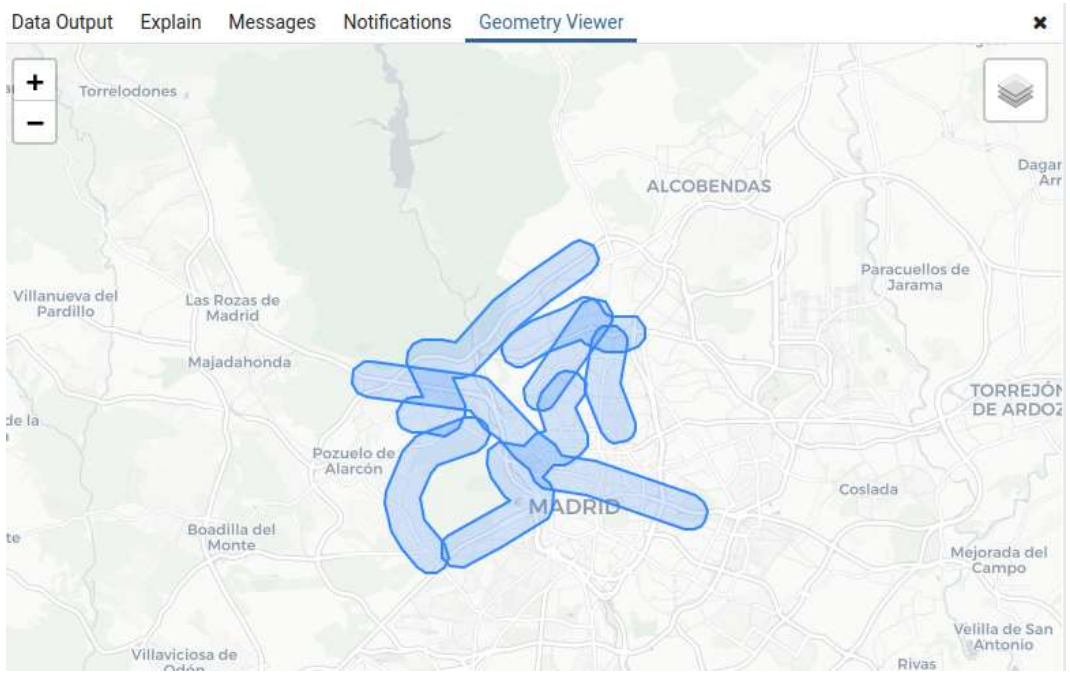
CLOUD DATABASES – VENTAJAS Y DESVENTAJAS

- Fácil de escalar
- Reduce el costo de tener un DBMS
- Mejora la tolerancia a fallas
- La seguridad es un tema en si misma
- Puede haber implicancias legales

EJEMPLOS

- Google Cloud Spanner
  - https://cloud.google.com/spanner?hl=es#section-6
- Amazon (AWS)
  - https://aws.amazon.com/es/
- Azure
  - https://azure.microsoft.com/es-es/
- Salesforce
  - Salesforce.com

¿QUÉ TIPOS DE DATOS CONOCEMOS?



BASE DE DATOS ESPACIAL

- Un sistema de base de datos espacial es un sistema de base de datos.
- Cuenta con los tipos de datos espaciales (SDT) en su modelo de datos y lenguaje de consulta.
- Es compatible con los tipos de datos espaciales en su aplicación, proporcionando al menos la indexación espacial y algoritmos eficientes para la unión espacial.
- Las bases de datos geográficas están incluidas en este grupo

Fuente: https://www.geomapik.com/analisis-gis/postgis-analisis-espacial-funciones/

BASES DE DATOS ESPACIALES

- Tienen que poder sacar ventaja de los datos de "ubicación"
- Los queries que resuelven incluyen los siguientes tipos
  - Objetos de un cierto tipo en un cierto rango
  - Vecino mas cercano de un cierto tipo de objeto
  - Intersección o superposición entre 2 "objetos"

BASES DE DATOS ESPACIALES

- POSTGIS
  - Incorpora los tipos: punto, línea, polígono, y también colecciones de los mismos.
  - Los datos se pueden importar desde varios formatos standard
  - Provee funciones para calcular la intersección de dos figuras geometricas, unión , etc.
  - También permite calcular las distancias
  - Integra esas funciones en el SQL, inclusive pueden usarse como parte de un join
  - Incluye funciones de búsqueda de objeto mas cercano.

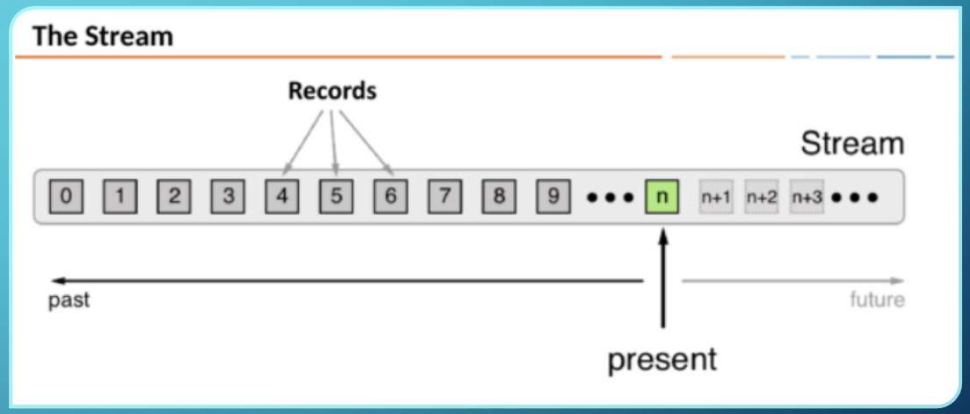
POSTGIS – EJEMPLO

- **“Approximately how many people live on (within 50 meters of) Atlantic Commons?”**
- `SELECT Sum(popn_total)`
- `  FROM nyc_census_blocks`
- `  WHERE ST_DWithin(  geom,`
- `   ST_GeomFromText('LINESTRING(586782 4504202,586864`
- `4504216)', 26918),   50  );`

Fuente : https://postgis.net/workshops/postgis-intro/spatial_relationships_exercises.html

STREAM DATABASES

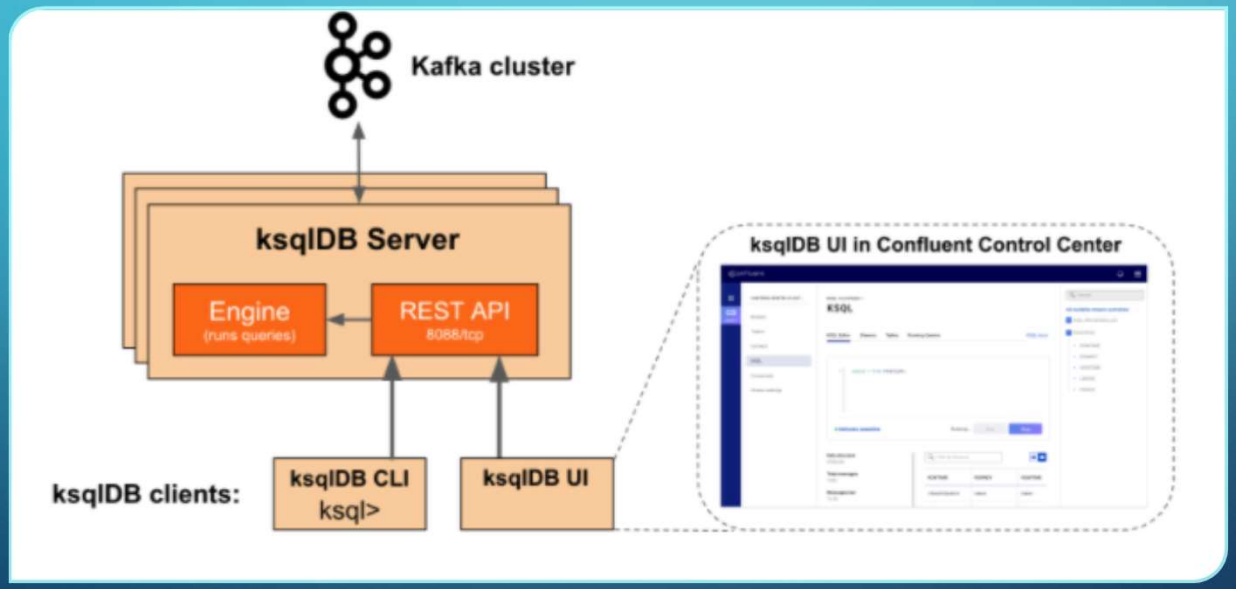
- los datos llegan en forma continua, por ejemplo…????
- Los datos están fuertemente asociados a un periodo de tiempo
- Son válidos en una ventana.
- Un stream, es una bolsa de pares <s, t> donde s es una tupla y t es el timestamp que denota la llegada lógica de la tupla al stream.



https://docs.ksqldb.io/en/latest/concepts/time-and-windows-in-ksqldb-queries/

KSQLDB

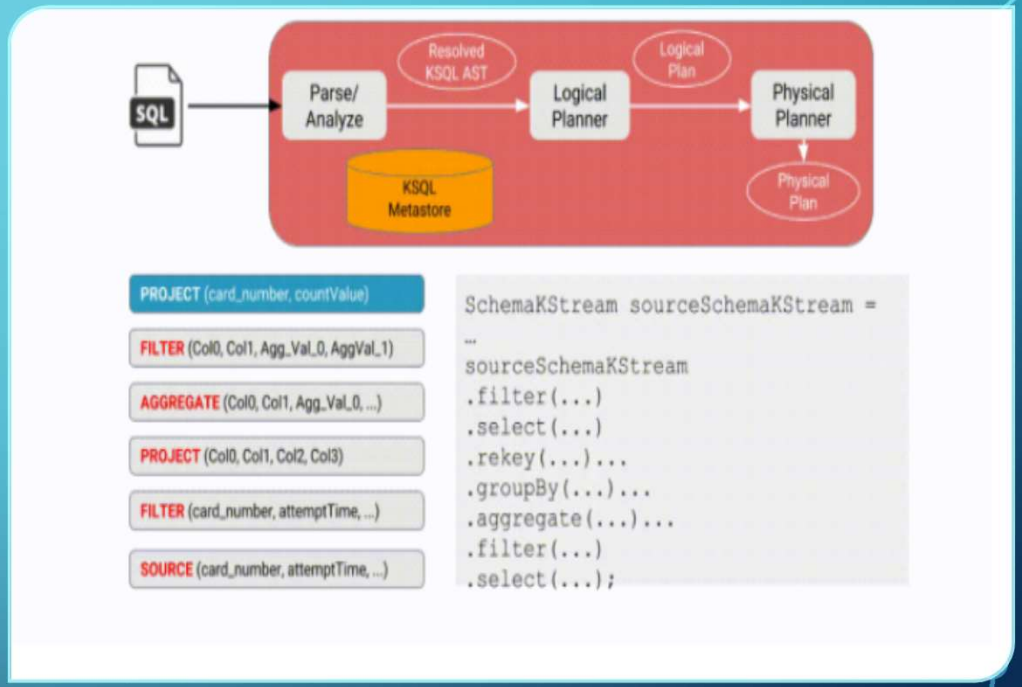
- Es una base de datos construida sobre Kafka Streams
- Sus principales componentes son



https://docs.ksqldb.io/en/latest/operate-and-deploy/how-it-works/

KSQLDB

- Tiene instrucciones de DDL
- **CREATE STREAM** readings (sensor VARCHAR **KEY**, location VARCHAR, reading INT) **WITH** (**kafka_topic** = 'readings', **partitions** = 3, **value_format** = 'json');
- Estas instrucciones modifican la metadata que cada KSQLDB tiene y transforman las instrucciones DML en aplicaciones de KAFKA Streams (plan físico)
- Pueden crearse streams o tablas a partir de otros objetos ya creados.

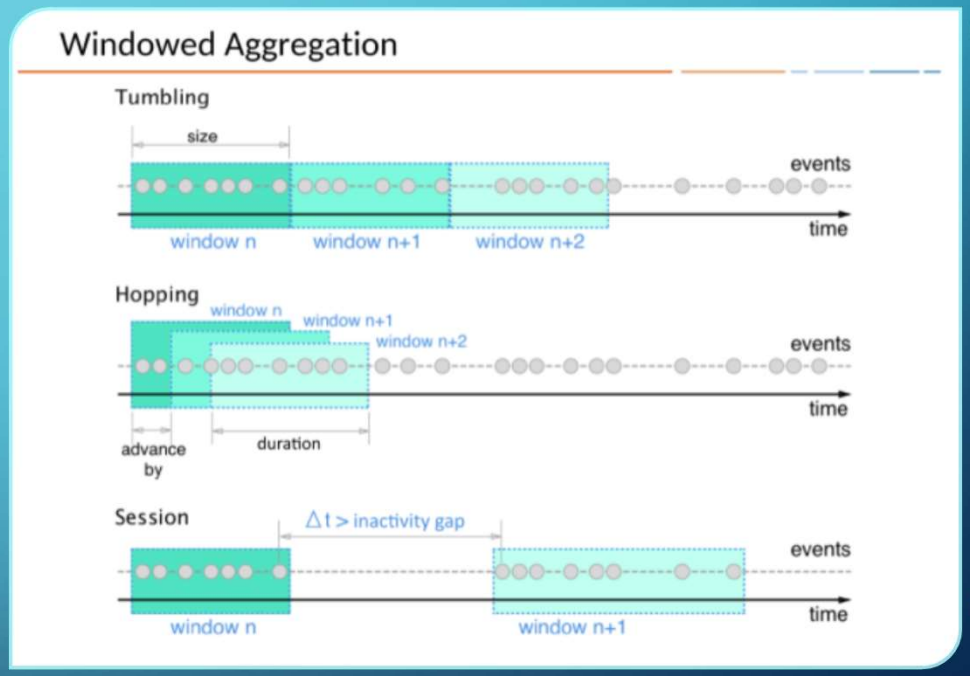


KSQLDB

- Tiene instrucciones DML
- **SELECT** sensor,
  reading,
  UCASE(location) **AS** location
  FROM readings
- Permite combinar con datos que están en otras bases de datos mediante el uso de conectores JDBC

STREAM DATABASES

- Las consultas deben contemplar la “validez” de los datos.
- Aparecen conceptos vinculados a las ventanas de tiempo

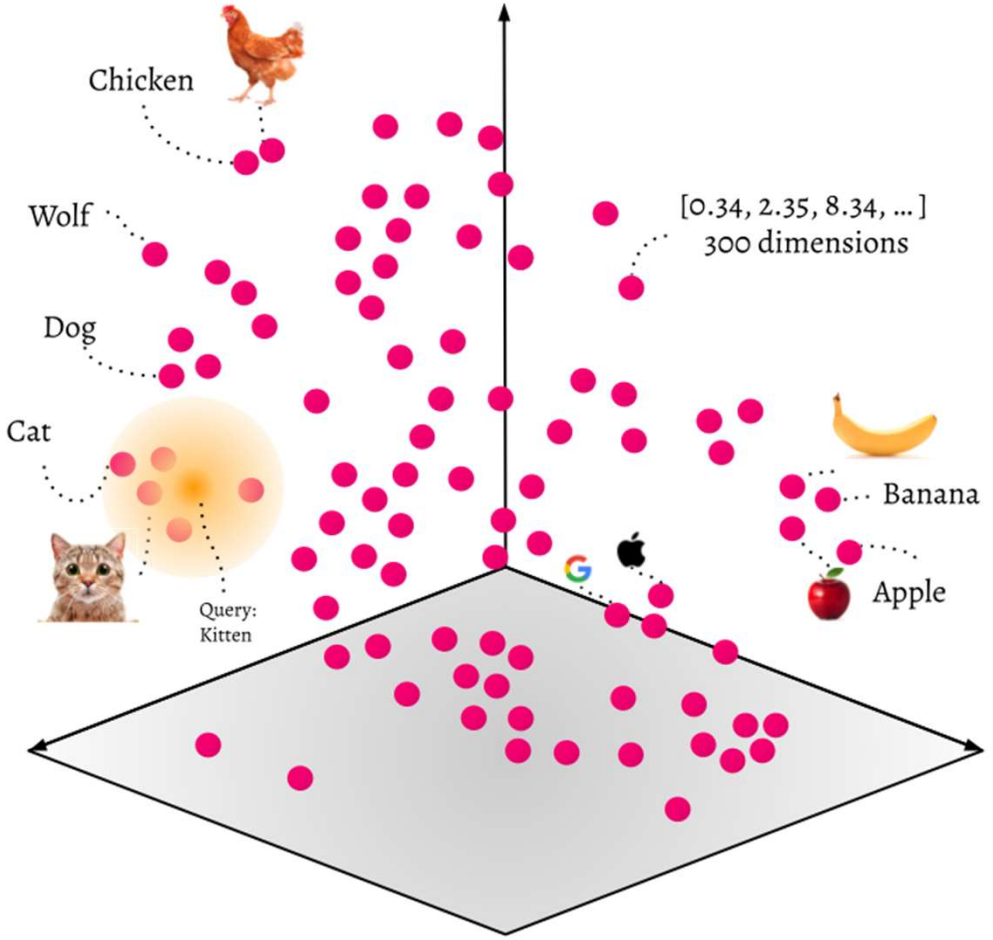


EJEMPLO

```
CREATE TABLE pageviews_per_region AS
SELECT regionid, COUNT(*)
FROM pageviews
WINDOW HOPPING (SIZE 30 SECONDS, ADVANCE BY 10
SECONDS, RETENTION 7 DAYS, GRACE PERIOD 30
MINUTES)
WHERE UCASE(gender)='FEMALE' AND LCASE
(regionid) LIKE '%_6' GROUP BY regionid EMIT
CHANGES;
```

VECTOR DATABASE

- Son bases de datos especialmente diseñadas para almacenar y efectuar búsquedas sobre información representada como vectores
- En espacios de gran dimensionalidad los puntos que se encuentran “cerca” corresponden a información similar
- Los resultados son aproximados
- .



Fuente : https://gustavo-espindola.medium.com/mejorando-las-respuestas-de-los-modelos-de-lenguaje-con-bases-de-datos-vectoriales-b746223f05b7

EJEMPLO

- SELECT…
- FROM house_for_sale
- WHERE price <= (SELECT budget FROM customer …)
- AND city in (SELECT search_city FROM customer …)
- ORDER BY vector_distance(house_vectors, :input_vector);

RDF MANAGEMENT SYSTEMS

- Cada vez mas fuentes de datos se exportan en format RDF (Resource Description Format), estandarizado por la W3C
- Los datos se representan como triples.
- Un dataset de tipo RDF tiene 2 tipos de datos
  - Explícitamente declarados
  - Implícitos, originados en restricciones semánticas. Estos datos e obtienen por medio de un proceso llamado “entailment”, que consiste en “extender” los datos explícitos con las restricciones semánticas para convertirlos en explícitos. Esta extensión se produce aplicando las “reglas” de entailment.
- Un ejemplo famoso es
  - Sócrates es humano
  - Los humanos son mortales
  - Y el entailment concluye que Sócrates es mortal.

RDF DATAMANAGEMENT

- Otro Ejemplo (Movies).
  - The fact that Stanley Kubrick has directed Jack Nicholson in the movie The Shining can be represented by the following triples
  - doi0 hasName Stanley Kubrick;
  - doi0 hasDirected doi1i;
  - Doi1 is a Movie;
  - doi1 hasName The Shining;
  - doi2 hasStarred doi;
  - doi2 hasName Jack Nicholson

RDF MANAGEMENT

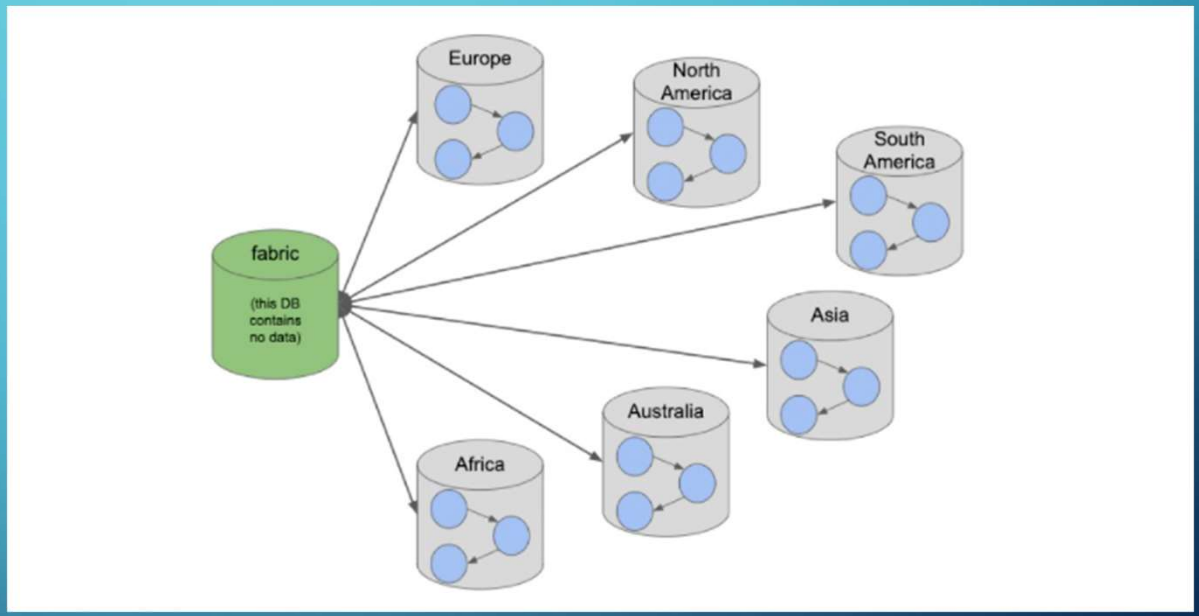
- Un ejemplo de query (expresado en BGP (Basic Graph Pattern)) puede ser
- q1(x) :- x hasStarred y; (0)
- w hasDirected y; (1)
- w hasNationality French (2)
- Esto se traduce como “quienes protagonizaron películas dirigidas por un francés”

RDF MANAGEMENT

- Dada una consulta existen 2 formas de resolverla en una base de datos relacional
  - Haber convertido todos los datos implícitos en explícitos (saturación)
  - Reescribir el query de tal forma que contenga las “expansiones”. Existen reglas para efectuar esta reformulación
- La aproximación que más se usa ahora es la de representar este tipo de datos en bases de datos de grafos y las consultas también y resolverlas ahí
- SPARQL es un lenguaje standard para esto

NEO4J - SHARDING

- Permite hacer sharding
- Tiene una base de datos virtual que actúa como punto de entrada



NEO4J - REPLICAS

- Tienen dos roles
  - Core
  - Read replicas
- Confirma las transacciones cuando la mayoría de los “core servers” de un cluster lo hicieron

MULTI TENANT

- Es un concepto que se usa para indicar que muchos usuarios comparten una misma aplicación
- Si uno quiere desarrollar una aplicación de estas características el principal problema es el aislamiento de los datos
- Esto puede hacerse de 3 formas
  - Una base para cada cliente
  - Una única base de datos , pero con un schema para cada cliente
  - Un único schema para todos los clientes

COMPARACIÓN

	1 base de datos por cliente	1 schema por cliente	Una única base y un único esquema
Facilidad de Instalación	+	++	+++
Posibilidad de Customización	+++	++	+
Aprovechamiento de recursos	+	++	+++
Garantía de seguridad	+++	++	+
Facilidad de recuperación frente a fallas	+++	++	+

REFERENCIAS

- VECTOR DATABASES :
  - https://milvus.io/
  - https://www.trychroma.com/
- RDF
  - Damián Bursztyn **Optimización de consultas RDF reformuladas.** Tesis de Licenciatura, Univ. de Buenos Aires.
  - POSTGIS: https://postgis.net/
  - SPARQL: https://www.w3.org/TR/rdf-sparql-query/
- Database of databases
  - Dbdb.io

REFERENCIAS

- KSQLDB :
  - https://ksqldb.io/
  - https://www.confluent.io/blog/how-real-time-stream-processing-works-with-ksqldb/?_ga=2.121371919.625545257.1621019927-1295274408.1621019927
- NEO4j
  - https://neo4j.com/
- CLOUD DATABASES
  - Shende, S.B. and Chapke, P.P. 2015. Cloud Database Management System (CDBMS). *COMPUSOFT: An International Journal of Advanced Computer Technology*. 4, 1 (Oct. 2015).
- Database of databases
  - Dbdb.io

REFERENCIAS

- INMEMORYDATABASES :
  - Stavros Harizopoulos, Daniel J. Abadi, Samuel Madden, and Michael Stonebraker. 2008. OLTP through the looking glass, and what we found there. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (SIGMOD '08). Association for Computing Machinery, New York, NY, USA, 981–992. DOI:https://doi.org/10.1145/1376616.1376713
  - Xiangyao Yu, George Bezerra, Andrew Pavlo, Srinivas Devadas, and Michael Stonebraker. 2014. Staring into the abyss: an evaluation of concurrency control with one thousand cores. Proc. VLDB Endow. 8, 3 (November 2014), 209–220. DOI:https://doi.org/10.14778/2735508.2735511
  - https://www.oracle.com/a/tech/docs/database-in-memory-ds-19c.pdf
- GENERALES
  - https://hostingdata.co.uk/nosql-database/
  - https://nosql-database.org/select-the-right-database.html