## Método de prevención de fraude con bases de datos no-relacionales

### Abstracto

Se ejemplificará un caso de uso de bases de datos no relacionales para detectar fraude de falsificación de identidad, haciendo uso de las ventajas (en tiempo de ejecución) que tienen las bases de este tipo para encontrar relaciones entre datos.

El tipo de fraude al que me referiré es aquel en el que una persona falsifica una o varias identidades para solicitar créditos a la institución sin tener la obligación de pagarlos, y dejando a la institución con dificultades para recuperar la deuda otorgada.

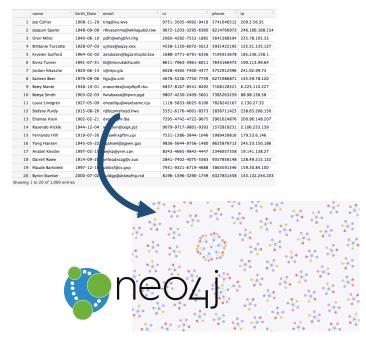
### El arte de la falsificación

La falsificación de identidad puede generar grandes pérdidas a las instituciones bancarias. Una vez hecho el fraude, no habrá ningún individuo para hacer el cobro y el banco perderá tiempo buscando un fantasma. Pero se necesita un poco de organización.

Un robo de identidad puede ser un buen punto de partida. A veces todo lo que se necesita es un documento real como un INE, estados de cuenta. Equipado con esto, un defraudador puede falsificar una primera identidad, o bien, puede inventar un nombre, edad, fecha de nacimiento, dirección, número de teléfono, etc. El objetivo es crear una identidad (falsa) que tenga todas las características de una persona real. Los estafadores se asegurarán de que su identidad pueda pasar por alto los controles de seguridad de sus objetivos. Aquí es donde tener una pieza "real", verificable, de identidad (como los estados de cuenta de una persona) es útil.

## <u>Implementación</u>

Primero generé una base de datos generados (relacional) con registros aleatoriamente (nombre, número teléfono, número de tarjeta de crédito, dirección de correo electrónico, dirección iP), y después establecí las restricciones y relaciones para migrar esta base de datos a un esquema no-relacional.

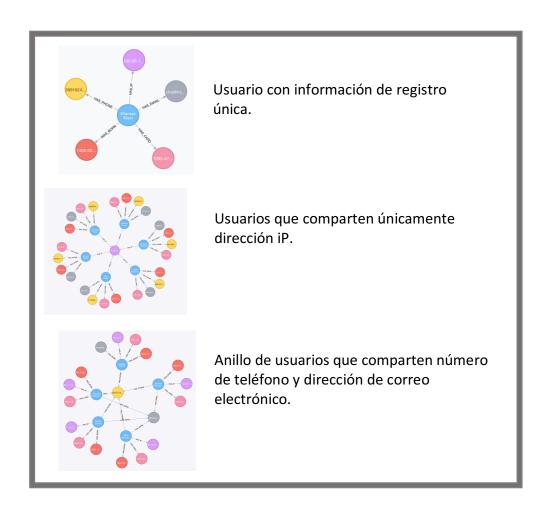


### Análisis

Por un lado puede obtenerse el máximo subgrafo (de información de registro) conexo a partir de un nuevo usuario y verificar las relaciones que pueda tener con otros usuarios, por otro lado puede hacerse una análisis más general del número y tamaño de anillos que forman usuarios que comparten información.

#### Rendimiento

La ventaja principal de la base de datos no-relacional para este caso es que las consultas o 'queries' mencionados anteriormente emplean un tiempo de ejecución mínimo y que además son tareas que pueden llegar a ser imposibles de ejecutar en un esquema relacional.



#### Las consultas efectuadas fueron:

• El máximo subgrafo (de información de registro) conexo a partir de un nuevo usuario:

• El número y tamaño de anillos que forman usuarios con información de registro duplicada (con tiempo de ejecución de 0.6 segundos aproximadamente):

```
MATCH (person:Person)-[]->(contactInformation)
WITH contactInformation,
count(person) AS RingSize
MATCH (contactInformation)<-[]-(person)
WITH collect(person.name) AS Persons,
contactInformation, RingSize
WHERE RingSize > 1
RETURN Persons AS FraudRing,
labels(contactInformation) AS ContactType,
RingSize
ORDER BY RingSize DESC
```

FraudRing	ContactType	RingSize
[Jimmie Koss, Junita Lesch, Meghann Schaefer, Harley Fay, Ted Hermiston, Elbert Paucek]	[IP]	6
[Mertie Mitchell, Werner Marquardt, Guillermo Hegmann, Lane Rippin, Lillie Mraz, Sabrina Abshire]	[IP]	6
[Necole Dare, Marion Gutmann, Deedra Franecki, Dotty Hyatt, Bryon Bruen, Lorina Berge]	[IP]	6
[Markus Glover, Edward Quigley, Lindsay Konopelski, Gigi Bailey, Fausto Huels, Leonarda Haag]	[IP]	6

# Bibliografía:

[1] CRAN, 2016, "Package 'RNeo4j', vía internet: <a href="https://cran.r-project.org/web/packages/RNeo4j/RNeo4j.pdf">https://cran.r-project.org/web/packages/RNeo4j/RNeo4j.pdf</a>

[2] Linkurious, 2016, "How to detect bank loan fraud with graphs: part 2", vía Internet: <a href="https://linkurio.us/how-to-detect-bank-loan-fraud-with-graphs-part-2/">https://linkurio.us/how-to-detect-bank-loan-fraud-with-graphs-part-2/</a>

[3] Nicole White, 2016, "Demo of RNeo4j part 1", vía internet: <a href="https://nicolewhite.github.io/2014/05/30/demo-of-rneo4j-part1.html">https://nicolewhite.github.io/2014/05/30/demo-of-rneo4j-part1.html</a>