Workshop Spatial, Graph y Machine Learning en base de Datos Oracle

HOL3 Graph

Contenidos

Requerimientos iniciales

Para la realización de este workshop se necesita un cliente de *Remote Desktop* de Windows.

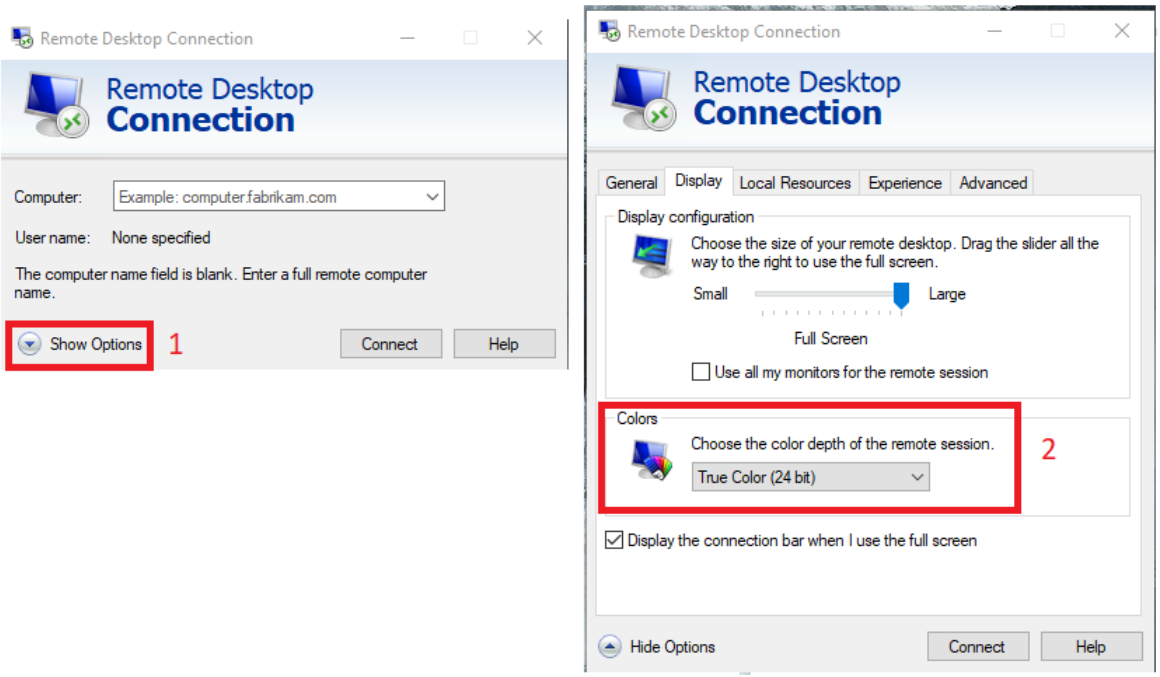
Este cliente está instalado por defecto en el sistema operativo Microsoft Windows, existiendo también clientes compatibles para MacOs y Linux.

La máquina del workshop se proporciona para uso individual de cada participante del workshop siendo para uso exclusivo del mismo.

Esta máquina está alojada en la nube pública **Oracle Cloud Infrastructure** (OCI) estando prohibida la reproducción o alteración de sus contenidos fuera de lo previsto en este manual de usuario.

Escritorio Remoto con Microsoft Windows

En la configuración del cliente es importante especificar el uso de *True Color (24 bit)* para evitar problemas con algunas herramientas. Desde el botón *Show Options*, como se muestra a continuación:



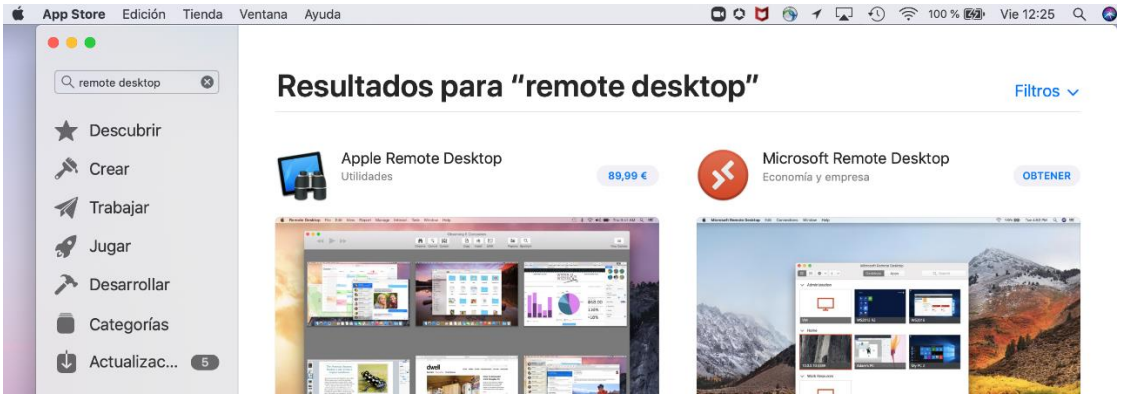
Como parte de la documentación del workshop se facilitarán los siguientes datos:

- Nombre de usuario y password
- IP pública de la maquina del workshop

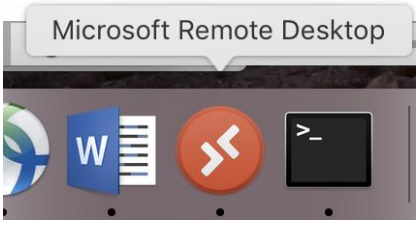
Escritorio Remoto con MacOS

En MacOs la aplicación para conectar a escritorio remoto de Windows no viene instalada por defecto, pero está disponible en el App Store de manera gratuita.

Buscando “remote desktop” se encuentra como “*Microsoft Remote Desktop*” tal y como se muestra a en la siguiente captura:



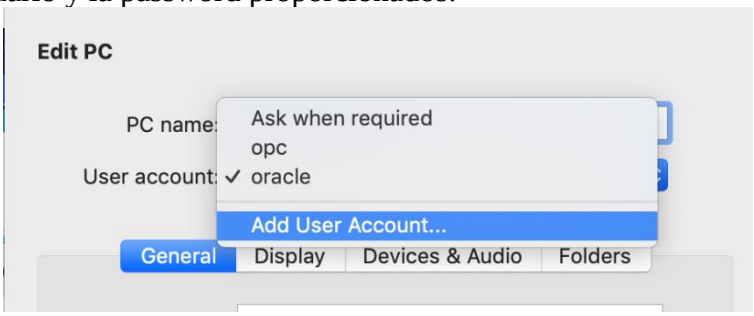
Una vez instalada esta aplicación, aparecerá un icono como el siguiente en la barra de aplicaciones para poder realizar las conexiones.

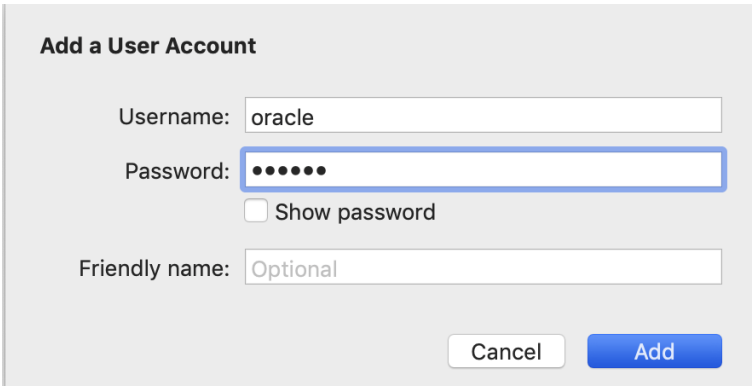


Como parte de la documentación del workshop se facilitarán los siguientes datos:

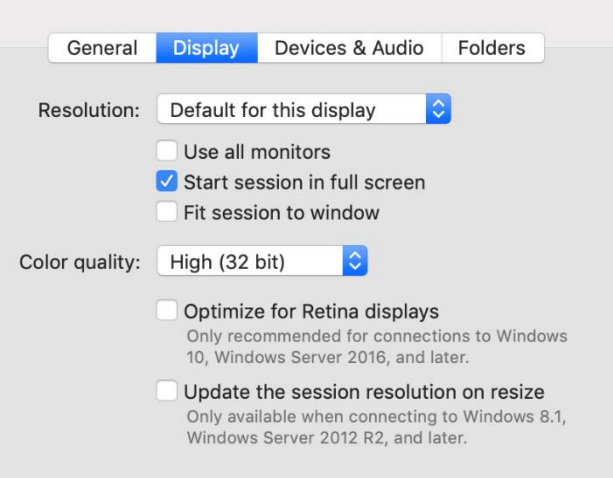
- Nombre de usuario y password
- IP pública de la maquina del workshop

Con los cuales se configura como se muestra a continuación.
Añadimos el usuario y la password proporcionados:

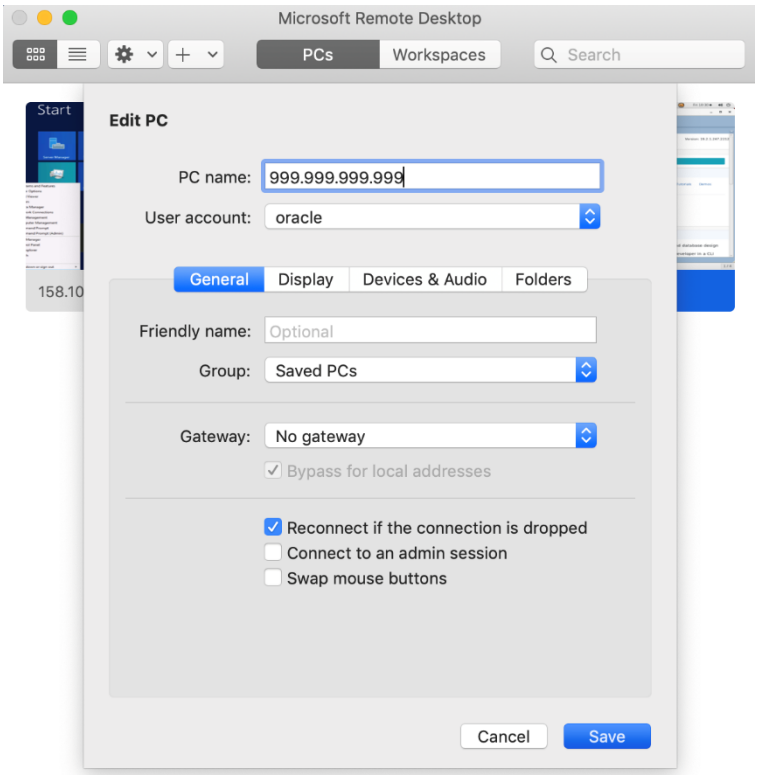




En la pestaña de Display se confirma que la calidad de color está seleccionada a 32 bits:



Se introduce la IP pública en '*PC name*", se guarda el acceso con el botón S*ave* y ya está preparado el acceso a la máquina virtual del workshop.

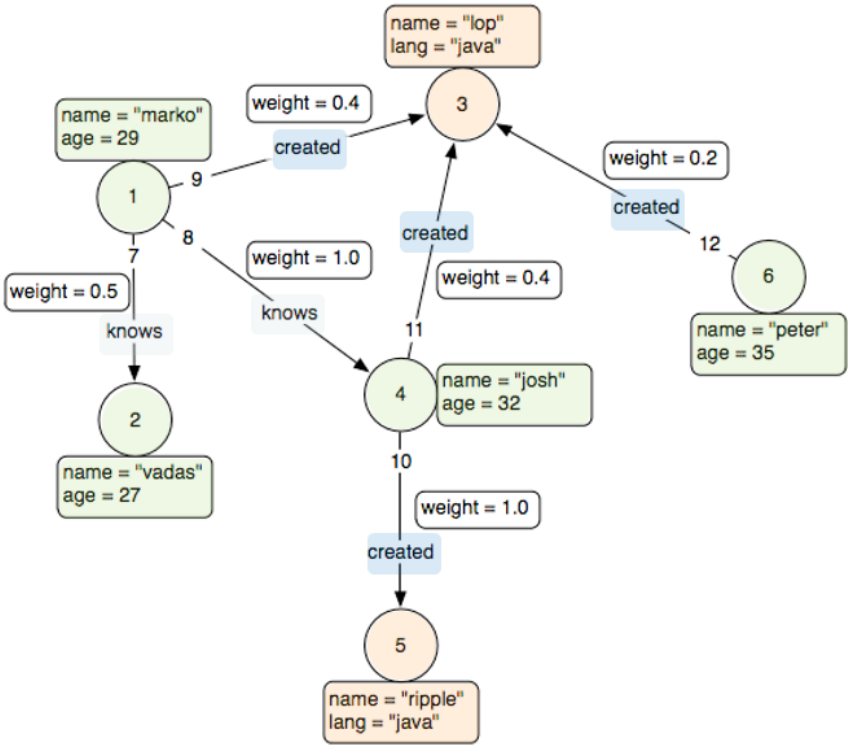


Property Graph (2 horas)

Analítica de Property Graph

La analítica de *property graph* permite encontrar información que está en las relaciones directas e indirectas entre los elementos de los datos.

En este tipo de analítica el modelo de datos representa las entidades como vértices y sus relaciones como aristas. Cada una de ellas puede opcionalmente tener varios atributos como muestra el siguiente ejemplo sencillo:



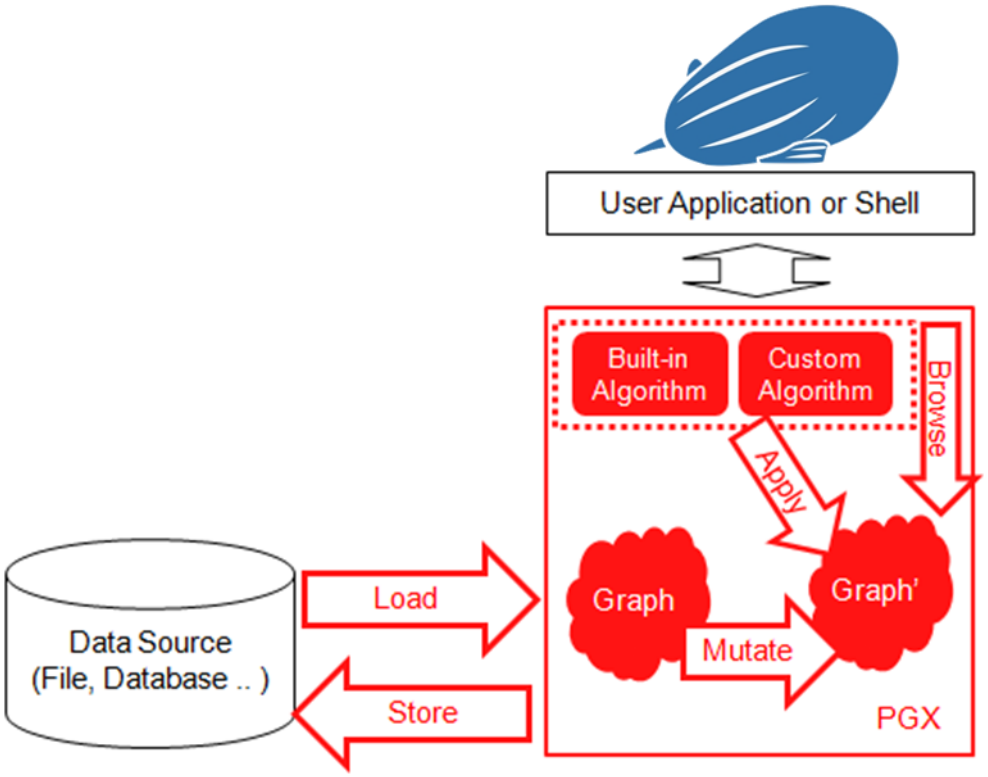
Descripción del entorno del workshop

Las actividades con *Property Graph* de este workshop se realizan con *Apache Zeppelin* como IDE.

Las distintas piezas implicadas en las actividades del workshop son:

- **In-memory PGX engine** o motor de analítica de *property graph*
- **Apache Zeppelin** como interfaz de usuario
- Fuentes de datos como **Oracle Database** o ficheros planos

A continuación, se muestra un diagrama con estos elementos a modo de referencia:



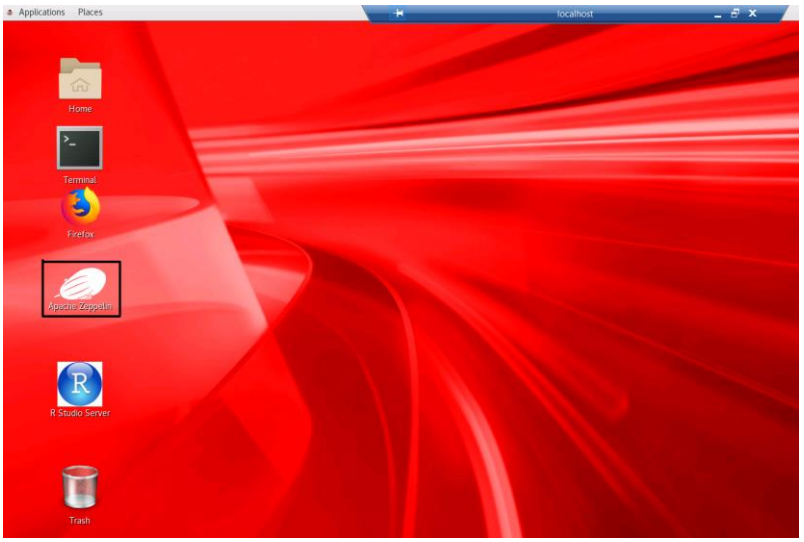
Para interactuar con el motor analítico PGX, Zeppelin dispone de un intérprete PGX preinstalado y configurado que se usará para enviar los diferentes comandos.
El entorno se encuentra ya instalado y configurado para empezar con las actividades.

Instrucciones del workshop

Siga las instrucciones proporcionadas para acceder al escritorio remoto. Se le proporcionará una dirección IP y un usuario/clave.

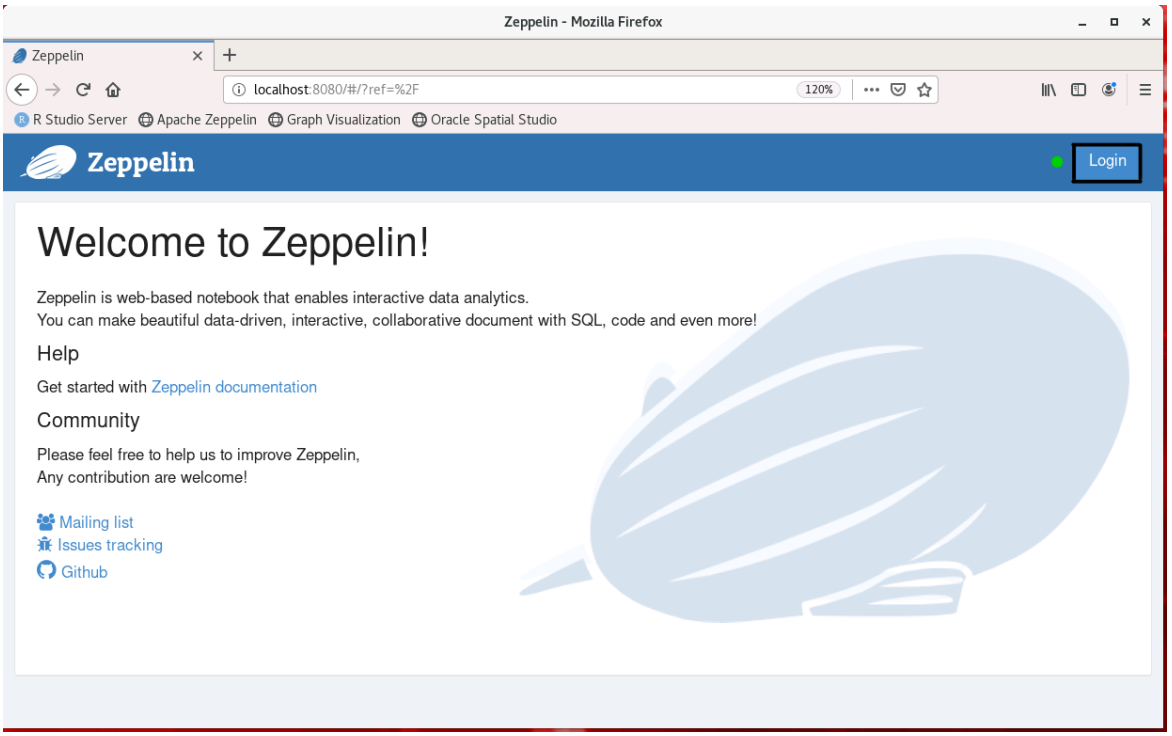
Una vez se haya accedido al servidor con el cliente de *Remote Desktop*, en el escritorio hay un acceso directo a *Zeppelin*.

Haga doble click para abrir un navegador que directamente abrirá la dirección de *Zeppelin*.

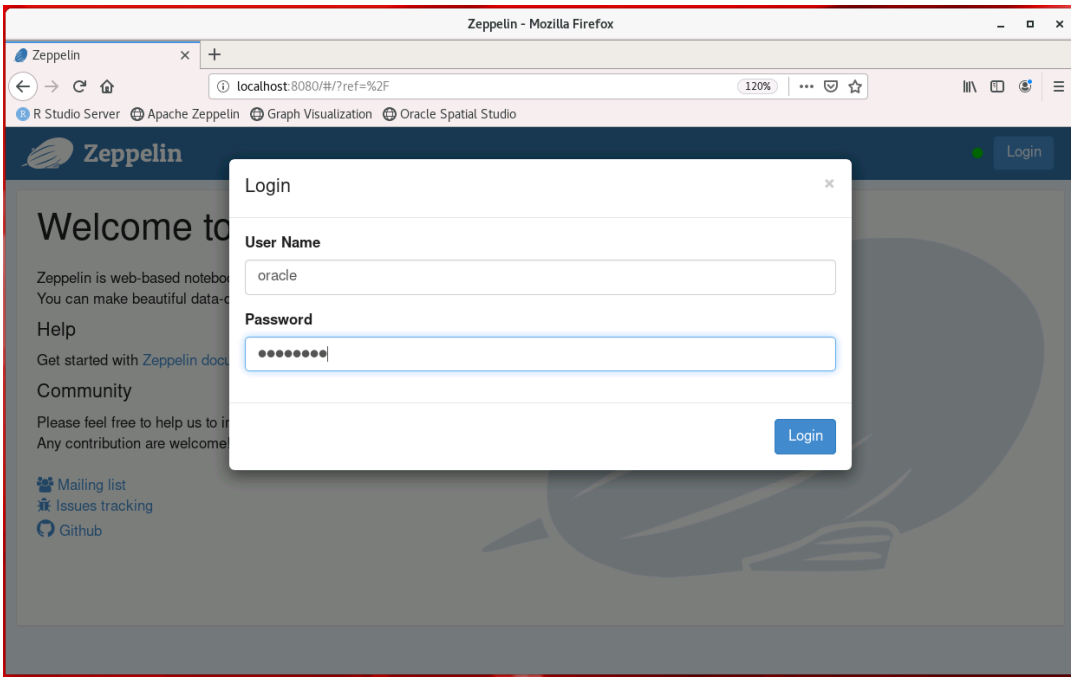


En la ventana del navegador aparecerá la pantalla de bienvenida a *Zeppelin*, tal y como se muestra a continuación.

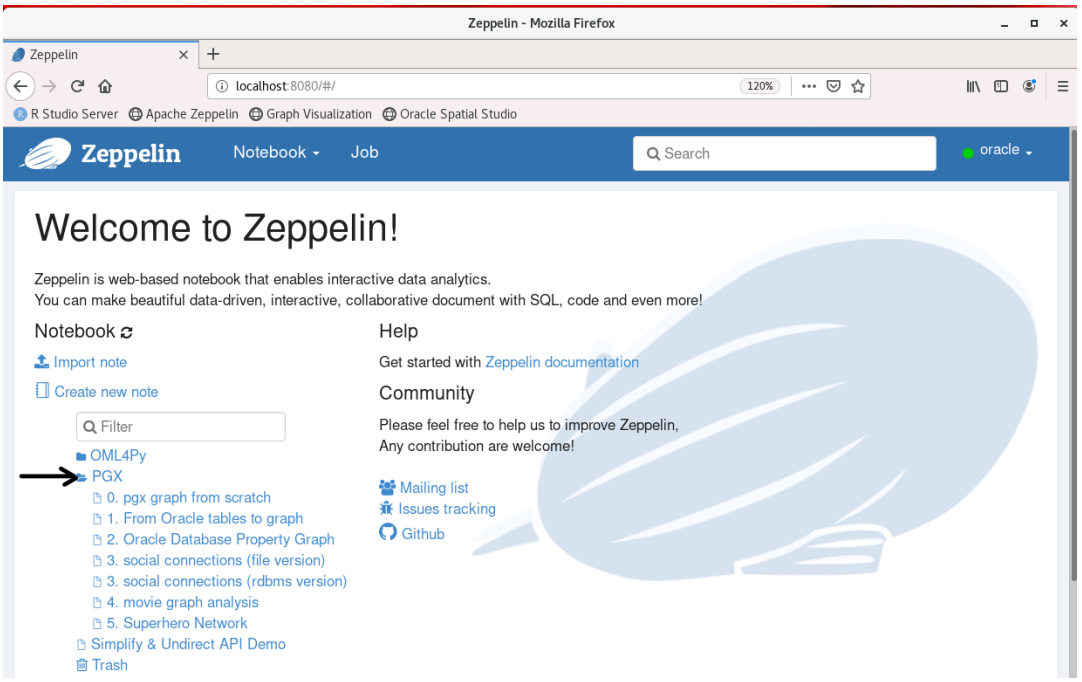
Haga click en el botón de *Login* de la esquina superior derecha.



A continuación, introduzca el usuario y clave que se le haya proporcionado para autenticarse en Zeppelin:

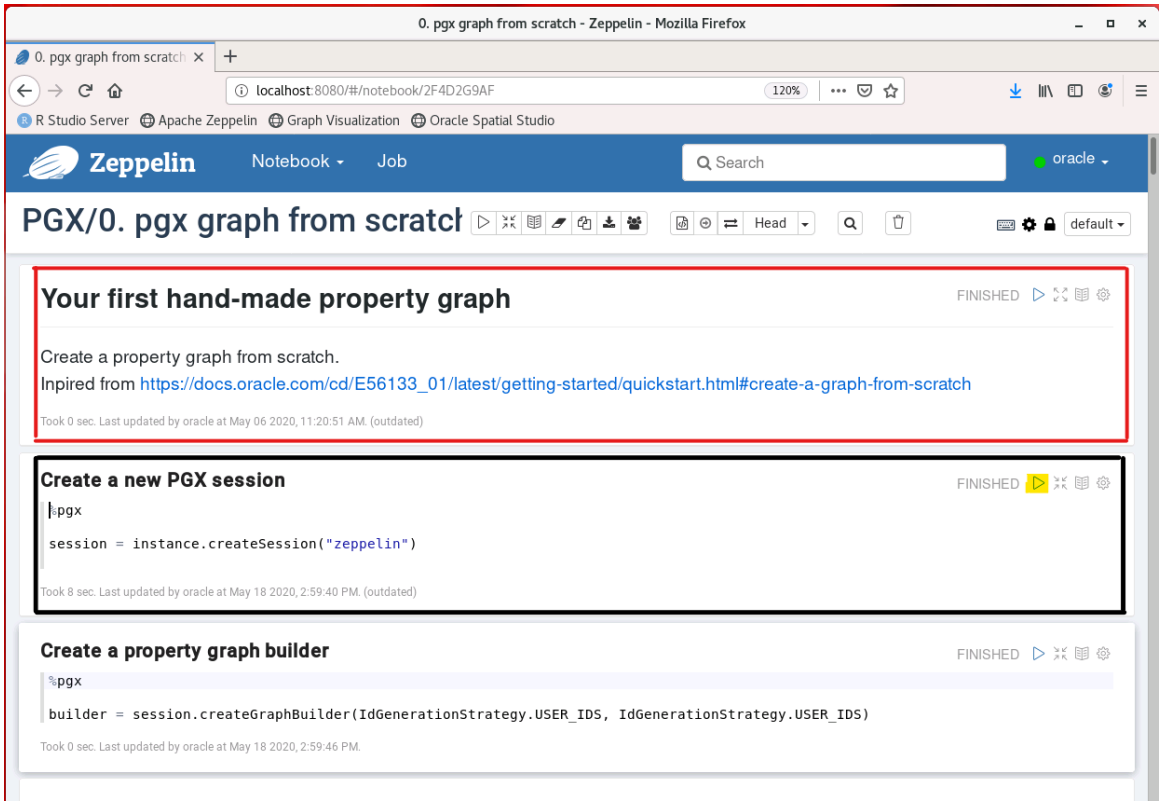


Una vez dentro de *Zeppelin*, en el área de Notebook, despliegue la carpeta **PGX** donde se encuentran los notebooks necesarios para el workshop. Haciendo click sobre el nombre del notebook, Zeppelin lo abre en la misma pestaña del navegador.



Se puede observar que el notebook está divido en secciones llamadas *párrafos* que tienen un *intérprete* concreto para ejecutar el código que contienen.

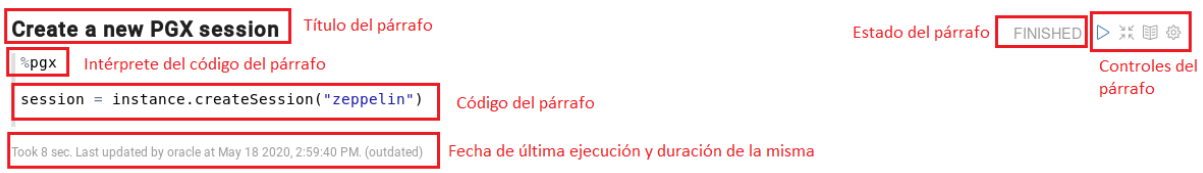
La captura siguiente destaca en rojo un párrafo de *Markdown* que contiene únicamente información y otro párrafo rodeado en negro que contienen el código que se envía al motor PGX. Este segundo tipo de párrafo es el que se irá ejecutando durante el workshop.



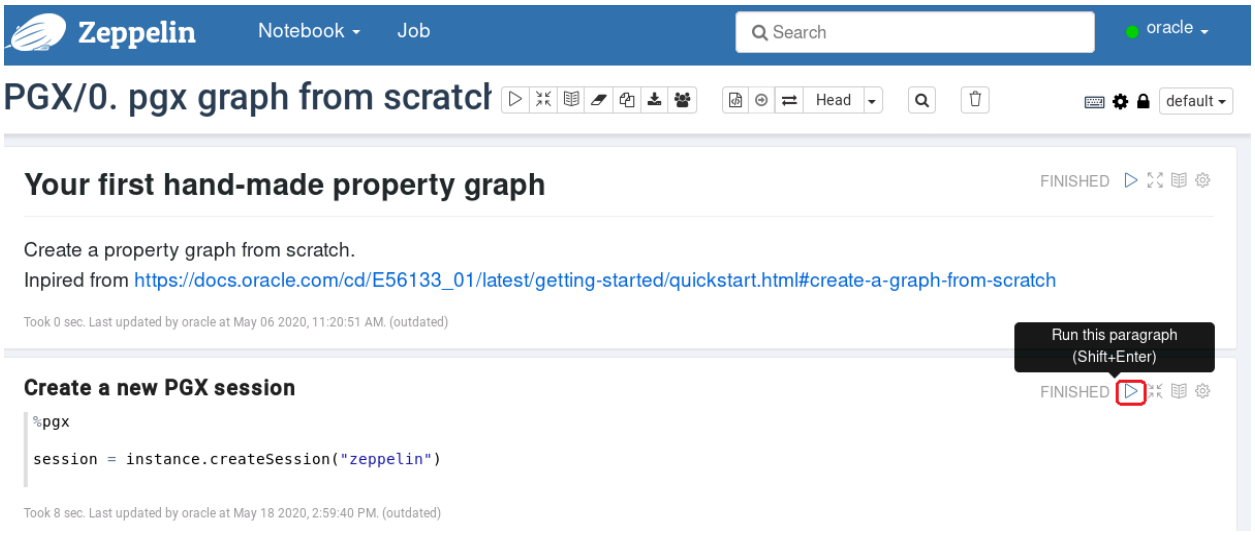
La primera línea del párrafo rodeado en negro contiene el texto '*%pgx*' que indica que será procesada por el intérprete *PGX* ; este párrafo tiene en la esquina superior derecha una barra de botones donde está el botón de ejecución (marcado en amarillo).

Para completar el workshop hay que leer los párrafos explicativos (Markdown) para posteriormente revisar y ejecutar los párrafos de *property graph* (pgx) para que se realicen las diferentes acciones en el motor PGX. No es necesario escribir código nuevo o alterar el existente, aunque es posible hacerlo.

En el siguiente diagrama se detallan las diferentes partes de un párrafo PGX:

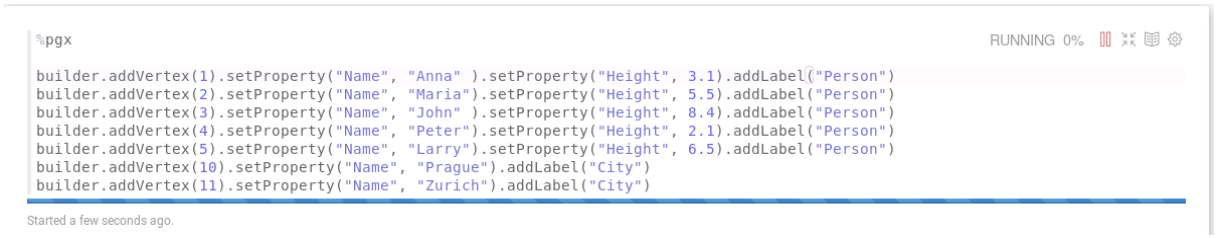


Para ejecutar el párrafo de código PGX, haga click con el ratón en el botón de forma triangular o ‘Play’:



También puede poner el foco en el párrafo y pulsar simultáneamente *Shift + Enter*.

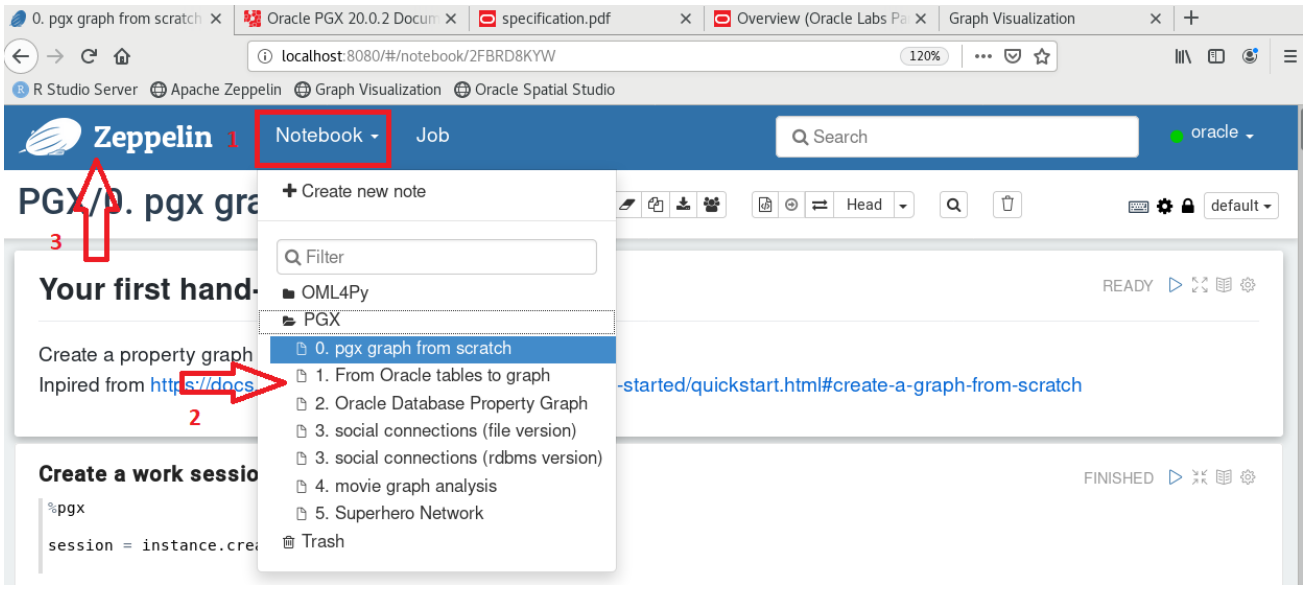
Cuando un párrafo está ejecutándose el estado cambiará a *RUNNING*, como se muestra a continuación:



Y aparecerá una barra de progreso en la parte inferior del párrafo. Espere a que un párrafo termine su ejecución antes de ejecutar el siguiente.

El orden de ejecución de los párrafos es importante, ya que en cada párrafo se construyen objetos, variables y estructuras que son necesarios para los párrafos que van a continuación. Si los párrafos se ejecutan de manera desordenada se producirán errores.

Para cambiar de un notebook a otro, en la parte superior despliegue en el menú *Notebook* (1) y escoja el notebook a abrir en Zeppelin (2) como se muestra en la siguiente captura. Puede cambiar de un notebook a otro en cualquier momento. Si quiere volver a la página de bienvenida de Zeppelin, puse sobre el icono de la esquina superior izquierda (3).



Contenido de los notebooks

A continuación, se muestran los contenidos y objetivos de cada uno de los notebooks.

0. pgx graph from scratch

El objetivo de este primer notebook es construir un *property graph* muy sencillo desde cero para familiarizarse con el uso de Zeppelin y los diferentes objetos implicados en este proceso.

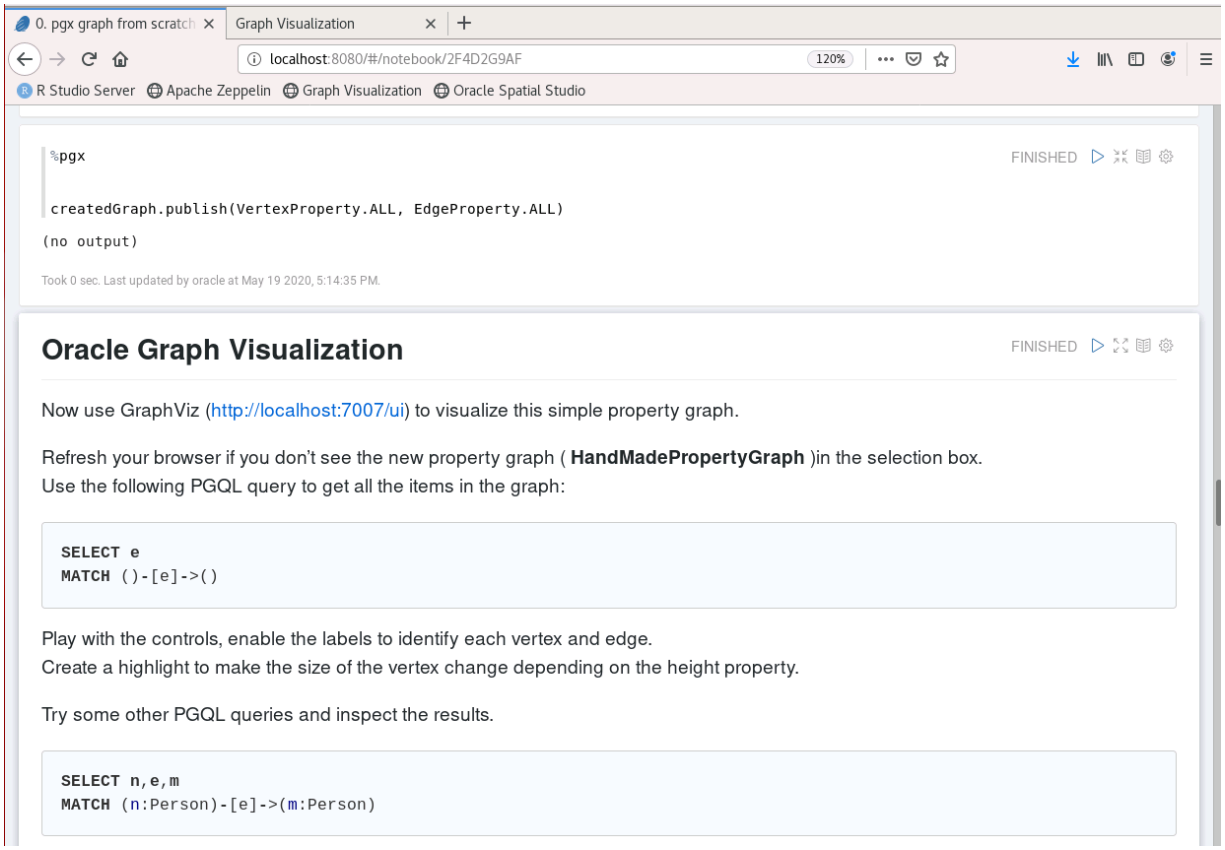
En este notebook se introducen los objetos básicos de trabajo con PGX:

- instance
- session
- graph
- analyst

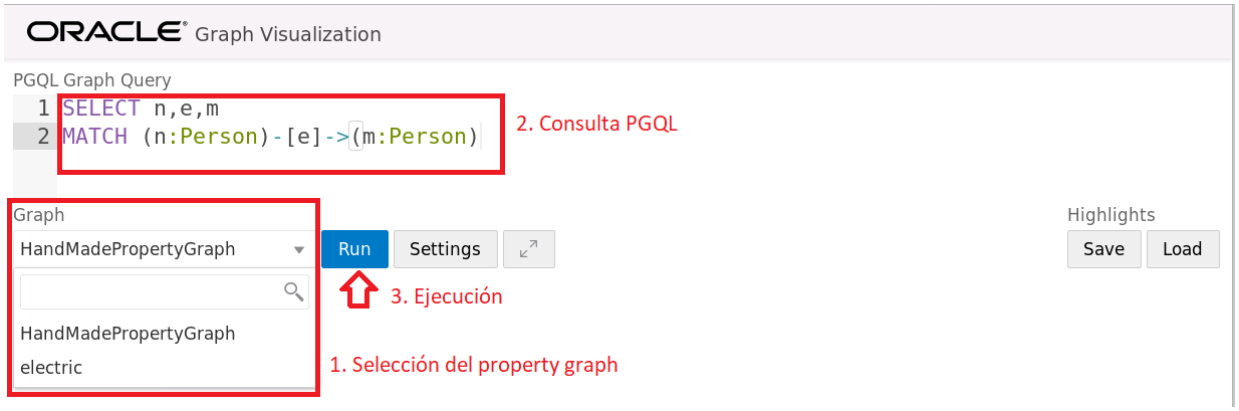
Con ellos se realiza la analítica de *property graphs* por medio del leguaje PGQL y el conjunto de algoritmos incluidos en PGX.

También se introduce la herramienta de visualización *Graph Visualization*.

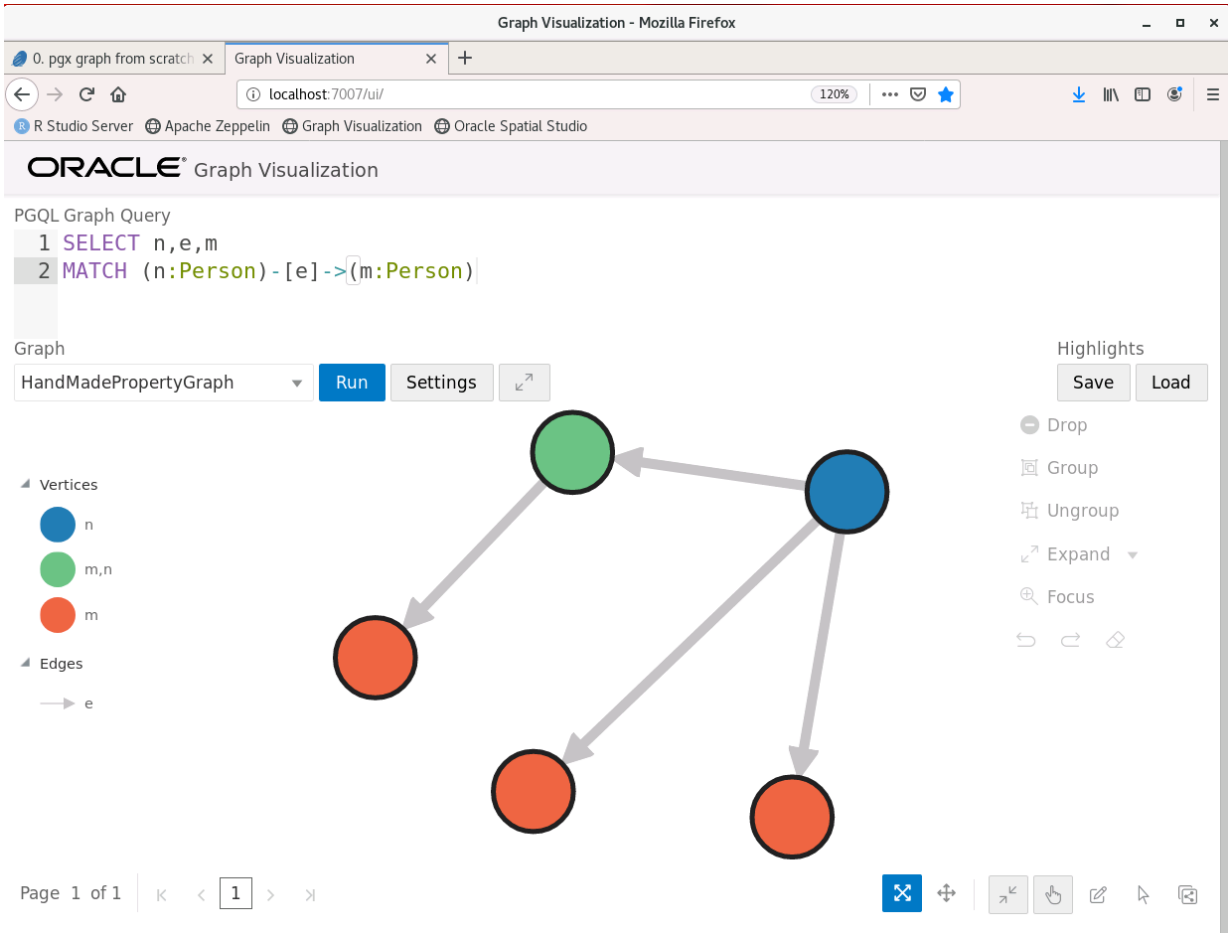
Ejecute el notebook hasta llegar a esta sección:



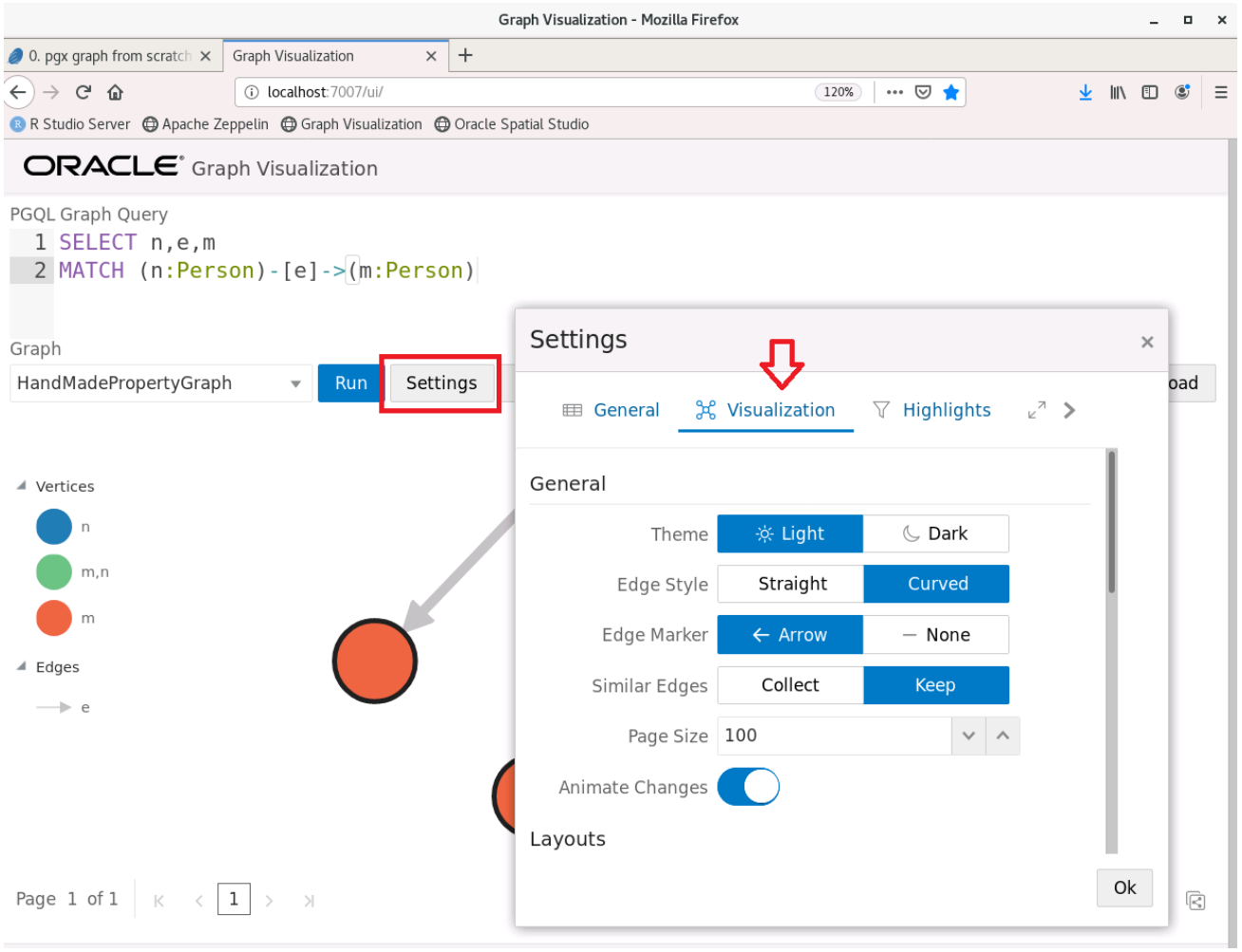
Una vez abierta la URL de Graph Visualization (http://localhost:7007/ui/), se selecciona el *property graph* creado en el notebook y que se llama *HandMadePropertyGraph* (refrescar el navegador si no aparece).
Se introduce una consulta PGQL para obtener aquella información que se desee visualizar (en el notebook se proporcionan varias) y se pulsa el botón *Run*, como se muestra a continuación:



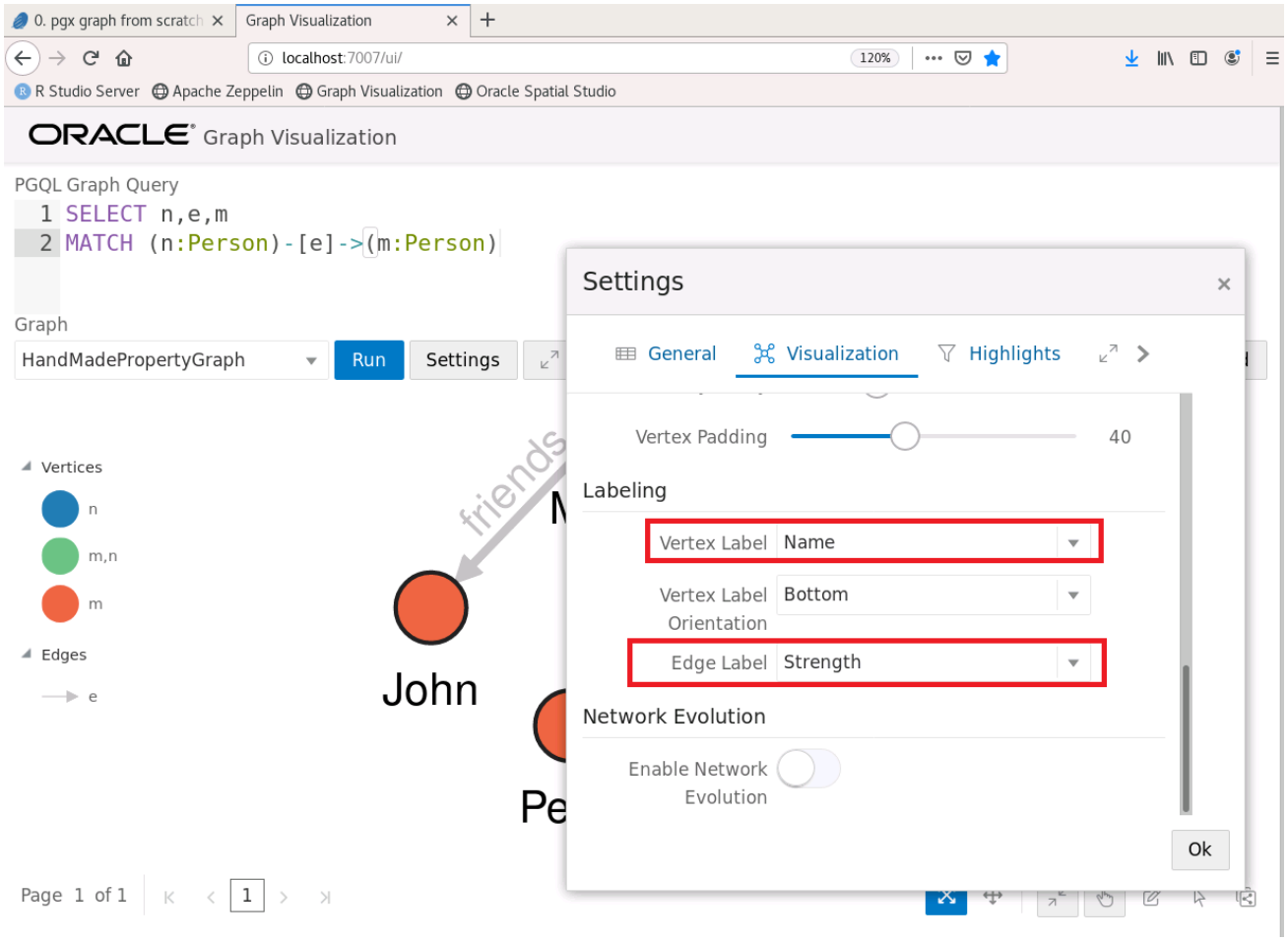
Aparecerá la visualización con sus nodos y aristas. Inicialmente sólo la estructura como se muestra en la siguiente captura:



Se pueden posicionar los nodos (círculos de colores) de la manera que se desee arrastrándolos con el ratón.
Para incluir más información, se hace click en el botón *Settings* y en el menú que aparece se selecciona la pestaña *Visualization* como se muestra a continuación:



Desde este menú se puede especificar la etiqueta (*Label*) de los vértices o aristas que se desea mostrar en la visualización. Esta etiqueta puede ser una cualquiera de las propiedades que tienen los vértices y aristas.

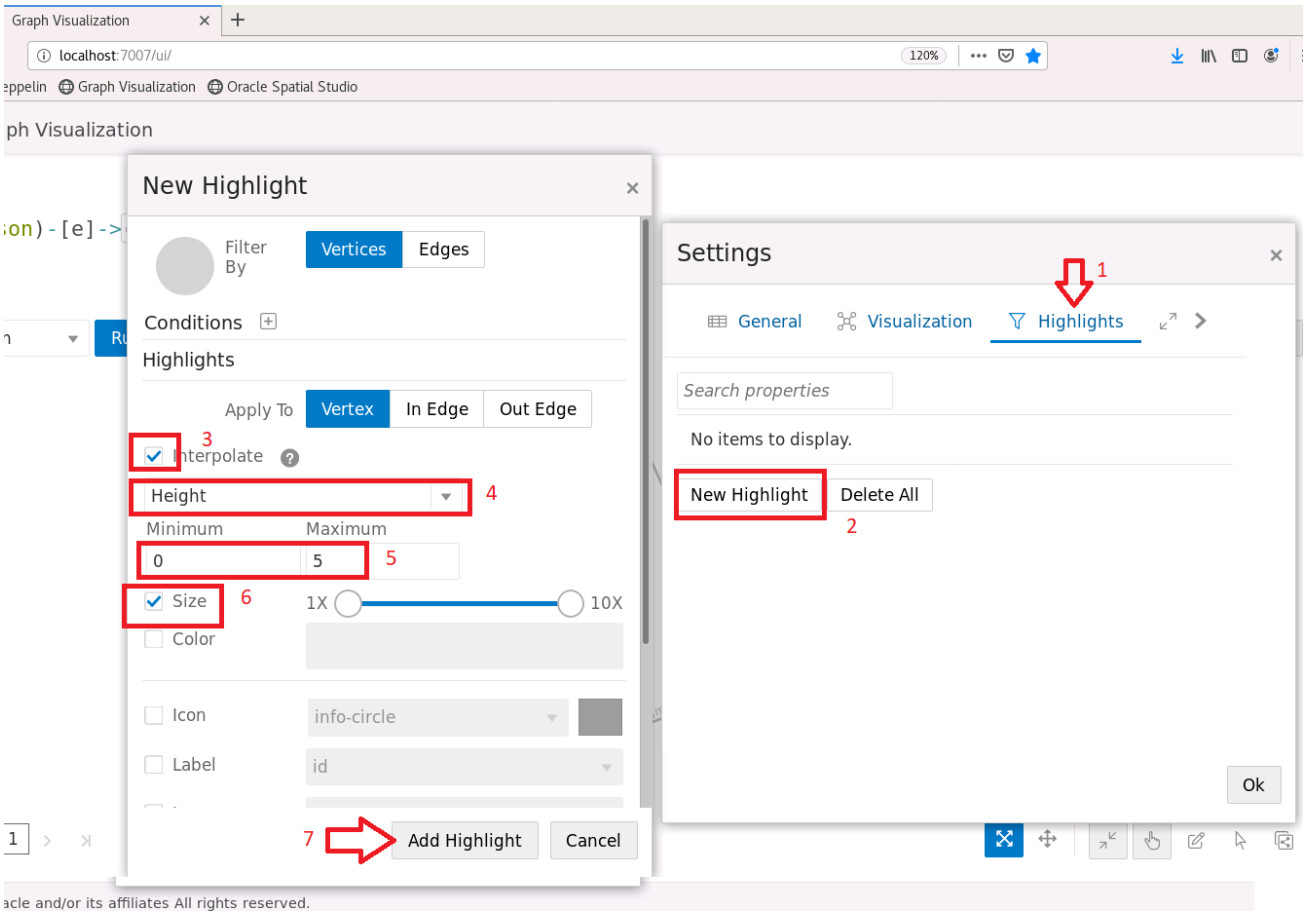


Para añadir más detalles en la visualización, en la pestaña *Highlights* de este mismo menú *Settings*, se selecciona *New Highlight* tal y como se muestra a continuación (Pasos 1 y 2). Aparece el menú *New Highlight* que se muestra en la captura.

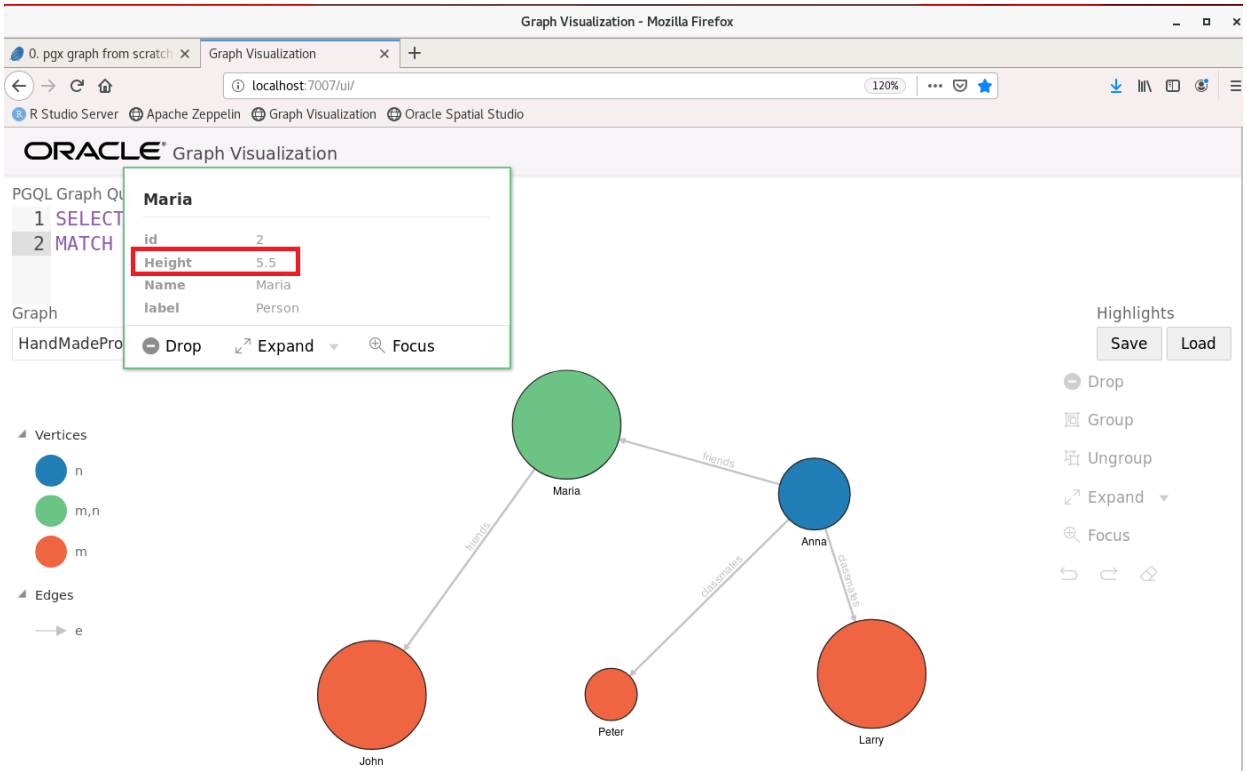
El objetivo del *Hightlight* que se va a crear es usar el valor de la propiedad *Height* de los vértices para determinar el radio de este. Esta propiedad tiene un valor numérico, por lo que se adapta perfectamente para esta función.

Se selecciona *Interpolate* (paso 3), se selecciona la propiedad *Height* en el paso 4 (aparecen las propiedades de los vértices puesto que está seleccionado *Apply to Vertex* por defecto).

Es necesario introducir unos valores mínimo y máximo para esta propiedad que se usa en la interpolación (paso 5) y por último se asocia al tamaño del vértice o *Size* (paso 6), tras lo cual se completa el proceso pulsando el botón *Add Highlight* y cerrando el menú *Settings*.



El resultado final del proceso se muestra en la siguiente captura, donde se puede apreciar que los vértices tienen ahora diferentes tamaños.
Se puede inspeccionar el valor de las propiedades pulsando el botón derecho del ratón sobre los vértices o las aristas.
De esta manera se puede observar el valor de la propiedad *Height* en los vértices y cómo aquellos más grandes tienen valores mayores.



1. From Oracle tables to graph

En este notebook se describe cómo convertir un modelo relacional de tablas en un esquema de base de datos Oracle en un *property graph*, que queda almacenado en tablas en ese mismo esquema de la base de datos.

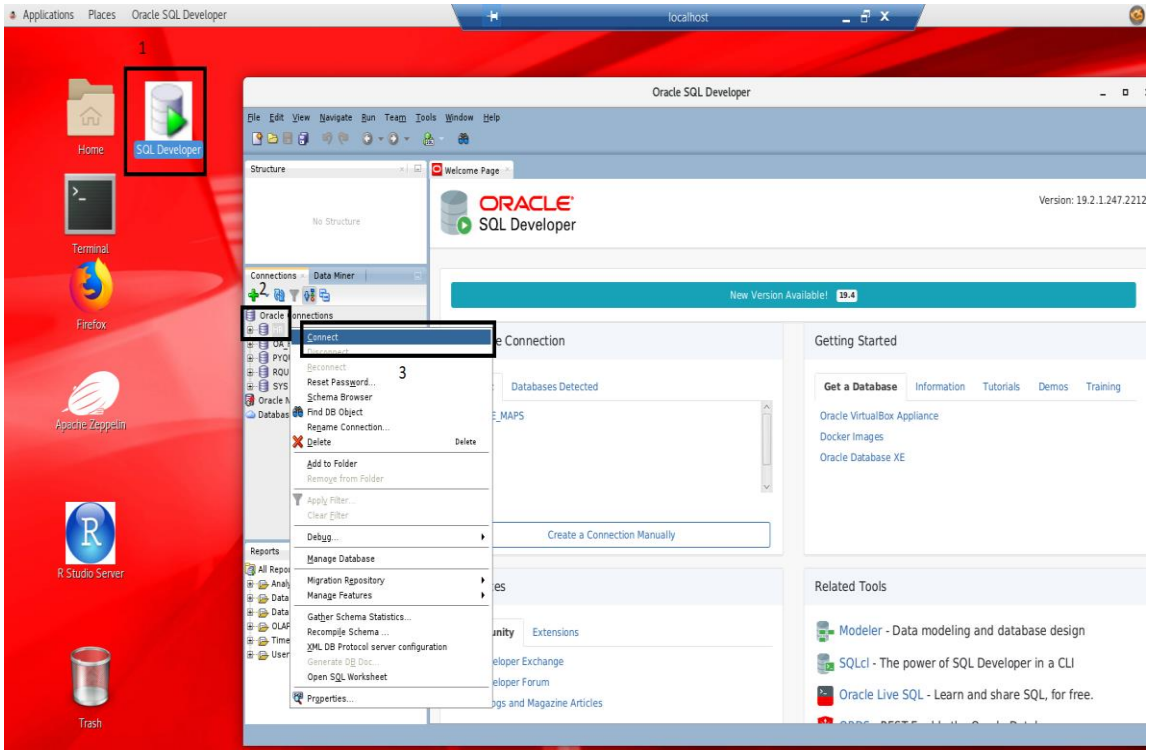
Esta conversión se hace mediante lenguaje PGQL que se invoca desde una sesión de trabajo del motor PGX, donde previamente se construye una conexión al esquema de base de datos donde están las tablas con las que se va a realizar.

Para este ejercicio se usa el esquema HR que forma parte de los esquemas de ejemplo habitualmente presentes en las bases de datos Oracle.

El objetivo es demostrar con un ejemplo cómo se puede hacer la conversión de datos en un modelo relacional en un *property graph* con la seguridad de la base de datos Oracle, ya que son necesarias credenciales de autenticación para acceder a la información.

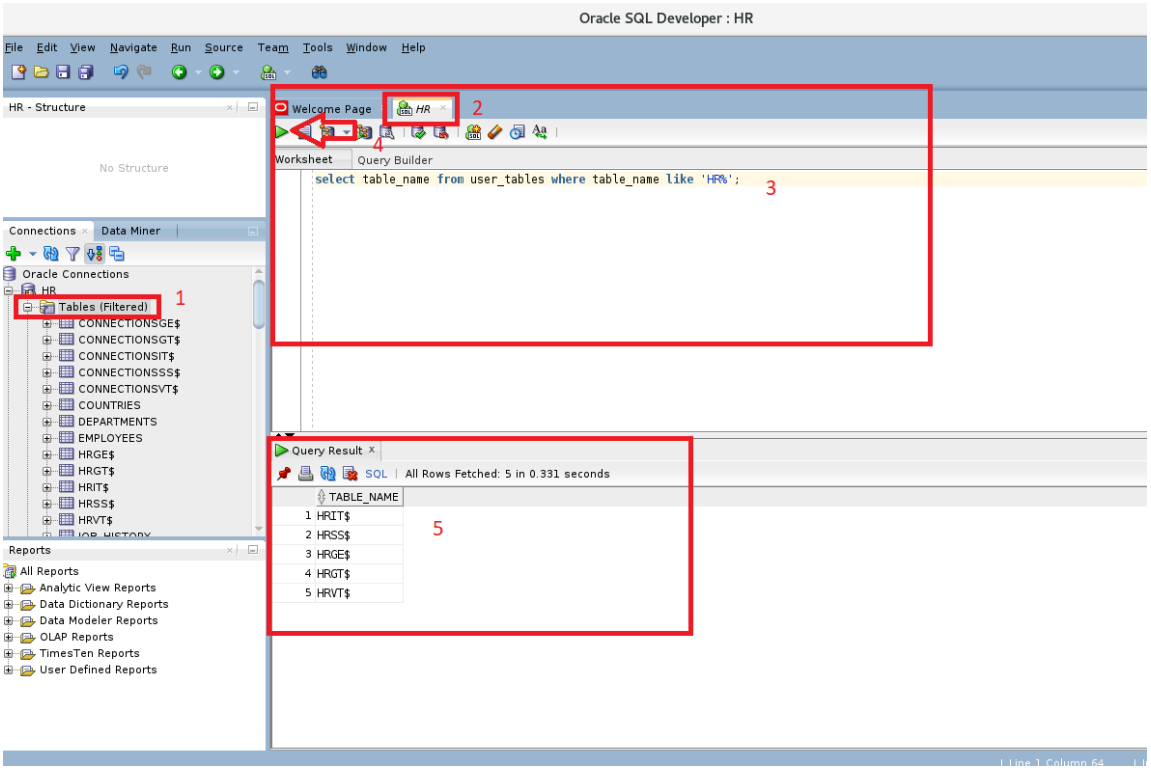
Para inspeccionar el esquema HR original, así como las tablas que se generan tras la creación del *property graph* se puede usar SQL*Developer desde el acceso directo del escritorio.

En la pestaña *Connections*, seleccione HR y con el botón derecho *Connect*.



Todos los datos para la conexión, incluyendo la clave, están ya preconfigurados.

Desplegando el nodo *Tables* (1) puede ver la lista de tablas.
En la hoja de trabajo (3) correspondiente a HR (2), puede escribir las sentencias SQL que quiera ejecutar en este esquema. Al pulsar el botón de ejecución (4) se envían al motor de la base de datos. Los resultados aparecen en el panel inferior (5).



2. Oracle Database Property Graph

Una vez completado el notebook anterior y creado el *property graph* a partir de tablas, se carga el mismo desde la base de datos Oracle al motor *in-memory PGX*.

Esto paso es necesario para poder trabajar con el *property graph* en el motor PGX, donde se van a hacer algunas de las tareas habituales de esta analítica como:

- inspeccionar las características del *property graph*
- realizar consultas PGQL
- visualizaciones con *Graph Visualization*
- analítica con algunos algoritmos como *PageRank*

Se analizan varios aspectos del *property graph* para comprobar que la información es correcta y observar cómo ha ocurrido la transformación desde el modelo basado en tablas al modelo de *property graph*.

El acceso al *property graph* está protegido por los mecanismos de autorización de la base de datos Oracle, es por ello que se vuelve a crear una conexión desde PGX al esquema de base de datos HR.

El resultado de las consultas PGQL se puede en muchas ocasiones mostrar en forma de tabla o como una visualización. En otras ocasiones el resultado es un subconjunto del *property graph*.

Por último, se persiste el *property graph* en un fichero de disco. Esta es una forma alternativa de guardarlo para una sesión de trabajo posterior.

3. social connections (file version) / 3. social connections (rdbms version)

Estos dos notebooks trabajan sobre el mismo juego de datos.

El primero lo carga a partir de ficheros planos almacenados en el sistema de ficheros del propio servidor, mientras que el segundo lo carga a partir de tablas de base de datos usando las credenciales de acceso necesarias. En los primeros párrafos se pueden observar las diferencias entre los dos métodos de carga.

Las consultas PGQL, visualizaciones y algoritmos que se ejecutan son las mismas en los dos notebooks.

Los datos analizados son de una red social, un tipo de información que se adapta muy bien a la analítica de *property graph*; contienen relaciones entre personajes famosos y algunas organizaciones internacionales.
Con a la analítica de *property graph* se identifican aquellos nodos más influyentes, enemigos comunes, etc

4. movie graph analysis

Este notebook usa un juego de datos relativo a alquileres de películas. Contiene información de películas, clientes que las alquilaron y recomendaciones hechas por los clientes.

Los datos se cargan en esta ocasión a partir de ficheros planos.

Se analizan clientes con gustos comunes, recomendaciones que pueden hacerse a clientes en función de las que hicieron otros con gustos similares, etc

5. Superhero Network

Este notebook analiza relaciones entre los personajes de los cómics de Marvel en busca que aquellos que sean más relevantes y los que actúan de conectores entre las diferentes historias.

El *property graph* analizado tiene más de medio millón de aristas o conexiones.

Resumen

¡Enhorabuena! Ha conseguido llegar al final de las actividades de este workshop.

En este workshop usted ha conseguido los siguientes retos:

- familiarizarse con Zeppelin como interfaz de usuario para trabajar con el motor analítico PGX
- conocer los elementos de trabajo de PGX: session, graph, PGQLy analyst.
- transformar un modelo de datos relacional basado en tablas de Base de Datos Oracle en un modelo de property graph basado en vértices y aristas.
- realizar analítica de property graph usando lenguaje PGQL y los algoritmos disponibles en el analyst.
- realizar visualizaciones con la herramienta GraphViz incluida.

Más información

Puede consultar la documentación oficial en:
https://docs.oracle.com/cd/E56133_01/latest/index.html

Puede encontrar más ejemplos y casos de uso en el siguiente repositorio de github:
https://github.com/oracle/pgx-samples