

NoSQL

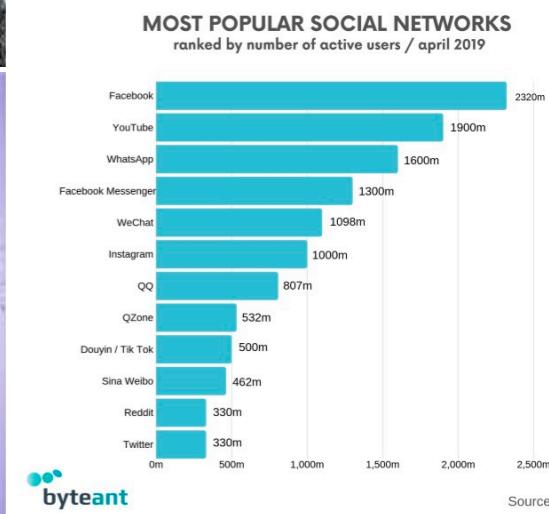
Alex Di Genova

07/06/2022

NoSQL

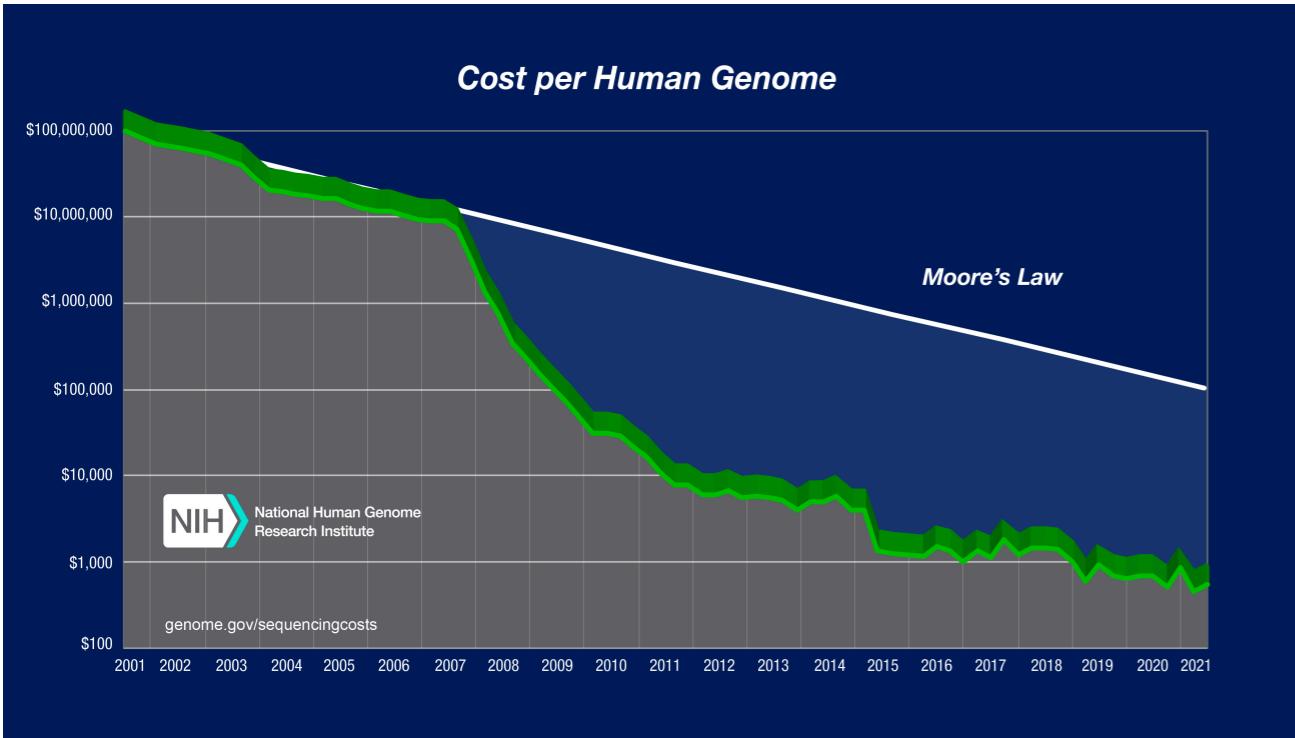
NoSQL

- El término NoSQL se utilizó por primera vez en 1998 para una base de datos que (aunque relacional) no tenía una interfaz SQL.
- Más importante durante la década de 2000, especialmente con la rápida expansión de Internet (web-scale databases).
- Los componentes de procesamiento orientados a la consistencia a menudo dificultan el procesamiento eficiente de grandes cantidades de datos (big data).



NoSQL

Ejemplo genómica



Ultima Genomics



Alex Di Genova @digenoma · May 30

\$1/Gb go Ultima genomics!!!! reads of ~400bp!!!

...

bioRxiv @biorxivpreprint · May 30

Cost-efficient whole genome-sequencing using novel mostly natural sequencing-by-synthesis chemistry and open fluidics platform
biorxiv.org/cgi/content/sh... #bioRxiv

1

5

29

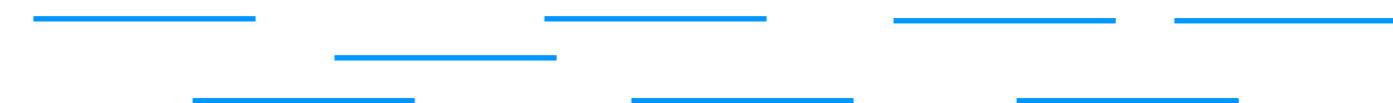
↑

...

Genome

Sequencing technology

Reads



Assembler

Overlap

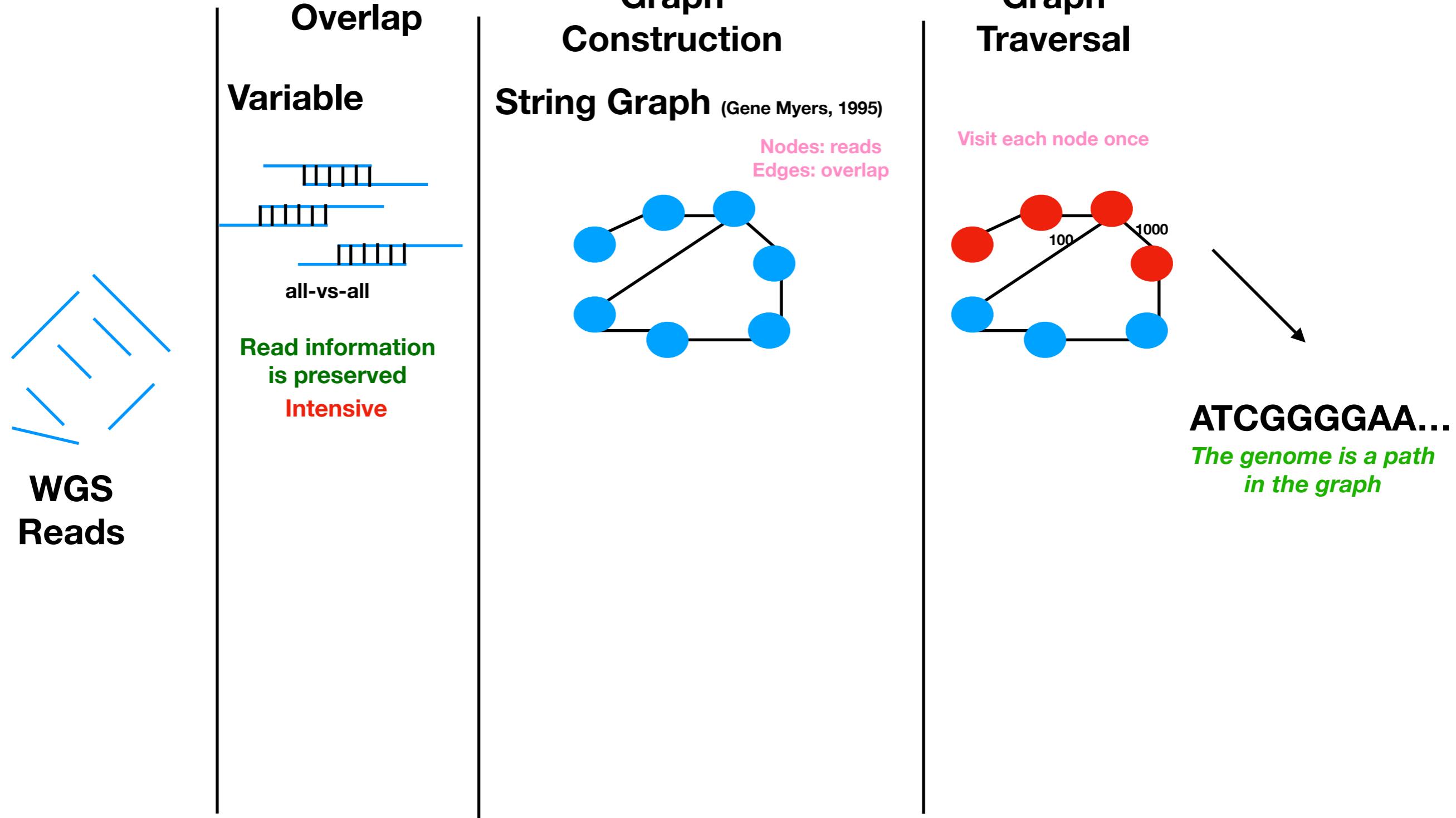


Contigs



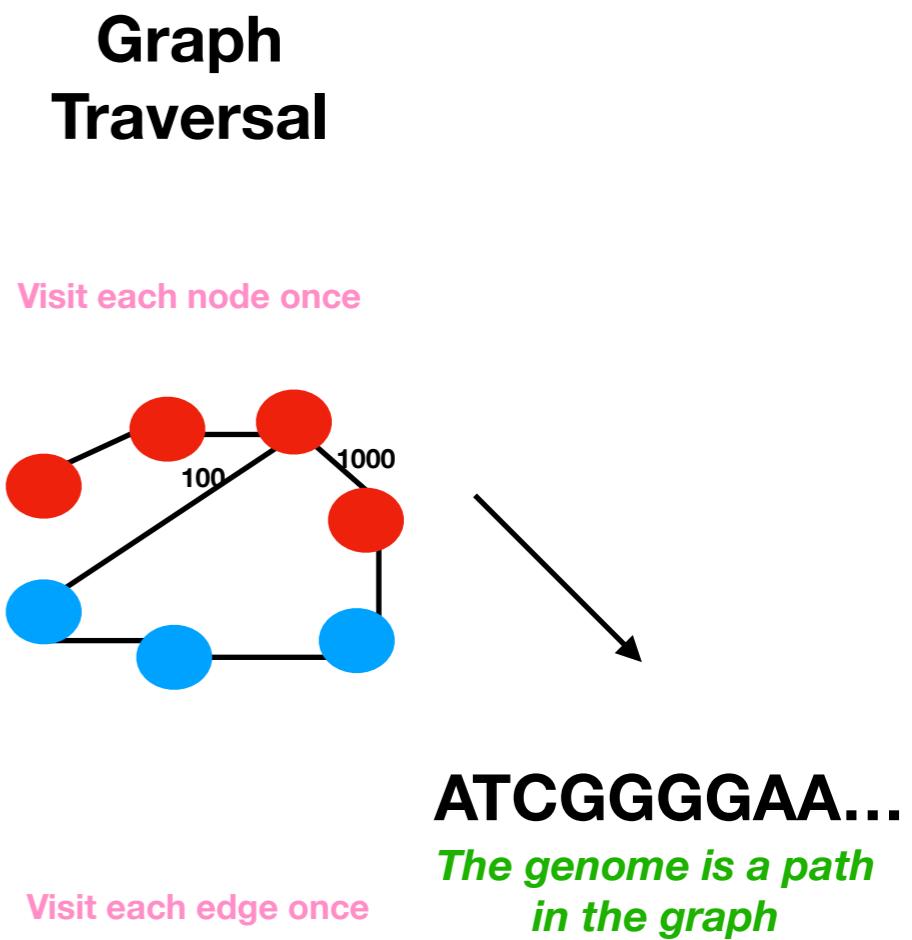
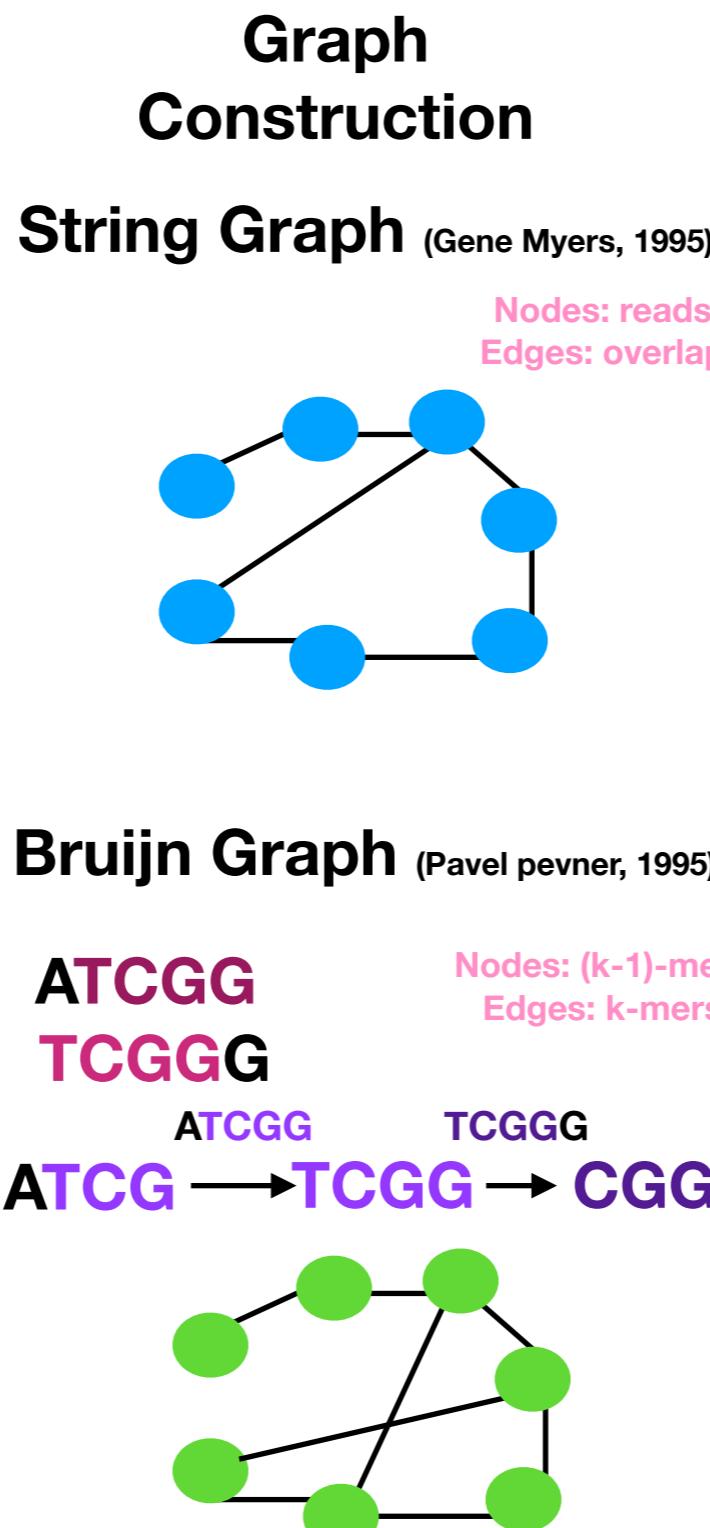
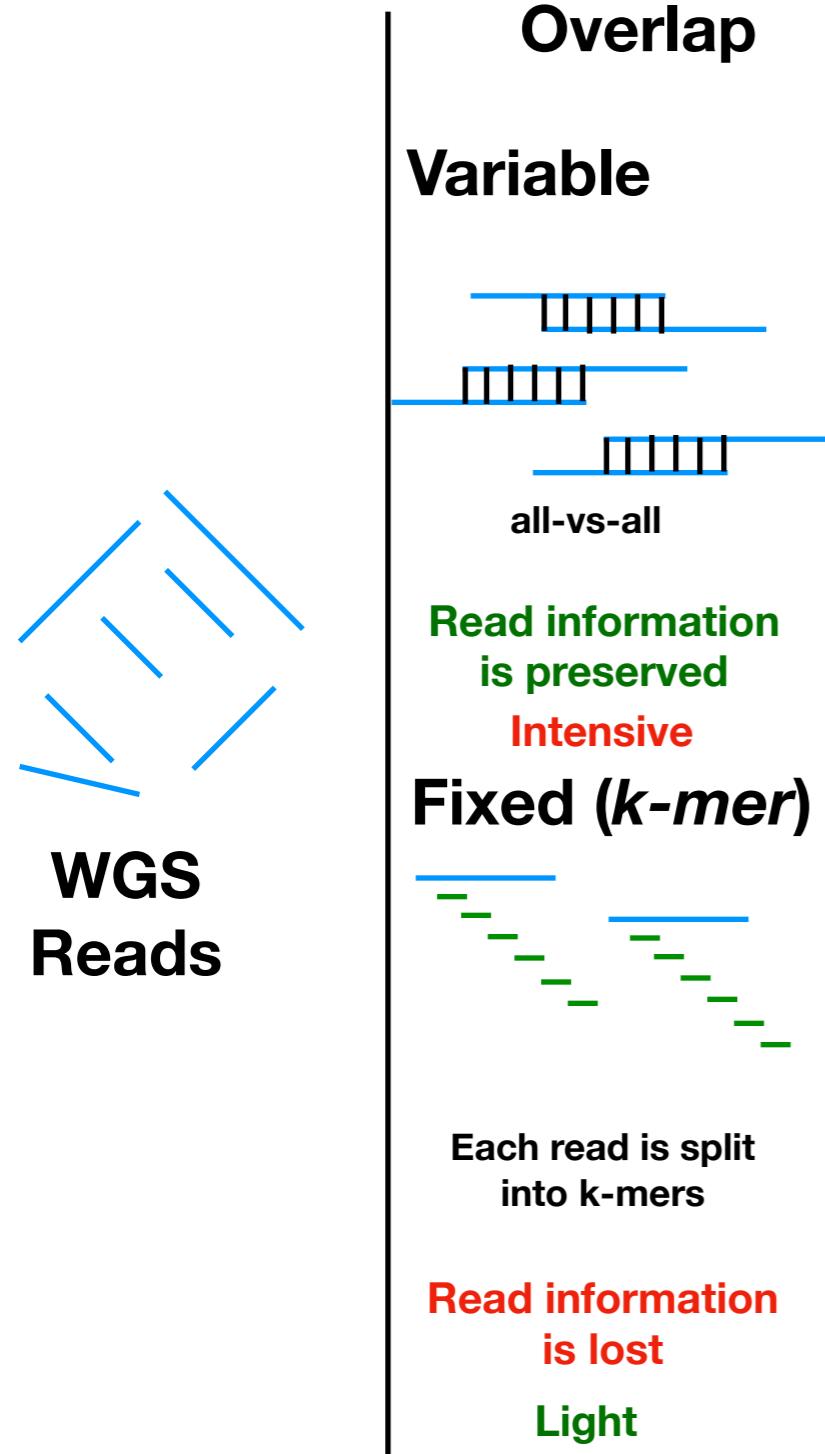
NoSQL

Ejemplo genómica



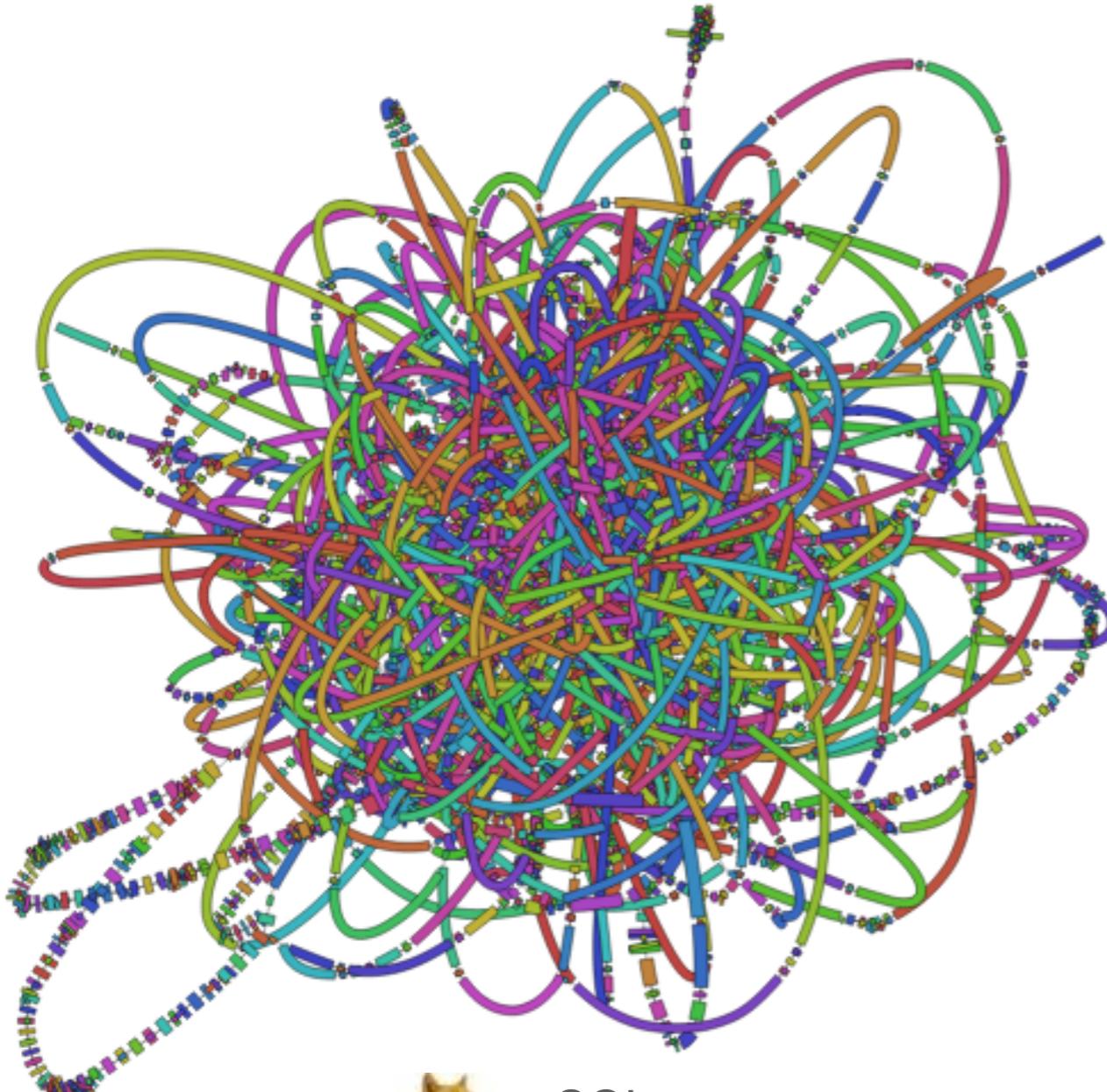
NoSQL

Ejemplo genómica



NoSQL

Ejemplo genómica



Hospitales
Ventas
Organizaciones



- Genómica
- Astronomía
- Web
- Redes sociales

- Genoma Humano:
- 3Gb
- 180Gb por persona
- Grafo
- >3000,000,000 nodos.
- >3000,000,000 conexiones.,,

ARTICLES

<https://doi.org/10.1038/s41587-020-00747-w>

nature
biotechnology

Check for updates

OPEN

Efficient hybrid de novo assembly of human genomes with WENGAN

Alex Di Genova^{1,2}, Elena Buena-Atienza^{3,4}, Stephan Ossowski^{3,4} and Marie-France Sagot^{1,2}

Generating accurate genome assemblies of large, repeat-rich human genomes has proved difficult using only long, error-prone reads, and most human genomes assembled from long reads add accurate short reads to polish the consensus sequence. Here we report an algorithm for hybrid assembly, WENGAN, that provides very high quality at low computational cost. We demonstrate de novo assembly of four human genomes using a combination of sequencing data generated on ONT PromethION, PacBio Sequel, Illumina and MGI technology. WENGAN implements efficient algorithms to improve assembly contiguity as well as consensus quality. The resulting genome assemblies have high contiguity (contig NG50: 17.24–80.64 Mb), few assembly errors (contig NGA50: 11.8–59.59 Mb), good consensus quality (QV: 27.84–42.88) and high gene completeness (BUSCO complete: 94.6–95.2%), while consuming low computational resources (CPU hours: 187–1,200). In particular, the WENGAN assembly of the haploid CHM13 sample achieved a contig NG50 of 80.64 Mb (NGA50: 59.59 Mb), which surpasses the contiguity of the current human reference genome (GRCh38 contig NG50: 57.88 Mb).

NoSQL

Actualmente

- El término ha adquirido diferentes significados.
- Una interpretación común es "no solo SQL"
- La mayoría de los sistemas NoSQL modernos difieren del modelo relacional o la funcionalidad RDBMS estándar:

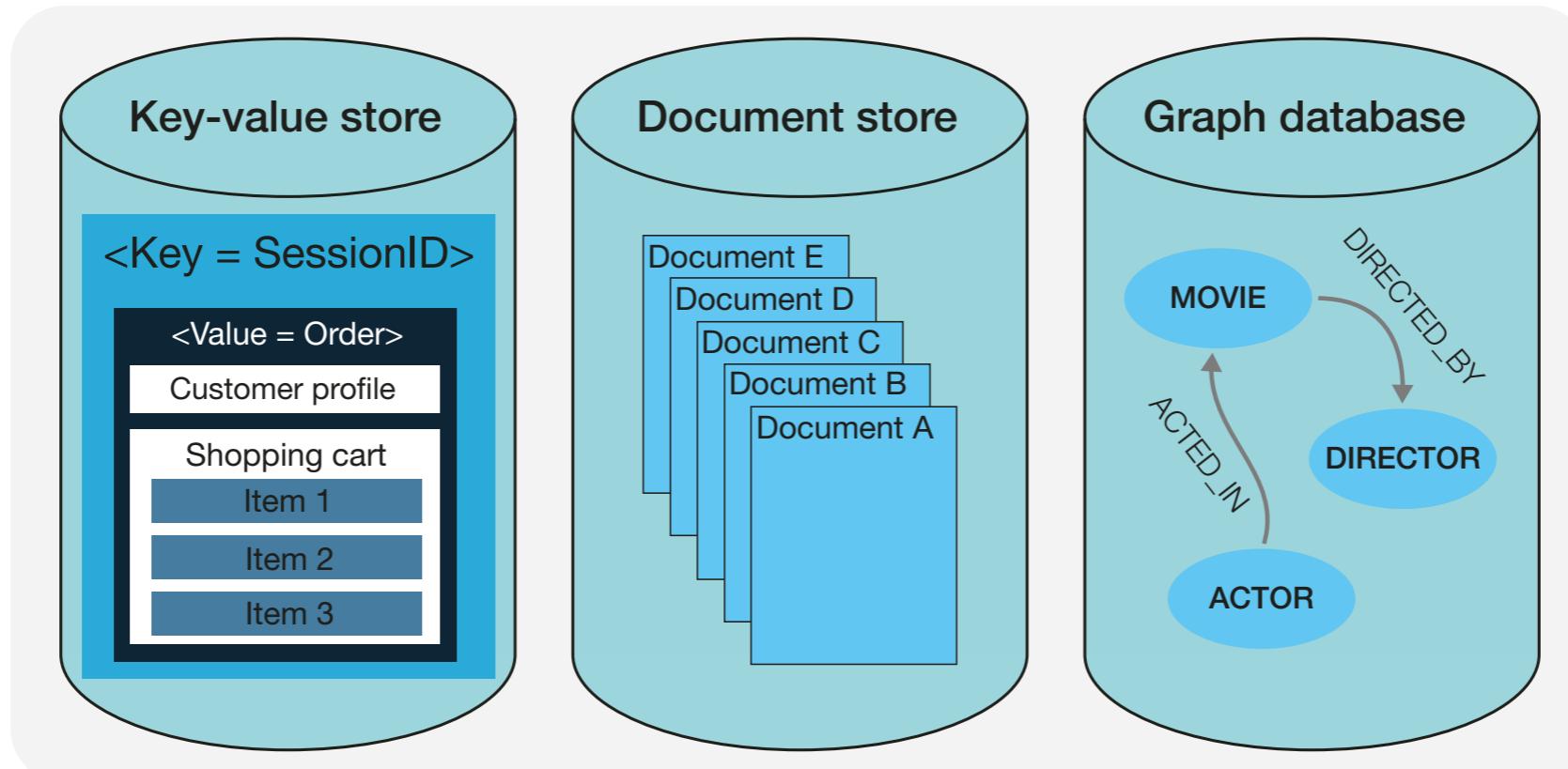
	SQL	NoSQL
Modelo de datos	Relaciones Tuplas Atributos Dominios Normalización	Documentos Grafos clave/valor
Modelo de consultas	SQL Algebra relacional	Recorridos en grafos Busqueda en texto Map/reduce
Implementación	Esquemas rígidos Propiedades ACID	Esquemas flexibles BASE

NoSQL hoy en día es más comúnmente destinado a ser algo así como "no relacional".

NoSQL

Categorías primarias de NoSQL

- Categorías generales de sistemas NoSQL
 - Almacenamiento basado en Clave/valor
 - Almacenamiento basado en documentos
 - Almacenamiento basado en grafos

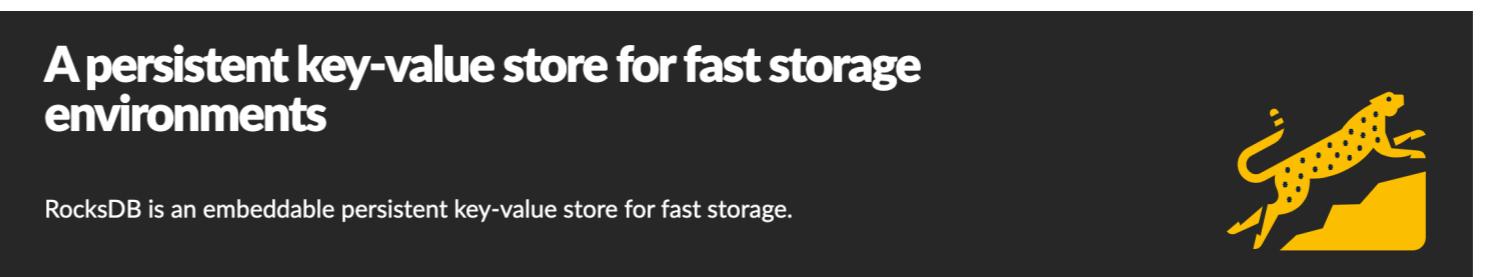


DynamoDB
Azure Table Storage
Riak
Redis
Aerospike
FoundationDB
LevelDB
Berkeley DB
Oracle NoSQL Database
GenieDb
BangDB
Chordless
Scalaris
Tokyo Cabinet/Tyrant
Scalien
Voldemort
Dynomite
KAI
MemcacheDB
Faircom C-Tree
LSM
KitaroDB
HamsterDB
STSdb
TarantoolBox
Maxtable
Quasardb
Pincaster
RaptorDB
TIBCO Active Spaces
Allegro-C
nessDB
HyperDex
SharedHashFile
Symas LMDB
Sophia
PickleDB
Mnesia
LightCloud
Hibari
OpenLDAP
Genomu
BinaryRage
Elliptics
Dbreeze
RocksDB
TreodeDB
www.nosql-database.org
www.db-engines.com
www.wikipedia.com

NoSQL

Almacenamiento basado en Clave/valor

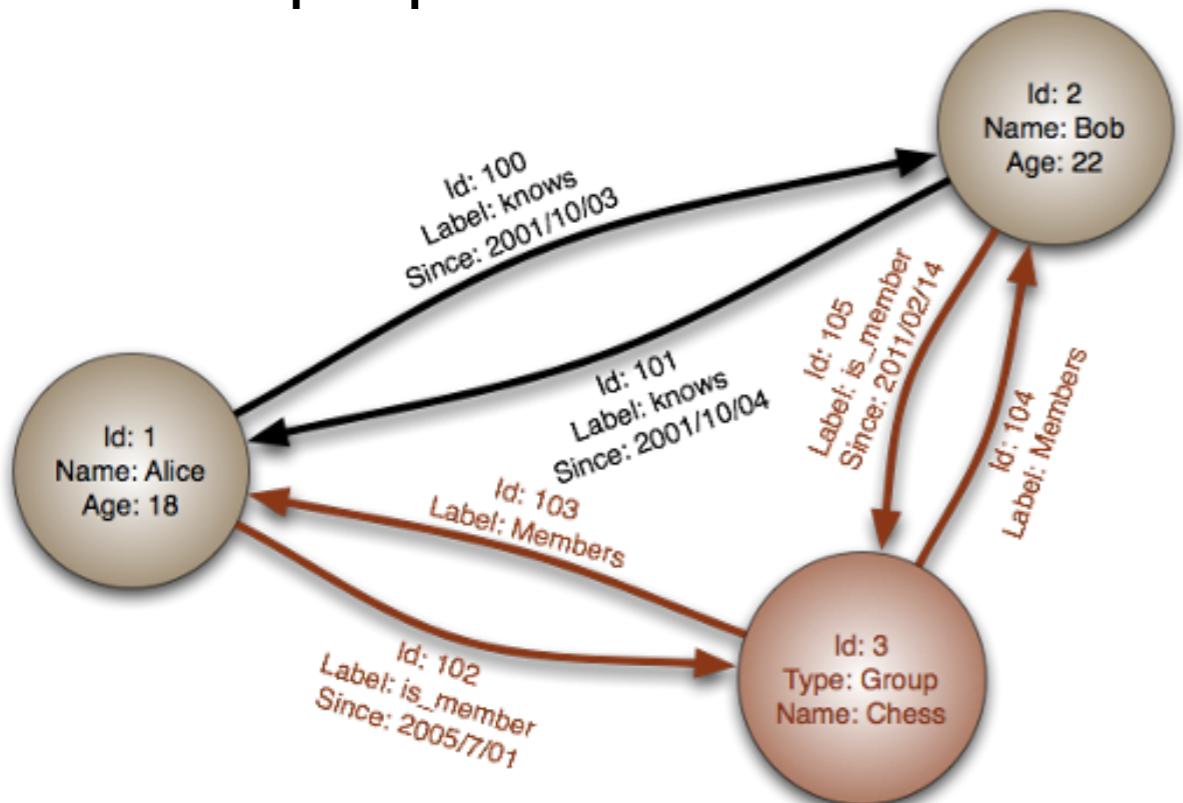
- El modelo básico de datos:
 - La base de datos es una colección de pares clave/valor
 - La clave para cada par es única.
- Operaciones primarias
 - `insert(key, value)`
 - `delete(key, value)`
 - `update(key, value)`
 - `lookup(key)`



NoSQL

Almacenamiento basado en grafos

- El modelo básico de datos:
 - Grafos dirigidos
 - Nodos y arcos con propiedades



NEO4J GRAPH DATA PLATFORM

Blazing-Fast Graph, Petabyte Scale

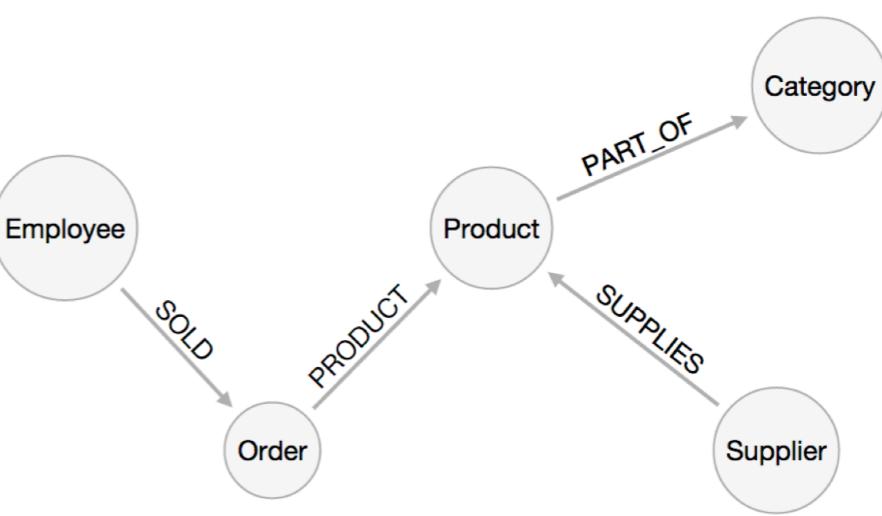
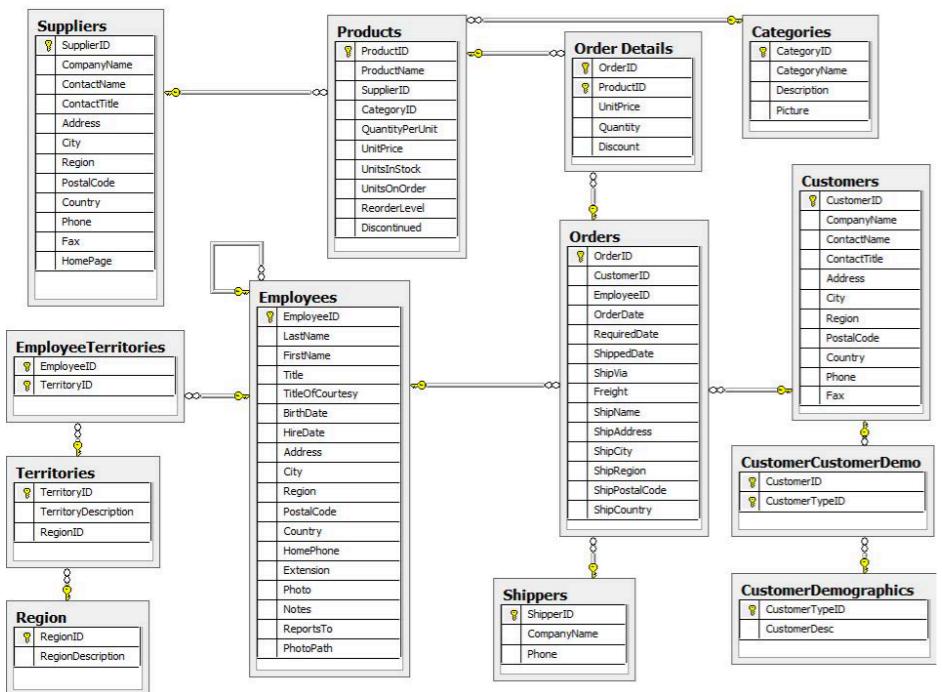
With proven [trillion+ entity performance](#), developers, data scientists, and enterprises rely on Neo4j as the top choice for high-performance, scalable analytics, intelligent app development, and advanced AI/ML pipelines.

AllegroGraph
ArangoDB
Bigdata
Betsy
BrightstarDB
DEX/Sparksee
Execom IOG
Fallen *
Filament
FlockDB
GraphBase
Graphd
Horton
HyperGraphDB
Item G Native Store
InfiniteGraph
InfoGrid
jCoreDB Graph
MapGraph
Meronymy
Neo4j
Orly
OpenLink virtuoso
Spatial and Graph
Simple NoSQL Database
OrientDB
OQGraph
Ontotext OWLIM
R2DF
ROIS
Sones GraphDB
SPARQLCity
Sqrrl Enterprise
Stardog
Teradata Aster
Titan
Trinity
TripleBit
VelocityGraph
VertexDB
WhiteDB
www.nosql-database.org
www.db-engines.com
www.wikipedia.com

NoSQL

Almacenamiento basado en grafos

- El modelo básico de datos:
 - Grafos dirigidos
 - Nodos y arcos con propiedades



SQL

```
SELECT p.ProductName, p.UnitPrice  
FROM products AS p  
WHERE p.ProductName = 'Chocolade';
```

Cypher es un lenguaje de consulta potente, intuitivo y optimizado para grafos que comprende y aprovecha las conexiones de datos.

Cypher

```
MATCH (p:Product)  
WHERE p.productName = "Chocolade"  
RETURN p.productName, p.unitPrice;
```

AllegroGraph
ArangoDB
Bigdata
Betsy
BrightstarDB
DEX/Sparksee
Execom IOG
Fallen *
Filament
FlockDB
GraphBase
Graphd
Horton
HyperGraphDB
IBM System G Native Store
InfiniteGraph
InfoGrid
jCoreDB Graph
MapGraph
Meronymy
Neo4j
Orly
OpenLink virtuoso
Oracle Spatial and Graph
Oracle NoSQL Database
OrientDB
OQGraph
Ontotext OWLIM
R2DF
ROIS
Sones GraphDB
SPARQLCity
Sqrrl Enterprise
Stardog
Teradata Aster
Titan
Trinity
TripleBit
VelocityGraph
VertexDB
WhiteDB
www.nosql-database.org
www.db-engines.com
www.wikipedia.com

NoSQL

Almacenamiento basado en documentos

- El modelo básico de datos:
 - La noción general de un documento: palabras, frases, oraciones, párrafos, secciones, subsecciones, notas al pie, etc.
 - Esquema flexible: la estructura de los subcomponentes se puede anidar y variar de documento a documento.
 - Metadatos: título, autor, fecha, etiquetas incrustadas, etc.
 - Clave/identificador.

```
{  
    "_id": 1,  
    "first_name": "Tom",  
    "email": "tom@example.com",  
    "cell": "765-555-5555",  
    "likes": [  
        "fashion",  
        "spas",  
        "shopping"  
    ],  
    "businesses": [  
        {  
            "name": "Entertainment 1080",  
            "partner": "Jean",  
            "status": "Bankrupt",  
            "date_founded": {  
                "$date": "2012-05-19T04:00:00Z"  
            }  
        },  
        {  
            "name": "Swag for Tweens",  
            "date_founded": {  
                "$date": "2012-11-01T04:00:00Z"  
            }  
        }  
    ]  
}
```

AmisaDB
ArangoDB
BaseX
Cassandra
Cloudant
Clusterpoint
Couchbase
CouchDB
Densodb
Djondb
EJDB
Elasticsearch
eXist
FleetDB
iBoxDB
Inquire
JasDB
MarkLogic
MongoDB
MUMPS
NeDB
NoSQL embedded db
OrientDB
RaptorDB
RavenDB
RethinkDB
SDB
SisoDB
Terrastore
ThruDB

(www.nosql-database.org
www.db-engines.com
www.wikipedia.com)

NoSQL

Almacenamiento basado en documentos

- Las bases de datos de documentos suelen tener una API o un lenguaje de consulta que permite a los desarrolladores ejecutar operaciones (crear, leer, actualizar y eliminar).
 - **Crear:** Los documentos se pueden crear en la base de datos. Cada documento tiene un identificador único.
 - **Leer:** Los documentos se pueden leer desde la base de datos. La API o lenguaje de consulta permite a los desarrolladores consultar documentos utilizando sus identificadores únicos o valores de campo. Se pueden agregar índices a la base de datos para aumentar el rendimiento de lectura.
 - **Actualizar:** los documentos existentes se pueden actualizar, ya sea en su totalidad o en parte.
 - **Eliminar:** los documentos se pueden eliminar de la base de datos.

```
db.inventory.insertMany([
  { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
  { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
  { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
  { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
  { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);
```

```
SELECT * FROM inventory
db.inventory.find( {} )

SELECT * FROM inventory WHERE status = "D"
db.inventory.find( { status: "D" } )

SELECT * FROM inventory WHERE status = "A" AND qty < 30
db.inventory.find( { status: "A", qty: { $lt: 30 } } )
```

Consultas?

Consultas o comentarios?

Muchas gracias