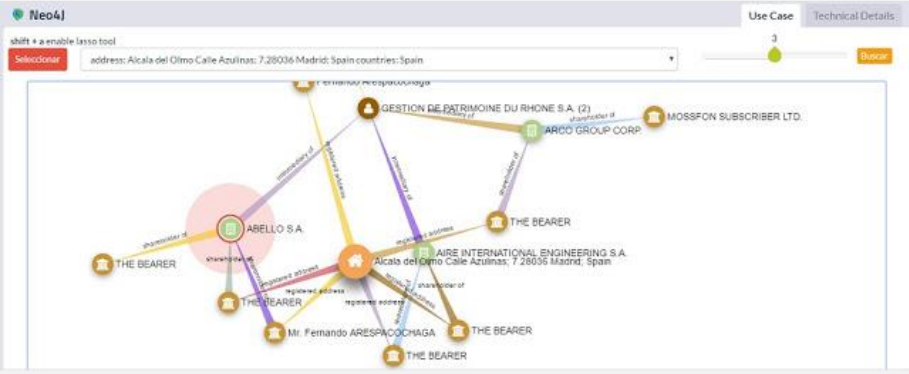
stratebi
open business intelligence

neo4j

Introducción, análisis y utilización de la herramienta Neo4J



CONTENIDO

1. DESCRIPCIÓN 3
2. BASES DE DATOS ORIENTADAS A GRAFOS 4
2.1. INTRODUCCIÓN 4
2.2. EJEMPLO 5
3. HERRAMIENTA Y TECNOLOGÍA 7
4. CONEXIÓN CON NEO4J 7
5. EJERCICIO CON NEO4J E INTEGRACIÓN 9
5.1. CONSULTAS CON NEO4J 9
5.2. INTEGRACIÓN CON PENTAHO DATA INTEGRATION 14
5.3. INTEGRACIÓN CON MICROSOFT POWER BI 20
6. COMPARATIVA ENTRE BASES DE DATOS SQL Y NOSQL 25
7. CONCLUSIONES 26

1. DESCRIPCIÓN

Neo4j es un software libre de base de datos orientada a grafos, creada por Neo Technology Inc e implementado en Java. Neo4j almacena datos estructurados en grafos en lugar de en tablas, es decir, la información se almacena de forma relacionada formando un grafo dirigido entre los nodos y las relaciones entre ellos.

Neo4j está disponible en sistemas operativos de las familias Windows, Linux y OS X, y también está disponible a través de una versión gratuita y una versión de pago. Esta última permite replicación, monitorización y alta disponibilidad..

Asímismo, Neo4j permite acceder a sus datos de diversas formas y usando distintos lenguajes de consulta. Destacan aquí Cypher, un lenguaje que permite consultar y manipular grafos, y Gremlin, un lenguaje que permite gestionar grafos. Sus datos pueden ser accedidos desde una consola de texto, un entorno web (con salida gráfica) y mediante APIs (a través de drives).

Actualmente se sitúa como la base de datos en grafo más popular, utilizando un modelo de grafos de propiedades etiquetados

2. BASES DE DATOS ORIENTADAS A GRAFOS

2.1. Introducción

Una base de datos orientada a grafos (BDOG) representa la información como nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se pueda usar cumpliendo con la teoría de grafos para recorrer la base de datos, ya que esta puede describir atributos de los nodos (entidades) y las aristas (relaciones). Sin embargo, no es necesario ser un experto matemático para poder usarla.

Una BDOG debe estar absolutamente normalizada, esto quiere decir que cada tabla tendría una sola columna y cada relación tan solo dos, con esto se consigue que cualquier cambio en la estructura de la información tenga un efecto solamente local.

Estas bases de datos permiten:

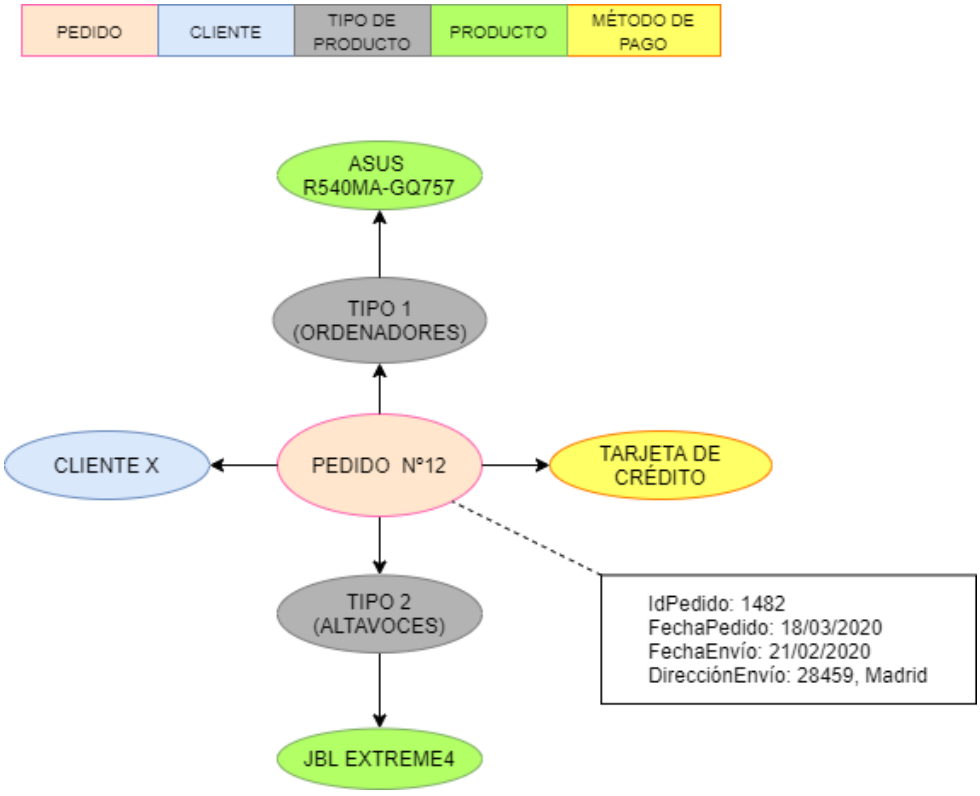
- Realizar consultas más amplias y no desmarcadas por tablas. Por ejemplo, "muestre todas las tablas que posean un nombre Lucas".
- No definir un número determinado de atributos. Por ejemplo, una persona puede tener relacionados cuatro nombres mientras que otra solamente dos, sin desperdiciar espacio.
- Registros de longitud variable, evitando tener que definir un tamaño y también posibles fallos en la base de datos.
- Recorridos directamente en la base de datos de forma jerárquica.
- No utilizar un modelo relacional.
- Mantener la disponibilidad y el acceso a la información.
- En las relaciones poder almacenar atributos.
- Relaciones sin dirección, unidireccionales y bidireccionales, lo que puede convertir la representación a grafos dirigidos, muy útiles en el cálculo de caminos.
- Tener alto rendimiento en la búsqueda de resultados y sobre todo en la búsqueda de caminos.

Las BDOG están pensadas para aquellos ámbitos y casos de uso donde es importante mantener un modelo extenso y amplio de la información, a la vez que un esquema flexible de los datos.

Un claro ejemplo son las redes sociales, donde tenemos el número de conexiones como las relaciones y las personas como nodos del grafo.

2.2. Ejemplo

A continuación, se verá un ejemplo de una BDOG:



Como se puede ver en este ejemplo, se tiene un pedido de ventas junto al cliente que lo ha realizado, los productos que ha comprado y su categoría o tipo, así como el método de pago utilizado para realizar este pedido. Además, se dispone de la información relacionada con el pedido.

Este es un claro ejemplo de BDOG para empresas que venden diferentes productos, ya que tendrán su BBDD organizada en un grafo donde cada nodo será un detalle de la venta, que tendrá además la información explicada anteriormente. Por ello, si la empresa quisiera, podría ver todos los productos del tipo "ordenador" que ha vendido. Asimismo, este modelo BDOG permite tener pedidos que contengan más o menos productos, así como productos cuyo nombre tenga un tamaño diferente al anterior.

Además, esto se podría ampliar para que cada pedido con su respectiva información sea un nodo diferente dentro de un grafo a nivel empresa formado por los diferentes pedidos, es decir, un mayor nivel de abstracción.

3. HERRAMIENTA Y TECNOLOGÍA

Neo4j está diseñado específicamente para trabajar con datos altamente conectados, ofrece un rendimiento ultrarrápido y permite una visión poderosa y procesable.

Al analizar intuitivamente los puntos de datos y las conexiones entre ellos, Neo4j impulsa aplicaciones inteligentes en tiempo real que abordan los desafíos más difíciles de la actualidad.

La forma de trabajar con los datos de Neo4j es totalmente diferente a lo habitual, se accede a partir de sus tipos (relaciones), sus interrelaciones (claves foráneas), sus identificadores (claves primarias) y sus atributos. En Neo4j, para identificar un conjunto de datos sobre los que aplicar una operación, se realiza *graph traversing*, es decir, una navegación por el grafo desde un inicio hasta obtener los resultados buscados.

Las peculiaridades de los grafos y el diseño de Neo4j hacen que esta base de datos soporte mejor la escalabilidad vertical que la horizontal. Neo4j está enfocada a garantizar la consistencia de los datos y la disponibilidad.

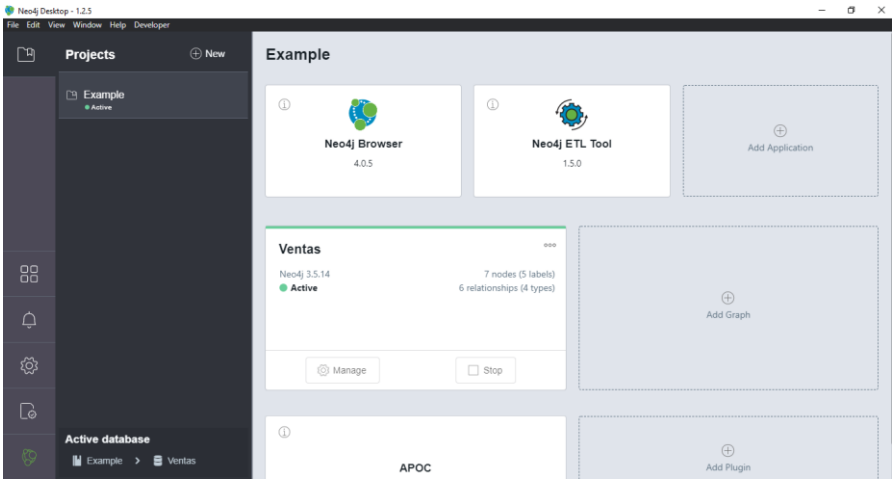
4. CONEXIÓN CON NEO4J

Primero, desde la página web https://neo4j.com/download/ se debe descargar la *Download Community Edition*. En teoría, el navegador debe lleva a la página de descarga de Neo4j para el sistema operativo que se disponga.

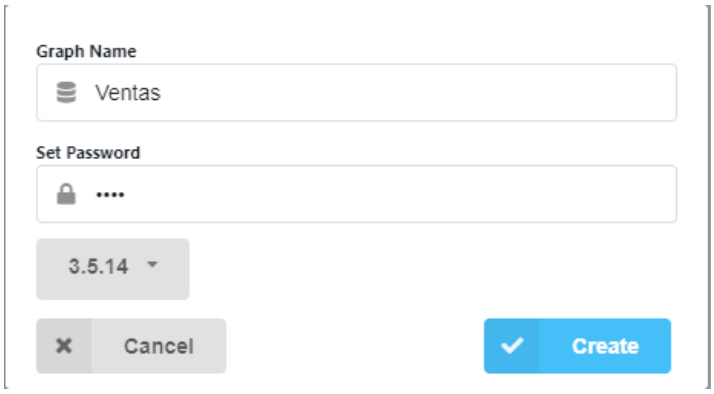
Después, se debe arrancar el servicio de Neo4j. En caso de tener un sistema operativo Windows, se debe hacer click sobre el icono de Neo4j en el menú de inicio. El sistema pedirá que se ubique la base de datos con la que trabajar y, si no se indica nada, se trabajará con una base de datos vacía. En caso de tener un sistema operativo distinto de Windows, se debe abrir un terminal y, una vez en el terminal, se debe ir al directorio donde se ha desinstalado el programa y ejecutar la siguiente sentencia*: bin/neo4j start*. Para el ejercicio que se propone a continuación, se utiliza Windows.

Al abrir la aplicación, se debe crear un proyecto de nombre *Example* en el cual se añadirán las diferentes bases de datos orientadas a grafos que se utilizarán a lo largo de este documento. Como ejemplo, se crea la primera BDOG, con nombre *Ventas*, según se explica a continuación:

Primero, se selecciona la opción *+New Project* y se le asigna el nombre *Example*.



Después, we selecciona *+Add Graph y Create Local Graph* y se le nombra Ventas. Además, se le asigna una contraseña propia para poder acceder a este grafo y se selecciona su versión de Neo4j, en este caso, se utiliza la versión 3.5.14.



Por último, se inicializa el grafo a través del botón *Start*. Recuerda que solo se puede tener una BDOG activa. De esta forma se crearán los demás grafos (BDOG) utilizados en el resto de ejemplos.

5. EJERCICIO CON NEO4J E INTEGRACIÓN

5.1. Consultas con Neo4j

Como se ha mencionado anteriormente, Neo4j permite el acceso usando distintos lenguajes, en este caso se utilizará Cypher. Para trabajar con Cypher y ampliar los conocimientos en este lenguaje, se recomienda la descarga de *Cypher Manual 4.0* en https://neo4j.com/docs/.

El objetivo de este ejercicio es crear el grafo que se muestra en la imagen del epígrafe 2.2 de este documento. Una vez se haya conectado a la base de datos creada (Ventas), se debe abrir *Neo4j Browser* que a su vez levanta una instancia en el local del equipo: *http://localhost:7474/browser/.*

En primer lugar, hay que crear los nodos con sus campos de información en caso de tenerlos:

- CREATE (n°12:Pedido{IdPedido:12,FechaPedido:'18/03/2020',FechaEnvio:'21/03/2020', DireccionEnvio:'28459, Madrid'}),(TarjetaDeCredito: MetodoDePago{Nombre:'TarjetaDeCredito'}), (X:Cliente{Nombre:'X'}), (Ordenador:TipoDeProducto{Nombre:'Ordenador'}), (Altavoz:TipoDeProducto{Nombre:'Altavoz'}), (AsusR540MA:Producto{Nombre:'AsusR540MA'}), (JBLextreme4:Producto{Nombre:'JBLextreme4'})

A continuación, se crean las relaciones entre nodos:

- MATCH (n°12:Pedido{IdPedido:12,FechaPedido:'18/03/2020',FechaEnvio:'21/03/2020', DireccionEnvio:'28459, Madrid'}) MATCH (X:Cliente{Nombre:'X'}) CREATE (n°12)-[:Bought_by]->(X)
- MATCH (n°12:Pedido{IdPedido:12,FechaPedido:'18/03/2020',FechaEnvio:'21/03/2020', DireccionEnvio:'28459, Madrid'}) MATCH (TarjetaDeCredito: MetodoDePago{Nombre:'TarjetaDeCredito'}) CREATE (n°12)-[:Pay_by]->(TarjetaDeCredito)
- MATCH (Ordenador:TipoDeProducto{Nombre:'Ordenador'}) MATCH (AsusR540MA:Producto{Nombre:'AsusR540MA'}) CREATE (Ordenador)-[:MoreDetails]->(AsusR540MA)
- MATCH (Altavoz:TipoDeProducto{Nombre:'Altavoz'}) MATCH (JBLextreme4:Producto{Nombre:'JBLextreme4'}) CREATE (Altavoz)-[:MoreDetails]->(JBLextreme4)
- MATCH (n°12:Pedido{IdPedido:12,FechaPedido:'18/03/2020',FechaEnvio:'21/03/2020', DireccionEnvio:'28459, Madrid'}) MATCH (Ordenador:TipoDeProducto{Nombre:'Ordenador'}) CREATE (n°12)-[:Type_of]->(Ordenador)

- MATCH (n°12:Pedido{IdPedido:12,FechaPedido:'18/03/2020',FechaEnvio:'21/03/2020', DireccionEnvio:'28459, Madrid'}) MATCH (Altavoz:TipoDeProducto{Nombre:'Altavoz'}) CREATE (n°12)-[:Type_of]->(Altavoz)

Si se quisiera eliminar el grafo anterior:

- MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r

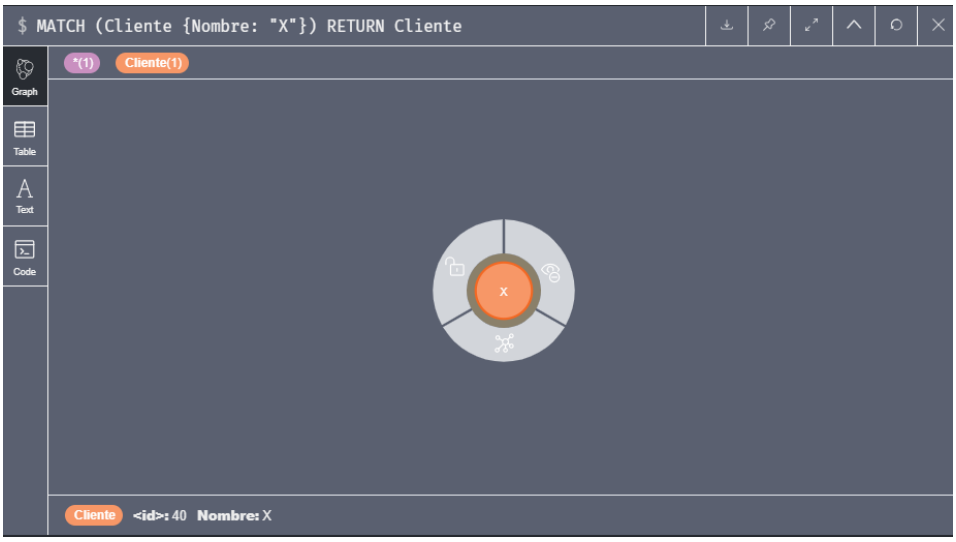
El grafo obtenido se visualiza con la siguiente sentencia:

- MATCH (n) RETURN n

Para realizar búsquedas a través del grafo, algunos ejemplos son los siguientes:

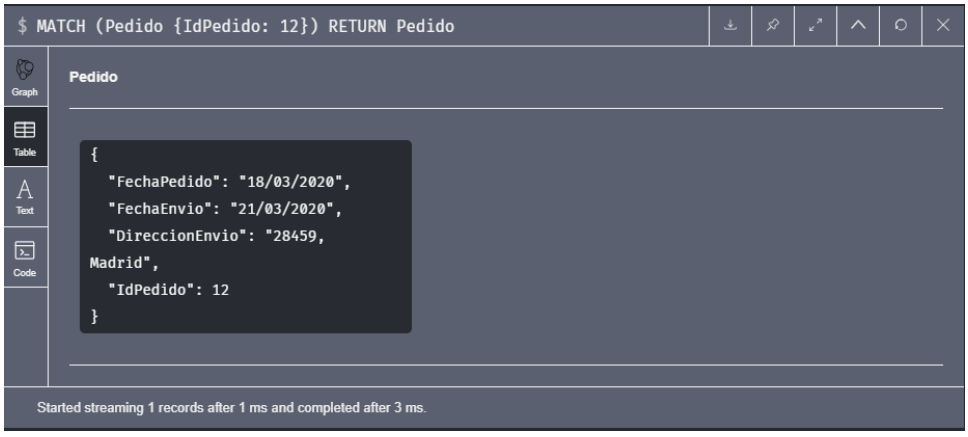
- Buscar los clientes del grafo:

MATCH (Cliente {Nombre: "X"}) RETURN Cliente



- Buscar el pedido del grafo:

```
MATCH (Pedido {IdPedido: 12}) RETURN Pedido
```



- Buscar el nombre del tipo de producto junto al Id del producto para aquellas relaciones Tipe_of:

```
MATCH (Pedido)-[:Tipe_of]->(TipoDeProducto) RETURN Pedido.IdPedido, TipoDeProducto.Nombre
```

```
$ MATCH (Pedido)-[:Tipe_of]→(TipoDeProducto) RETURN Pedido.IdPedido,…
```

Table | Text | Code

Pedido.IdPedido	TipoDeProducto.Nombre
12	"Ordenador"
12	"Altavoz"

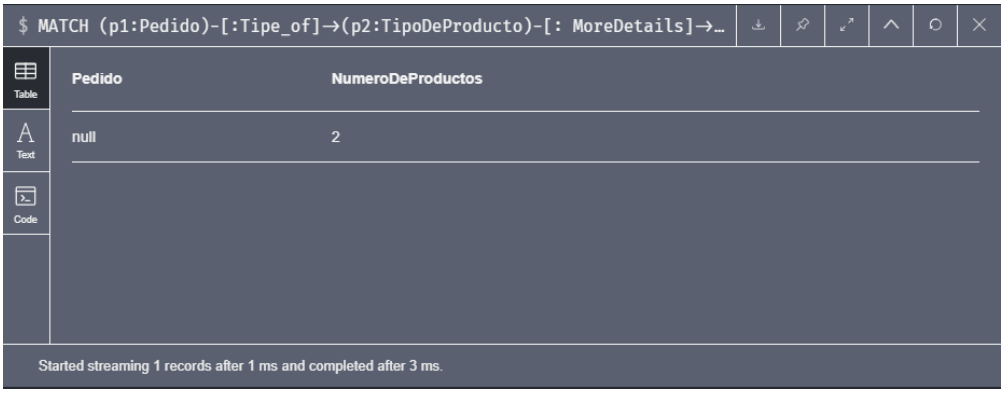
Started streaming 2 records after 2 ms and completed after 4 ms.

- Buscar aquellos pedidos cuyo Id sea mayor que 10:

```
MATCH (p:Pedido) WHERE p.IdPedido > 10 RETURN p
```

- Contar el número de productos por pedido:

```
MATCH (p1:Pedido)-[:Tipe_of]->(p2:TipoDeProducto)-[: MoreDetails]->(p3:Producto) RETURN
p1.Nombre AS Pedido,  COUNT(p3) AS NumeroDeProductos
```



Estos son algunos de los muchos ejemplos de búsqueda que se pueden realizar en cualquier grafo, es decir en cualquier BDOG a través de Neo4j. Como se ha visto, se pueden realizar búsquedas por nodo, relación, nodo y relación, con restricciones (clausulas *where*) y con funciones agregadas.

5.2. Integración con Pentaho Data Integration

Pentaho Data Integration (PDI) es la herramienta ideal para alimentar Neo4j ya que puede unir datos de diferentes fuentes, limpiarlos, transformarlos y formatearlos, así como agregar datos de las búsquedas. Las posibilidades son casi ilimitadas ya que tiene todo el poder de una herramienta ETL junto a una forma fácil para transformar los datos de acuerdo con sus requisitos.

Para reproducir este ejemplo, en primer lugar, se debe instalar *Pentaho Data Integration (PDI)* y una vez se haya hecho esto, hay que descargar los drivers de Neo4J para Pentaho Integración. Para ello, se debe descargar el archivo *Neo4JOutput-4.0.0-beta1.zip* desde https://github.com/knowbi/knowbi-pentaho-pdi-neo4j-output/releases/, descomprimirlo e incluirlo en la carpeta *'plugins'* de PDI. Tras esto, ya se puede trabajar con Neo4J en PDI.

En este caso, se presenta un ejemplo con datos de Aeropuertos que tienen el siguiente formato:

- **Id:** Identificador único del aeropuerto.
- **Nombre:** Nombre del aeropuerto.
- **Descripción:** Descripción del aeropuerto.
- **País:** Nombre del país.
- **IATA:** Código IATA de tres letras del aeropuerto.
- **ICAO:** Código ICAO de cuatro letras del aeropuerto.
- **Ubicación:** Latitud y longitud del aeropuerto, separadas por una coma.
- **Elevación:** Elevación del aeropuerto.

El siguiente paso, será crear un fichero CSV (aeropuertos.csv) con los datos sobre aeropuertos, a partir de los cuales se crearán los nodos que se visualizarán en Neo4j. Los datos son los siguientes:

1; Goroka; Goroka; Papua Nueva Guinea; GKA; AYGA; -6.08169,145.39188; 5282

2; Madang; Madang; Papua Nueva Guinea; MAG; AYMD; -5.20708,145.7887; 20

3; Mount Hagen; Mount Hagen; Papua New Guinea; HGU; AYMH; -5.82679,144.29586; 5388

4; Nadzab; Nadzab; Papua Nueva Guinea; LAE; AYNZ; -6.56983,146.72624; 239

5; Port Moresby Jacksons Intl; Port Moresby; Papua Nueva Guinea; POM; AYPY; -9.44338,147.22005;146

6; Wewak Intl; Wewak; Papua Nueva Guinea; WWK; AYWK; -3.58383,143.66919; 19

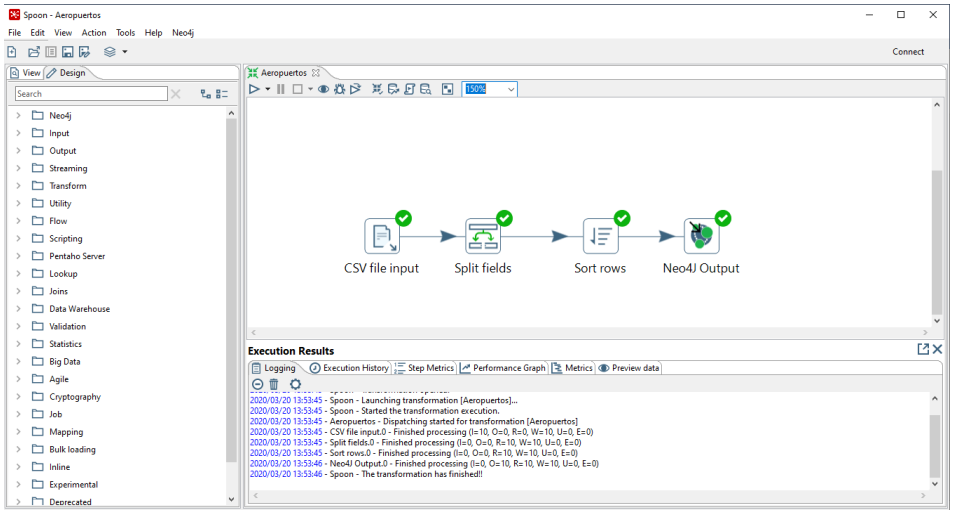
7; Narsarsuaq; Narssarssuaq; Groenlandia; UAK; BGBW; 61.16052, -45.42598; 112

8; Nuuk; Godthaab ; Groenlandia; GOH; BGGH; 64.19092, -51.67806; 283

9; Sondre Stromfjord; Sondrestrom; Groenlandia; SFJ; BGSF; 67.01697, -50.68932; 165

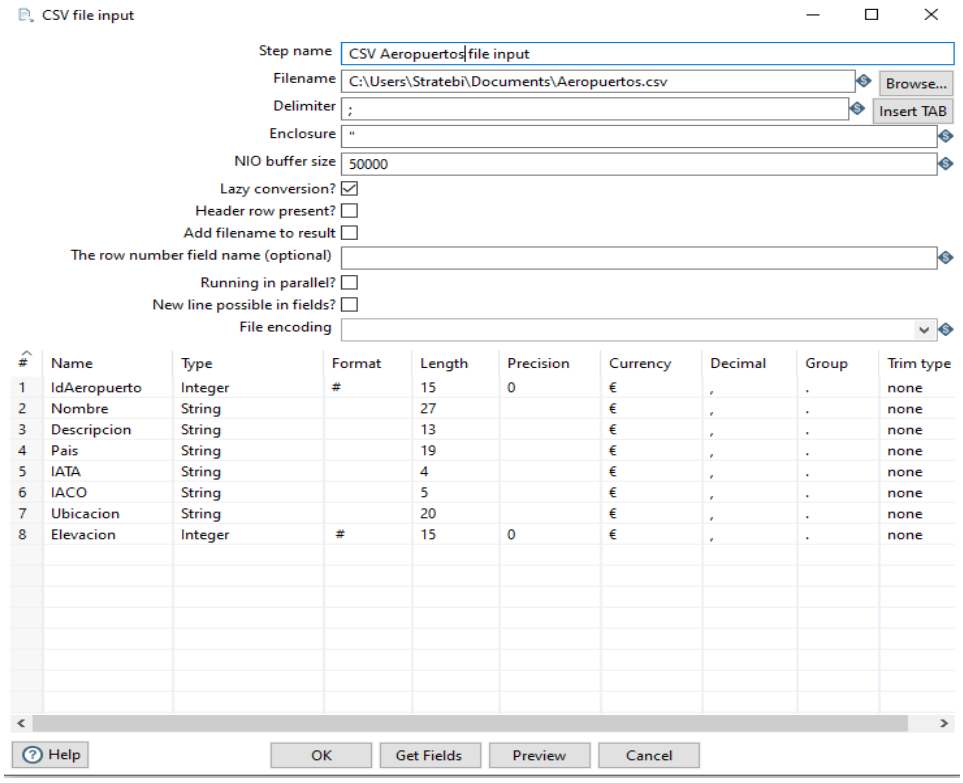
10; Thule Air Base; Thule; Groenlandia; THU; BGTL; 76.5312, -68.70316; 251

Una vez creado el fichero, se comienza con el proceso ETL en Pentaho Data Integration, creando una transformación como la que procede:



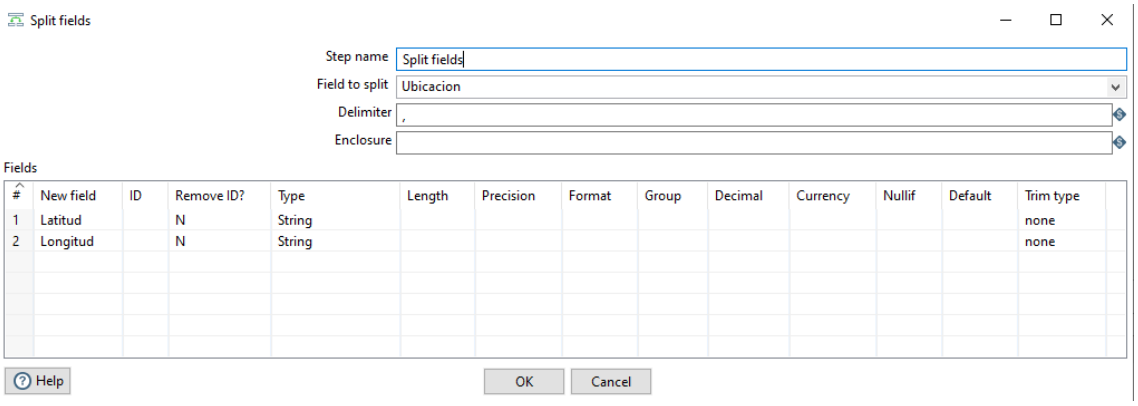
Cada uno de los pasos se explica detalladamente a continuación:

El primer paso es importar los datos del fichero .csv que se ha creado a través del paso *CSV file input*:



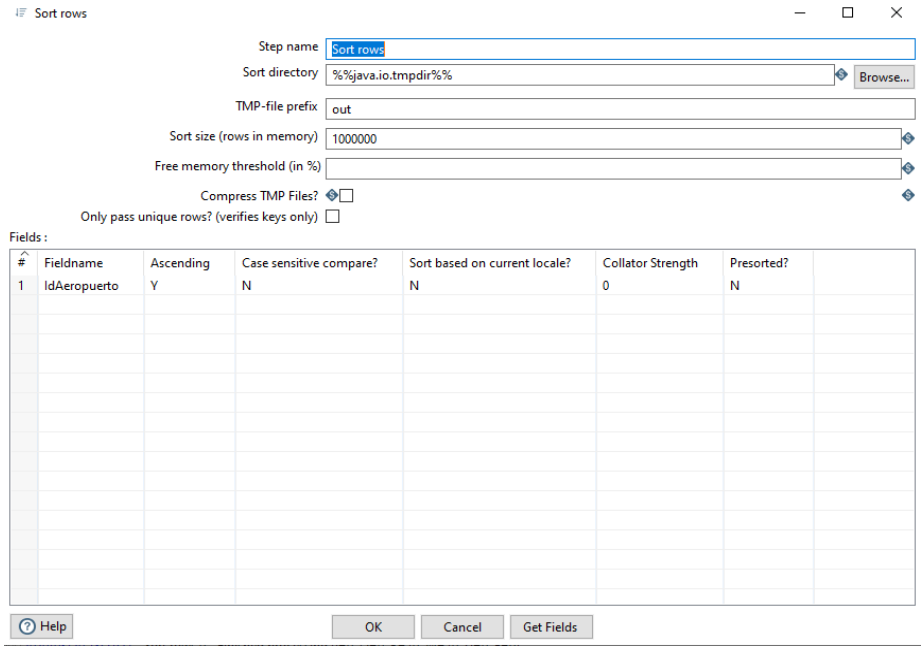
#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	IdAeropuerto	Integer	#	15	0	€	,	.	none
2	Nombre	String		27		€	,	.	none
3	Descripcion	String		13		€	,	.	none
4	Pais	String		19		€	,	.	none
5	IATA	String		4		€	,	.	none
6	IACO	String		5		€	,	.	none
7	Ubicacion	String		20		€	,	.	none
8	Elevacion	Integer	#	15	0	€	,	.	none

Como el campo U*bicación* está formado por la *Latitud* y la *Longitud*, se separa en dos nuevos campos mediante el paso *Split Fields*:



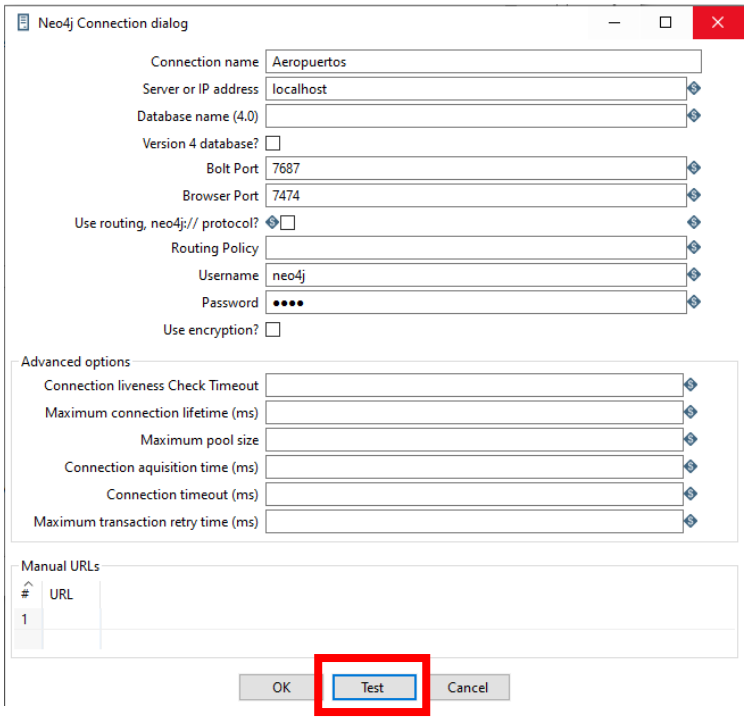
#	New field	ID	Remove ID?	Type	Length	Precision	Format	Group	Decimal	Currency	Nullif	Default	Trim type
1	Latitud		N	String									none
2	Longitud		N	String									none

Con el propósito de tratar con datos ordenados, utilizando el paso *Sort Rows,* se ordenan por el campo *IdAeropuerto*:

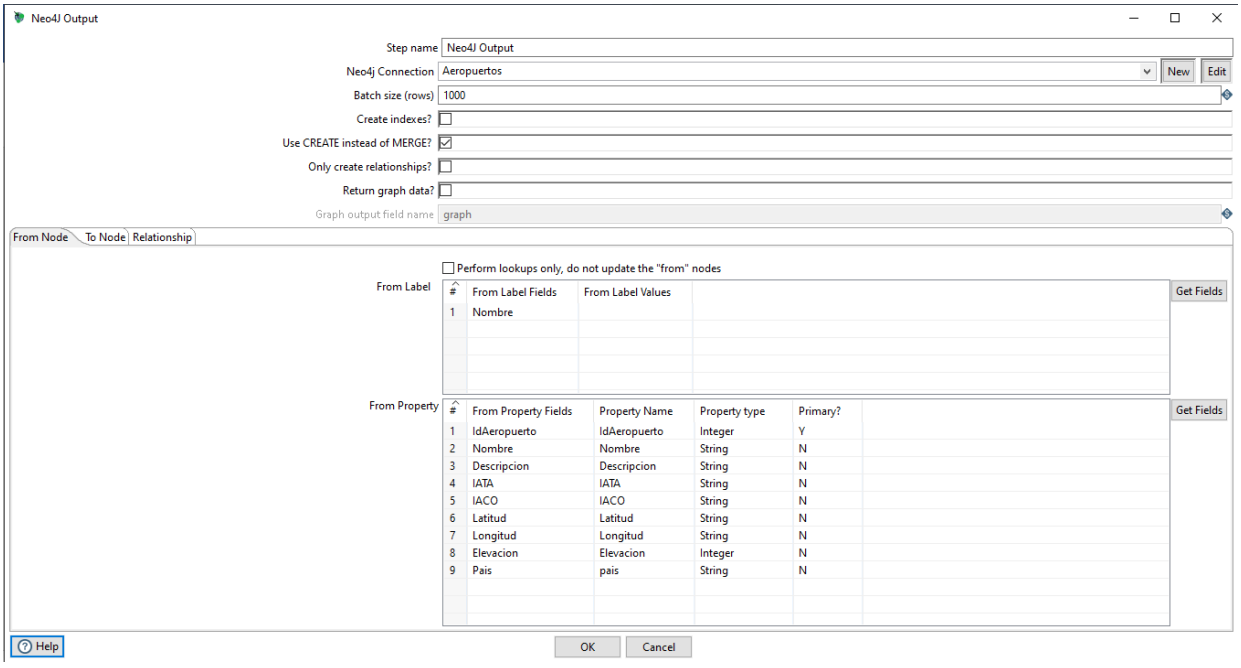


Por último, el último paso es la salida de los datos a Neo4j. Para ello, se utiliza el paso *Neo4j Output* mediante el cual se establece la conexión con la base de datos de *Neo4j* y se importan en ella los datos resultantes de este proceso ETL. Se debe tener en cuenta, que antes de realizar este paso, se debe crear en Neo4j una base de datos/conexión/grafo llamada *Aeropuerto*s. Este proceso se explica en el capítulo 4 de este documento.

Primero se establece la conexión y se comprueba que funciona correctamente a través del botón *Test*:



Después se completa el paso de salida de datos de la siguiente manera:

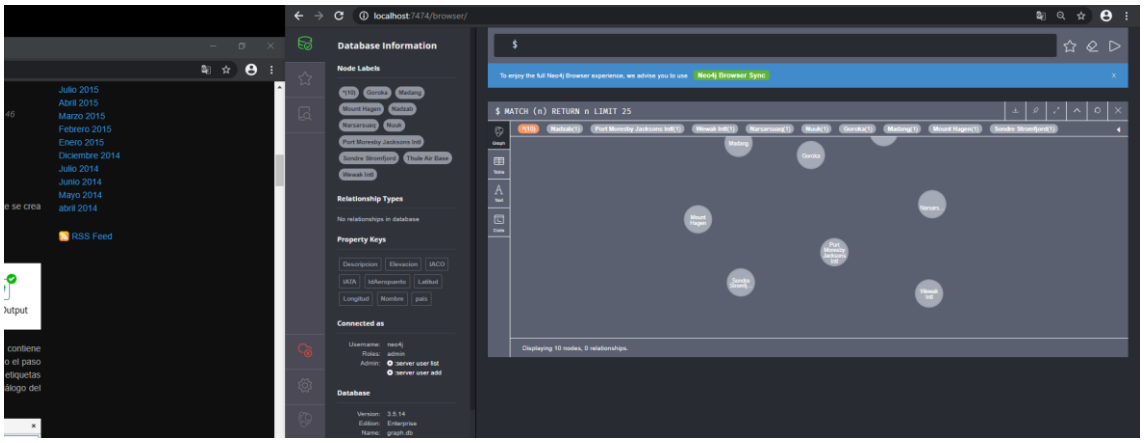


Una vez aceptado este paso, se pasa a ejecutar la ETL en el botón de *Run* de PDI:

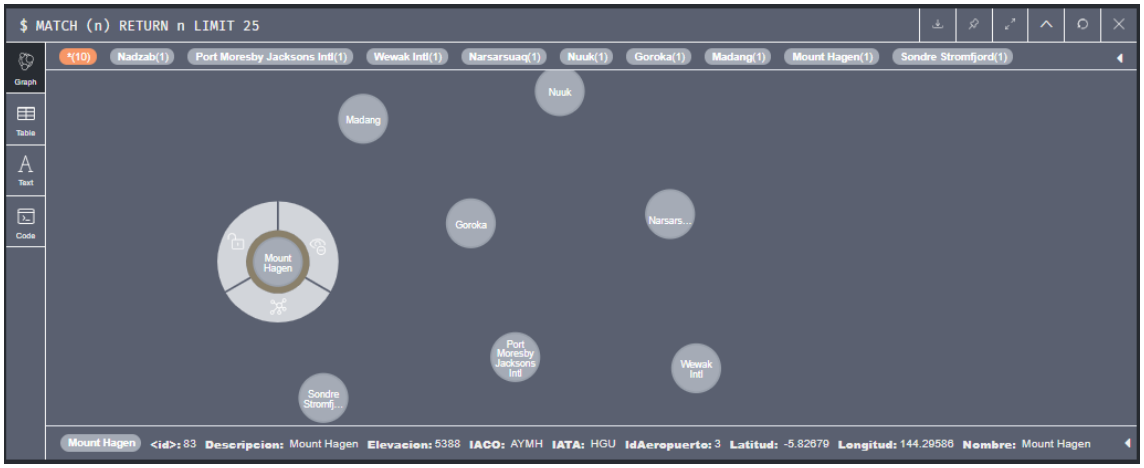
Cuando se termina de ejecutar la transformación en Pentaho Data Integration, se completa un proceso en el que los datos origen de los aeropuertos del fichero *.csv* ya están transformados e importados en la ruta local de Neo4j: *http://localhost:7474/browser/.*

Para visualizar el resultado final, se debe que ejecutar la sentencia dentro del browser:

```
MATCH (n) RETURN (n)
```



Visualmente se puede ver como cada nodo representa a un aeropuerto con su respectiva información:



Esto es solo un sencillo ejemplo de cómo se puede integrar un conjunto de datos origen en Neo4j a través de Pentaho Data Integration en el que se aprecia cómo, de manera transparente, se ejecuta el proceso de cargar un los datos de un origen en Neo4J sin conocimientos previos de cómo funciona Neo4J.

5.3. Integración con Microsoft Power BI

Neo4j, de momento, no está disponible de manera oficial para ser tratada a través de Power BI. Sin embargo, para poder utilizar Neo4j en Microsoft Power Bi, se puede descargar una extensión beta que permite visualizar los datos del grafo creado en Neo4j.

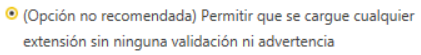
Para ello, lo primero es descargar el archivo *Neo4j.mez* en:

https://github.com/cskardon/Neo4jDataConnectorForPowerBi/releases/tag/1.2.0

Y guardar el archivo en la siguiente ruta (si no existe la ruta, se deben crear las carpetas necesarias):

C:\Users\Stratebi\Documents\Power BI Desktop\Custom Connectors

Una vez hecho esto, en ***Archivo -> Opciones y configuración -> Opciones -> Seguridad***, se debe habilitar la siguiente opción y reiniciar Power BI:

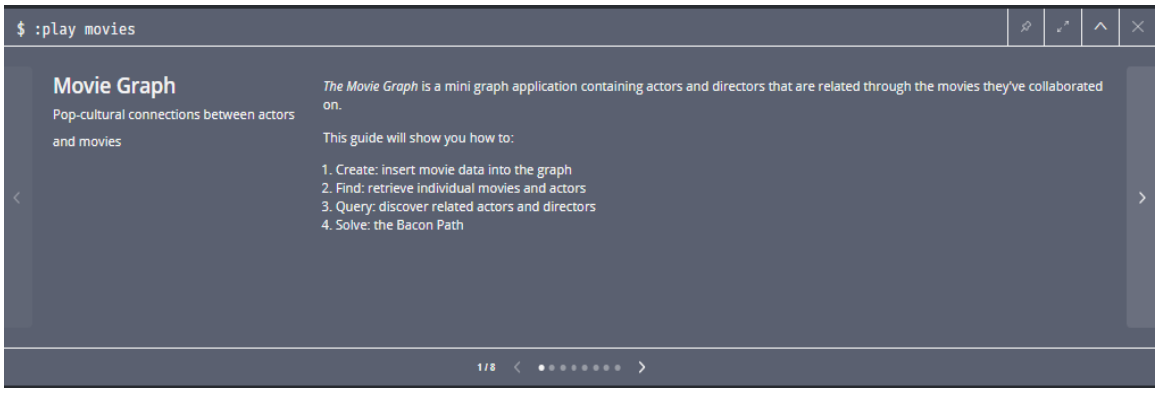


Para este ejemplo, se utilizará un ejemplo facilitado por Neo4j de películas de cine. Es un grafo de películas y actores que viene con la instalación de Neo4j llamado 'The Movie Graph', en el que se incluyen datos de películas, una descripción de las mismas y su año de lanzamiento.

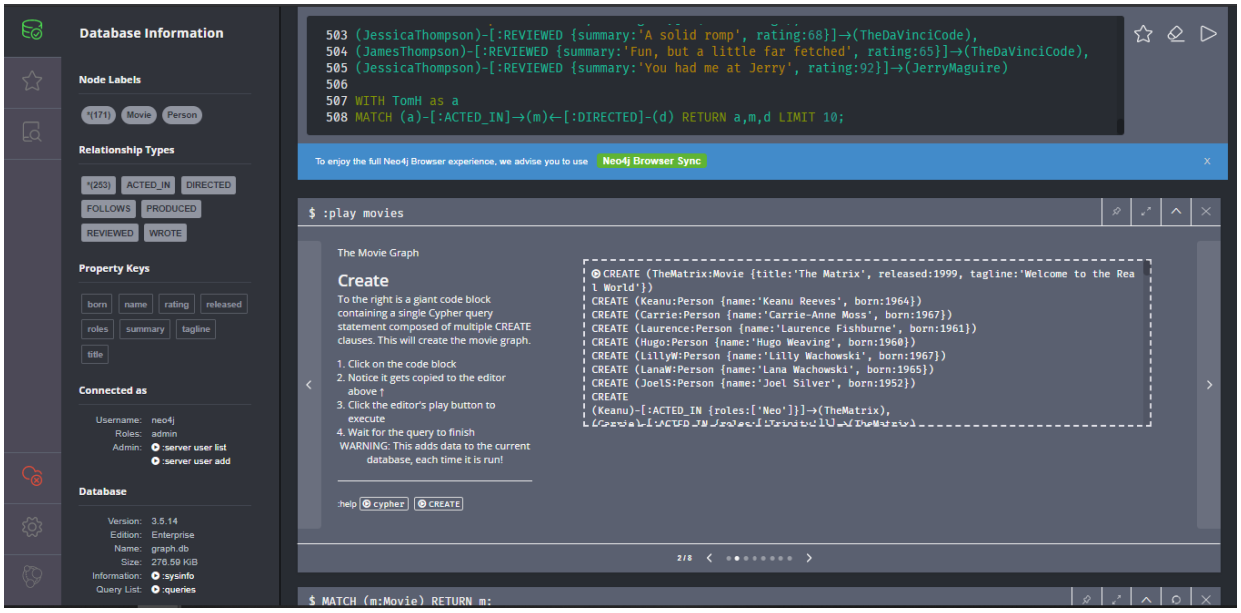
En primer lugar, se debe crear, un grafo llamado *movies*, tal y como se explica en el apartado 4, y continuación, se debe ejecutar en el browser/consola de Neo4j lo siguiente:

- :play movies

Aparecerá la siguiente pestaña en la cual, si se avanza de página, se podrá ejecutar la sentencia *create* de este grafo de películas a modo de ejemplo:

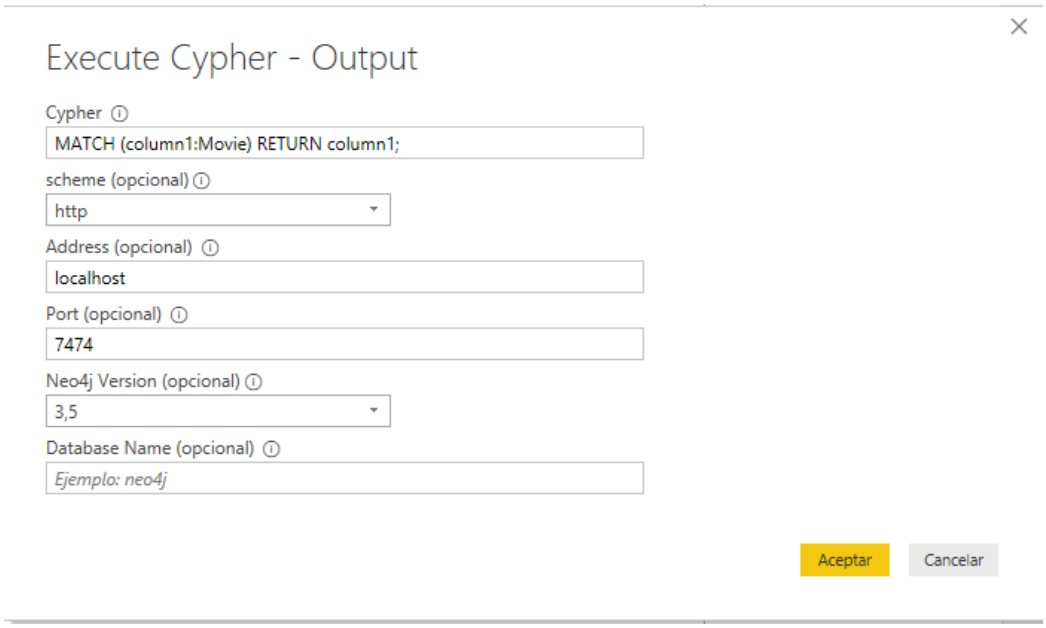


Se debe pulsar sobre *Create* y ejecutar en el browser la sentencia:



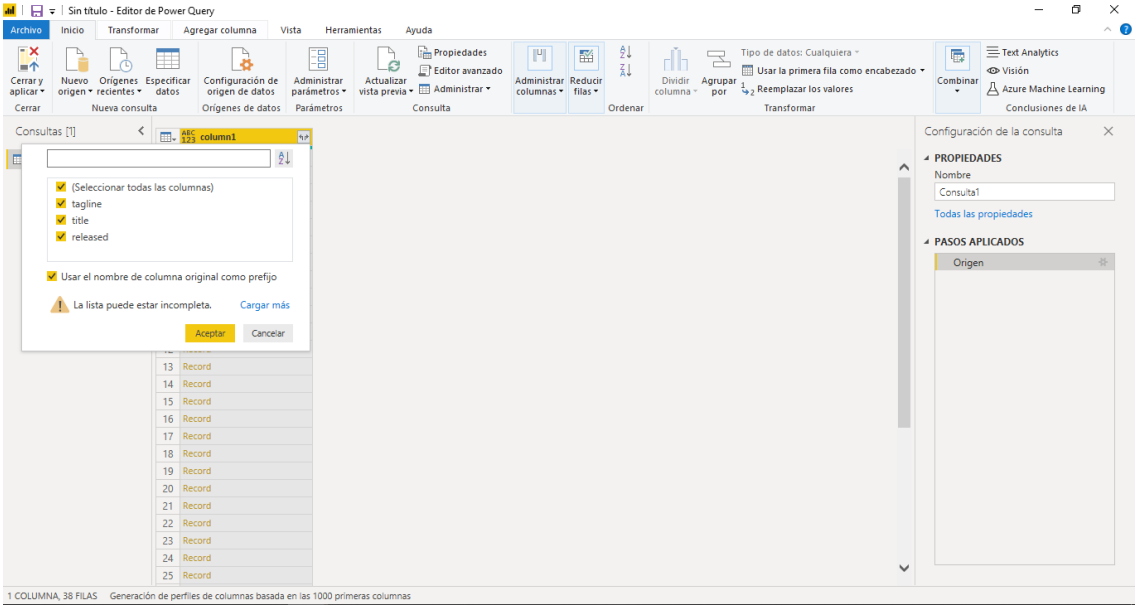
Ahora hay que abrir Power BI de nuevo y pulsar sobre la opción *Obtener Datos*, buscar Neo4j (beta) e introducir los campos según la imagen para conectar.

- **Cypher:** sentencia que se desea ejecutar, en este caso, mostrar todos los datos del grafo.
- **Scheme:** esquema de conexión.
- **Address:** dirección de la conexión.
- **Port:** puerto de la conexión.
- **Neo4j Version:** versión de Neo4j que se desea utilizar.



El usuario y contraseña son los que se han usado en Neo4J.

En la siguiente pantalla, elegir la opción Transformar Datos y, una vez abierto el editor de Power Query, pulsar sobre el botón derecho de la esquina superior donde pone *column 1*, después sobre *cargar más* y, a continuación, *aceptar*.

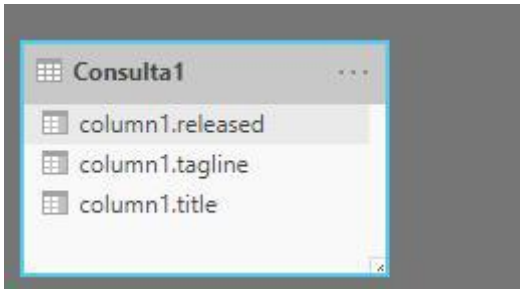


Una vez realizado lo anterior, seleccionar la opción cerrar y aplicar situada en la esquina superior izquierda. Ahora, se aplicarán los cambios y ya se podrán representar visualmente estos datos.

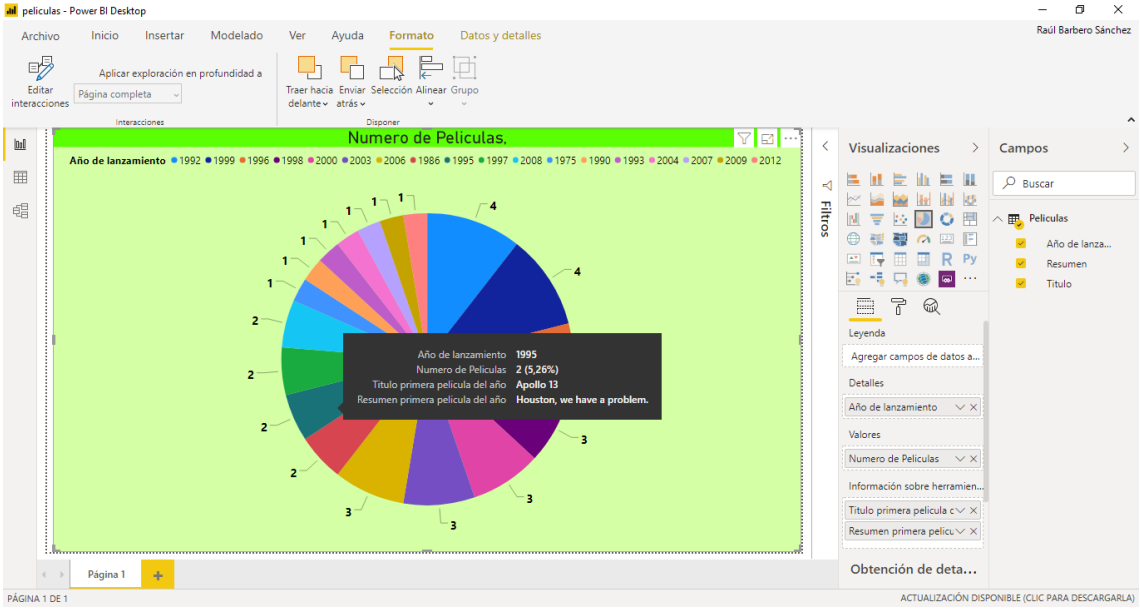
Los datos en este ejemplo de películas, que se visualizan en la pestaña de *Datos* de Power BI, son los siguientes:

Resumen	Titulo	Año de lanzamiento
If he's crazy, what does that make you?	One Flew Over the Cuckoo's Nest	1975
I feel the need, the need for speed.	Top Gun	1986
For some, it's the last real taste of innocence, and the first	Stand By Me	1986
A story of love, lava and burning desire.	Joe Versus the Volcano	1990
In the heart of the nation's capital, in a courthouse of the	A Few Good Men	1992
It's a hell of a thing, killing a man	Unforgiven	1992
He didn't want law. He wanted justice.	Hoffa	1992
Once in a lifetime you get a chance to do something differ	A League of Their Own	1992
What if someone you never met, someone you never saw,	Sleepless in Seattle	1993
The hottest data on earth. In the coolest head in town	Johnny Mnemonic	1995
Houston, we have a problem.	Apollo 13	1995
In every life there comes a time when that thing you drear	That Thing You Do	1996
Come as you are	The Birdcage	1996
Don't Breathe. Don't Look Back.	Twister	1996
Evil has its winning ways	The Devil's Advocate	1997
A comedy from the heart that goes for the throat.	As Good as It Gets	1997
After life there is more. The end is just the beginning.	What Dreams May Come	1998
At odds in life... in love on-line.	You've Got Mail	1998
Can two friends sleep together and still love each other in	When Harry Met Sally	1998
Welcome to the Real World	The Matrix	1999
First loves last. Forever.	Snow Falling on Cedars	1999
Walk a mile you'll never forget.	The Green Mile	1999
One robot's 200 year journey to become an ordinary man.	Bicentennial Man	1999

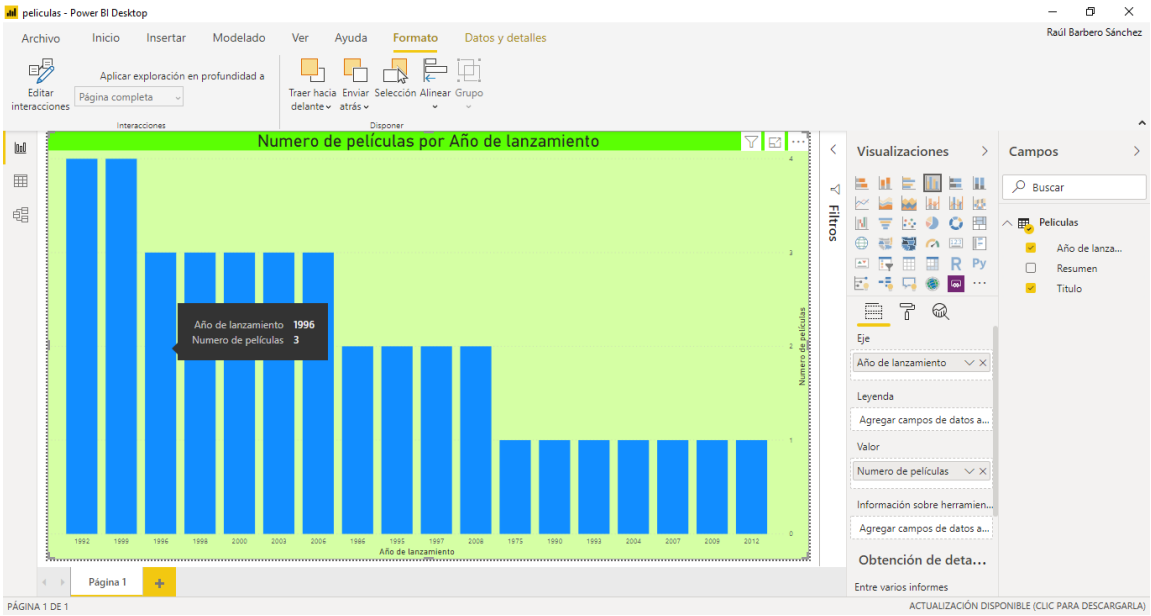
En definitiva, se puede apreciar que, a pesar de estar trabajando con una base de datos orientada a grafos, cuando en Power BI se importan los datos, en este ejemplo, éstos se comportan como si de una base de datos relacional se tratara y el comportamiento es totalmente transparente para la persona encargada de explotar estos datos en Power BI.



Por último, a modo de ejemplo, se crea un *pie chart* que muestra la distribución de películas por año de lanzamiento.



Otra alternativa de visualización podría ser un ranking de películas por año de lanzamiento mediante un gráfico de barras:



6. COMPARATIVA ENTRE BASES DE DATOS SQL Y NOSQL

Las **ventajas** del uso de este tipo de bases de datos, son las siguientes:

- Presentan una gran flexibilidad en la organización, representación y consumo de datos.
- Realizan un aprovechamiento máximo de las interacciones entre sus componentes, así como una presentación intuitiva y resumida de las relaciones.
- La velocidad de búsqueda depende únicamente del número de relaciones concretas, no del conjunto de datos.
- Contiene estructuras flexibles, es decir cuando aumentan las necesidades, las posibilidades de añadir más nodos y relaciones a un grafo ya existente son enormes.
- Gran rendimiento. El peso de los datos crece de forma acelerada y exponencial. En la misma medida o velocidad crecerán las relaciones en los datos. Ante tal dinámica, las BDOG, nos ofrece una amplia capacidad de responder a las exigencias estructurales de estos volúmenes de datos. Tienen un mayor rendimiento frente a las relacionales (SQL) y las no relacionales (NoSQL). La clave es que, aunque las consultas de datos aumenten exponencialmente, el rendimiento de Neo4j no desciende. Responden a las consultas actualizando el nodo y las relaciones de esa búsqueda y no todo el grafo completo. Esto optimiza mucho el proceso.
- Responden con agilidad en la gestión de datos. Si se quisiera llevar al límite sus capacidades, habría que superar un volumen total de 34.000 millones de nodos (datos), 34.000 millones de relaciones entre esos datos, 68.000 millones de propiedades y 32.000 tipos de relaciones.
- La velocidad de procesamiento que ofrece en un entorno de alta exigencia, donde las decisiones deben ser tomadas a toda velocidad, es fundamental.

Los **inconvenientes** en las BDOG se observan respecto a la atomicidad y a sus patrones de estandarización. Además, en algunos casos cuando se realizan particiones en función de propiedades locales, pueden presentarse dificultades similares a las de una base de datos relacional.

En una BDOG es difícil escalar, por estar diseñada para arquitecturas con un solo servidor. A su vez, con una menor importancia, no existe un lenguaje de consulta consistente y el uso de estas bases de datos, requiere un cambio conceptual para los desarrolladores.

Por tanto, las principales desventajas son la falta de estandarización y la ralentización de la búsqueda de nodos físicamente distribuidos.

7. CONCLUSIONES

Las bases de datos orientadas a grafos no deberían considerarse, por lo general, como mejores y absolutas sustitutas de las bases de datos convencionales. Las estructuras relacionales siguen siendo modelos estándar que garantizan una gran estabilidad e integridad de los datos, además de permitir un escalamiento flexible. Como en la mayoría de los casos, todo depende del uso que se les quiera dar.

Sin embargo, en un gran numero de situaciones y casos, estas bases de datos están dotadas de una gran flexibilidad, rendimiento, velocidad, agilidad y rendimiento por lo que son muy recomendables y muy a tener en cuenta. Quizás, si se encontrara una solución a algunos de sus inconvenientes, estas BDOG serían utilizadas en un mayor número de casos y serían la principal alternativa a las bases de datos relacionales, siendo Neo4J una de las que se colocan en el pódium de bases de datos orientadas a grafos.

Por último, a lo largo de este documento, se ha podido validar como su uso en procesos de Business Intelligence, concretamente en herramientas como *Pentaho Data Integration* o *Power BI*, es totalmente transparente para el usuario. Además, estas herramientas están totalmente preparadas y capacitadas para soportar casos con Neo4J, incluso, en Power BI tendría el mismo comportamiento que si se importara cualquier otra fuente de datos, almacenando los datos en tablas de Power BI.

También puedes echar un vistazo a esta aplicación que hemos creado con Neo4J sobre los 'Panama Papers':

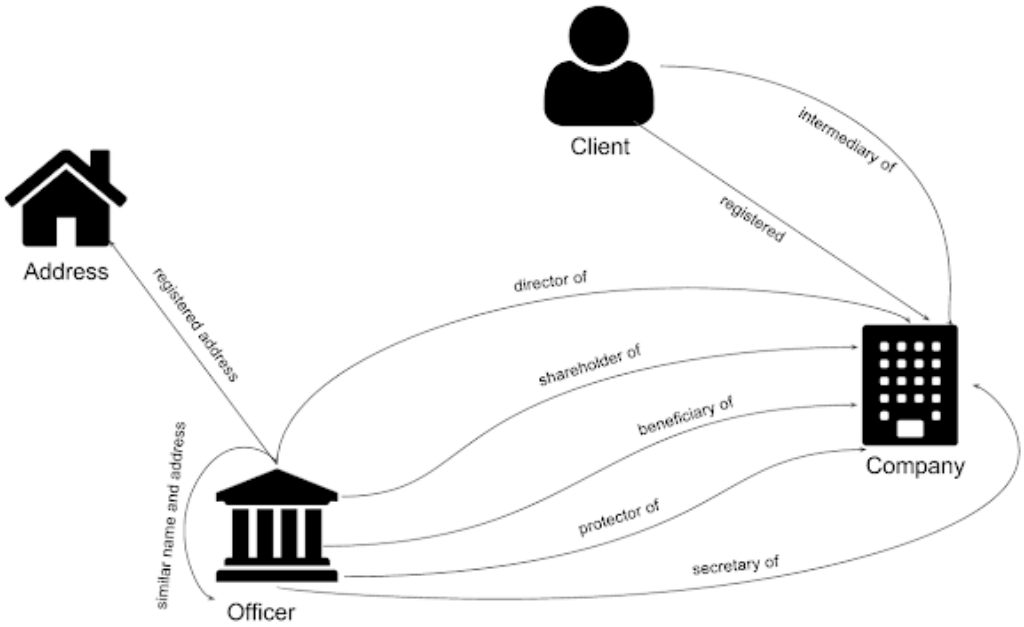
Acceso Aplicacion

En este ejemplo se usa Neo4j como Base de Datos basada en grafo para modelar las relaciones entre las entidades que forman parte de los Papeles de Panamá (PP). A partir de ficheros de texto con los datos y relaciones entre clientes, oficinas y empresas que forman parte de los PP, hemos creado este grafo que facilia la comprensión de las interacciones entre sujetos distintos en esta red.

La demostración comienza seleccionando una entidad de cualquier tipo (Address, Company, Client, Officer), según el tipo que seleccione se muestran los atributos de ese nodo, luego seleccione el atributos que desea e introduzca el filtro, agregando varios paneles para filtrar por más de uno si es

necesario. El parámetro "Deep" significa el número de conexiones al elemento seleccionado que se quiere mostrar.

En el servidor se hace una búsqueda BFS a partir del nodo seleccionado realizando consultas a Neo4j para cada tipo de relación donde una de sus partes sea el nodo actual, hasta llegar al nivel de profundidad solicitado. Se van guardando los nodos y los arcos para devolverlos como resultado.



Para la visualización del grafo se ha usado Linkurious , uno de los componentes más efectivos para este propósito en el mercado. Se puede interactuar con el grafo haciendo zoom, seleccionando elementos, moviendo elementos o usando el lasso tool para seleccionar varios nodos. Haciendo doble click sobre un nodo se cargan las conexiones a él que no estén visualizadas.

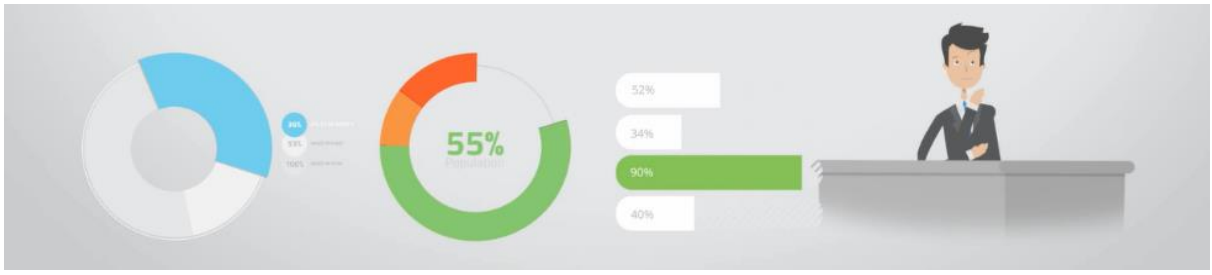
Neo4j y las Bases de Datos basadas en grafos en general tienen aplicaciones muy particulares, como Detección de Fraudes (descubriendo patrones de relaciones entre nodos), Recomendaciones en Tiempo Real (es relativamente sencillo, usando el peso de las relaciones de cada nodo, su tendencia, etc), Analítica de Redes Sociales (por la facilidad de implementar algoritmos de grafos en este tipo de Base de Datos)

Enjoy it!!

8. DESCRIPCIÓN DE STRATEBI

En Stratebi ofrecemos **gran cantidad de soluciones analíticas** por una compañía de **rápido crecimiento,** innovando en las áreas tecnológicas de mayor desarrollo en la actualidad: **Business Intelligence, Big Data y Social Intelligence**, muchas de ellas, basadas en soluciones **Open Source**.

Además, somos **Partners Certificados en Microsoft PowerBI y Vertica**, con gran número de proyectos con ámbas tecnologías



Desarrollamos nuevas soluciones analíticas basadas en Open Source, para la generación de Cuadros de Mando en tiempo real, con tecnologías IoT para SmartCities, machine learning, etc...

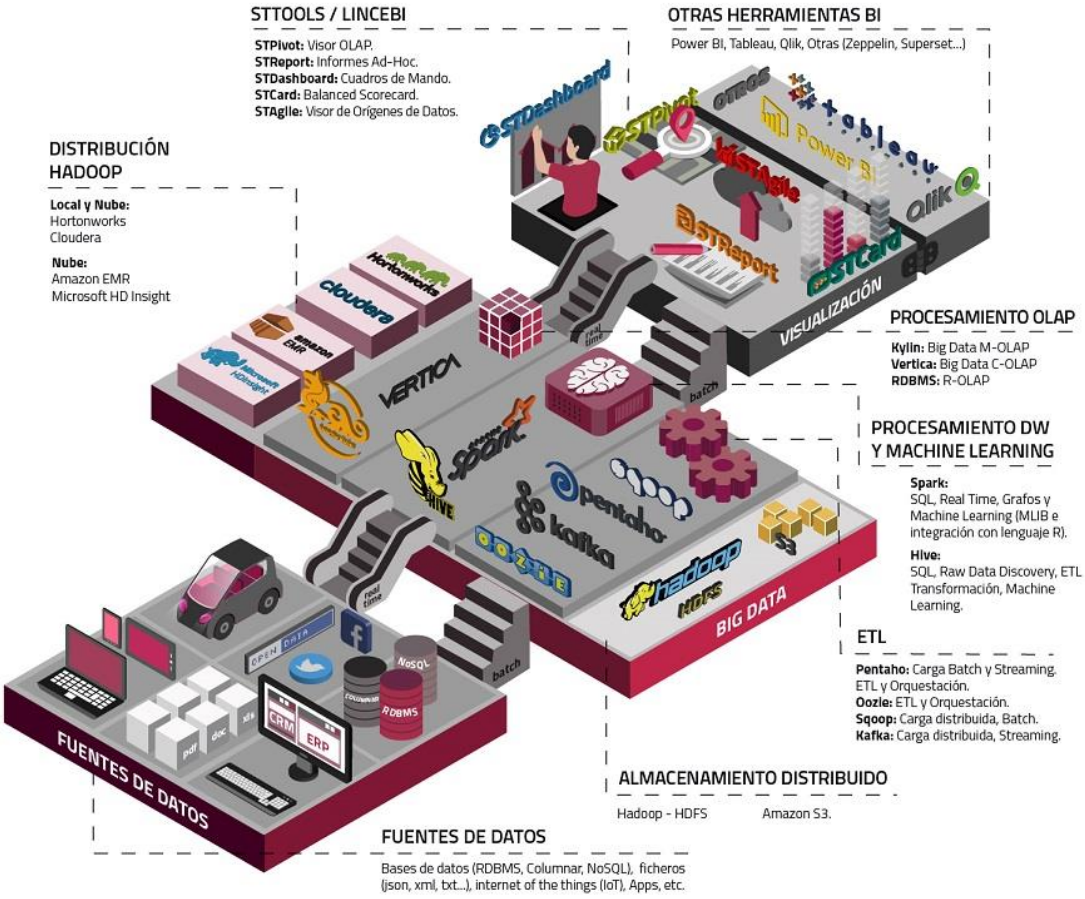
LinceBI
The best open source BI based solution

9. TECNOLOGÍAS

Recientemente, hemos sido nombrados Partners Certificados de Vertica, Talend, Microsoft, Snowflake, Kylligence, Pentaho, etc...

10. DESCRIPCIÓN DE LAS SOLUCIONES

Stratebi son los creadores de la solución LinceBI.com sobre la que podrás desarrollar e innovar.



11. INFORMACIÓN SOBRE STRATEBI

Stratebi es una empresa española, con sede en Madrid y oficinas en Barcelona, Alicante y Sevilla, creada por un grupo de profesionales con amplia experiencia en sistemas de información, soluciones tecnológicas y procesos relacionados con soluciones de Open Source y de inteligencia de Negocio.

Esta experiencia, adquirida durante la participación en proyectos estratégicos en compañías de reconocido prestigio a nivel internacional, se ha puesto a disposición de nuestros clientes a través de Stratebi.

Stratebi es la única empresa española que ha estado presente todos los Pentaho Developers celebrados en Europa (Mainz-Alemania, Barcelona, Lisboa, Roma, Amsterdam, Sintra, Amberes (2) Londres…), habiendo organizado el de Barcelona.

En Stratebi nos planteamos como **objetivo** dotar a las compañías e instituciones, de herramientas escalables y adaptadas a sus necesidades, que conformen una estrategia Business Intelligence capaz de rentabilizar la información disponible. Para ello, nos basamos en el desarrollo de soluciones de Inteligencia de Negocio, mediante tecnología Open Source.

Stratebi **son** profesores y responsables de proyectos **del Master en Business Intelligence de la Universidad UOC, UCAM, EOI…**

Los profesionales de Stratebi son los creadores y autores del primer weblog en español sobre el mundo del Business Intelligence, Data Warehouse, CRM, Dashboards, Scorecard y Open Source.

Stratebi es partner de las principales soluciones Analytics: Microsoft PowerBI, Talend, Pentaho, Vertica, Snowflake, Kyligence, Cloudera…

Todo Bi, se ha convertido en una referencia para el conocimiento y divulgación del Business Intelligence en español.

12. REFERENCIAS STRATEBI

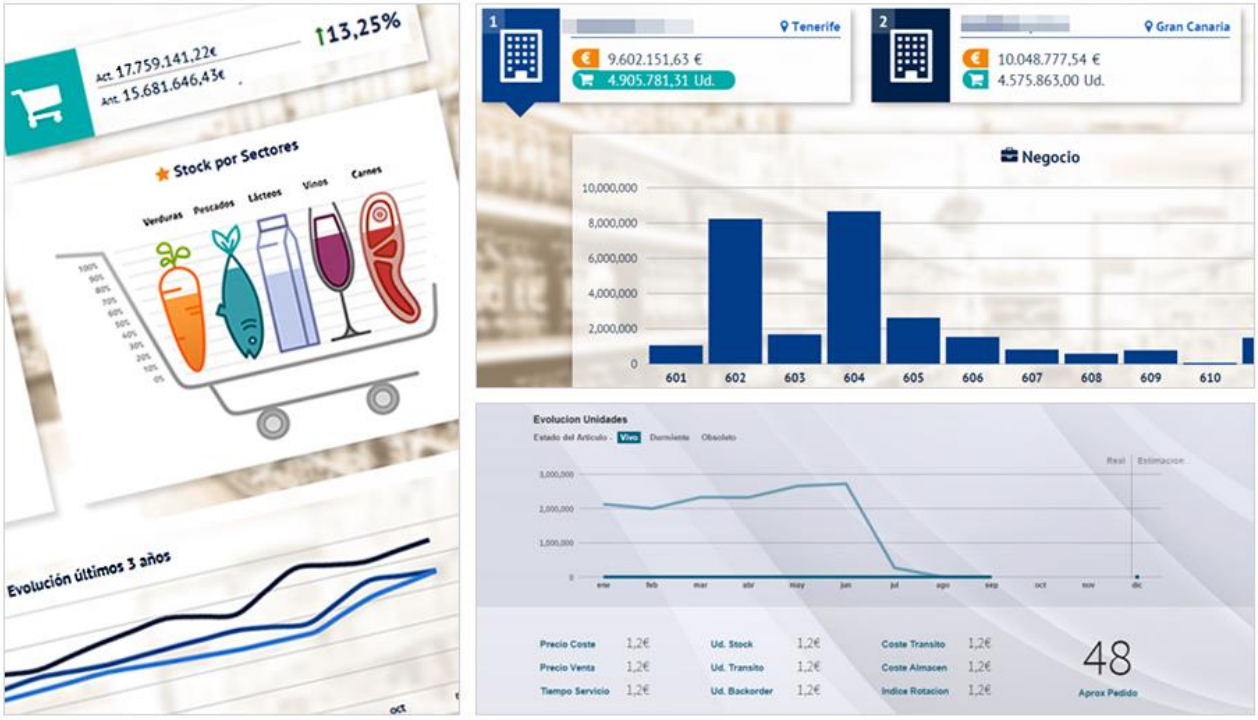
Trabajamos en los principales sectores y con algunas de las compañías y organizaciones más importantes de España.

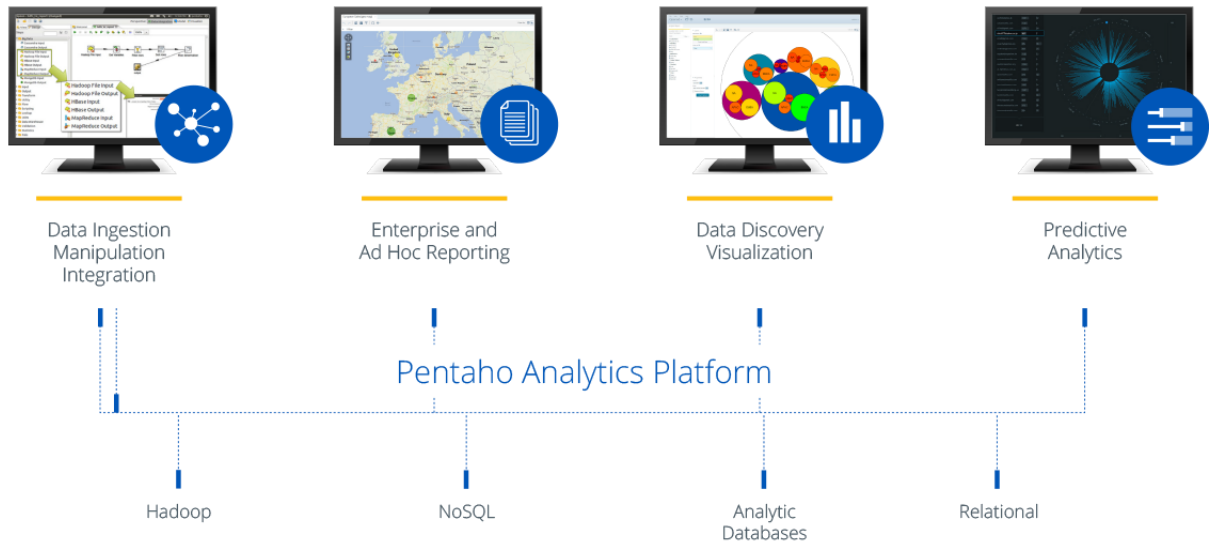
SECTOR PRIVADO

SECTOR PÚBLICO

13. EJEMPLOS DE DESARROLLOS ANALYTICS

A continuación se presentan **ejemplos de algunos screenshots** de cuadros de mando diseñados por Stratebi, con el fin de dar a conocer lo que se puede llegar a obtener, así como Demos Online en la web de Stratebi:

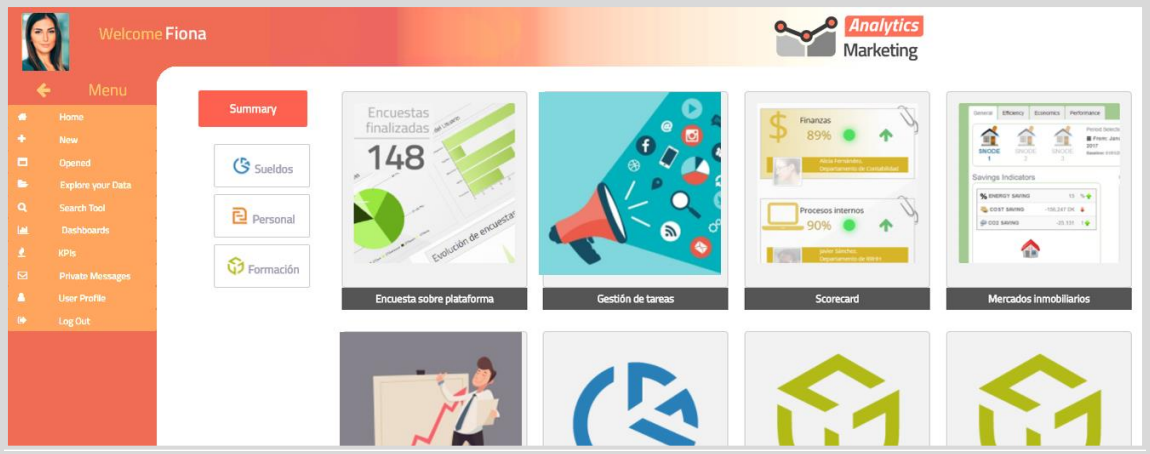





facebook DASHBOARD COMENTARIOS
Sexo
Sentimiento
Localización
65%
35%
28%
60%
12%
Tipo de recurso
Fotos
Status
Vídeos
Preguntas
Evolución Comentarios por días


smart MADRID
Distrito | Barrio
M30 | Urbano
Última Actualización
18:24
TOP 5 PUNTOS DE MEDIDA
PROXIMIDAD DE PUNTOS DE MEDIDA
EVOLUCIÓN INTENSIDAD DE TRÁFICO


Créditos Matriculados
Procedencia
FUENTE-ÁLAMO
Alumnos Matriculados: 50
Sexo
50% Masculino
50% Femenino
Rango de Edad
18-20
20-22
22-24
24-26
26-28
28-30
30-35
35-40
GDO/A EN BIOLOGÍA
Créditos Matriculados: 6.500
Buscar Plan de Estudio
INTERÉS PLATAFORMA
181 Inscritos en Cursos
8 Desinscritos en Cursos
181 Inicios de Cursos
110 Inscritos en Eventos Virtuales
63 Inscritos en Debates
181 Inscritos en Talleres
USUARIOS ACTIVOS POR PAÍS
INSCRITOS EN CURSOS POR COMUNIDAD EDU
JORNADA COMPLETA
IDIOMAS
TIC
TIPO DE CENTRO
TIPO DE AUSENCIA
Nº de profesores por sexo
36%
64%
Nº total de profesores
1.587
Media de edad
43 años
Búsqueda
TERRITORIAL: Madrid - Capital


Welcome Fiona
Analytics
Marketing
Menu
Home
New
Opened
Explore your Data
Search Tool
Dashboards
KPIs
Private Messages
User Profile
Log Out
Summary
Sueldos
Personal
Formación
Encuestas finalizadas
148
Evolución de encuestas
Encuesta sobre plataforma
Gestión de tareas
Finanzas
89%
Procesos internos
90%
Scorecard
Mercados inmobiliarios