



Diseño de Ecosistemas para Cloud Computing y Big Data

Master Universitario en Ingeniería informática

Práctica: definición de una base de datos para grafos e implementación de un set de consultas en JAVA

En este ejercicio se debe definir una base de datos para grafos para el almacenamiento de la información necesaria para la gestión de los artículos científicos publicados en revistas y conferencias.

La gestión de artículos científicos puede ser representada con el modelo entidad-relación de Figura 1.

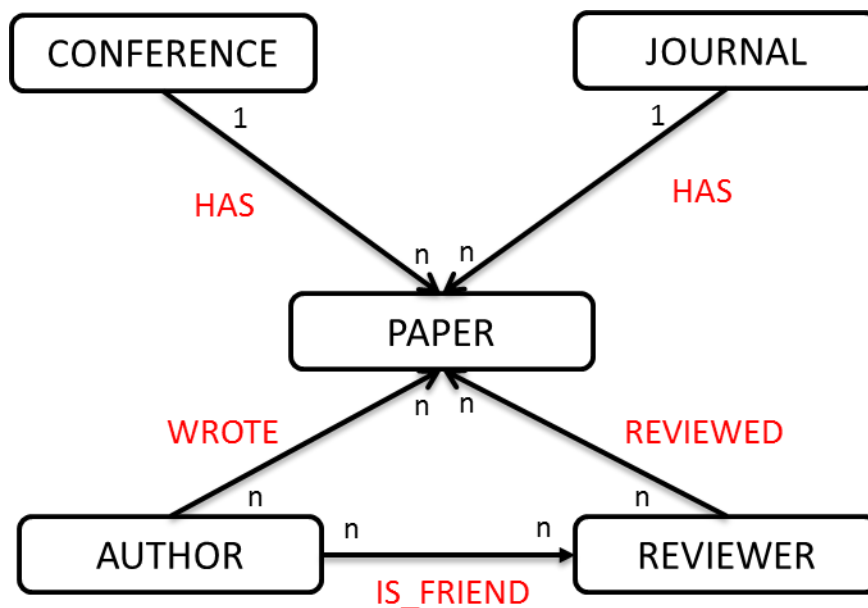


Figura 1 – Modelo Entidad-Relación para la gestión de artículos científicos.

Donde las entidades están definidas por los siguientes atributos.

Entidad	Atributos
Conference	Name, City, Year
Journal	Name, Year, Volume
Paper	Title, Year
Author	Surname
Reviewer	Surname

Los datos que se deberán almacenar en la base de datos serán proporcionados en 3 ficheros csv:

1. conferences.csv contiene la lista con los artículos publicados en conferencias. Contiene la siguiente información:
 - a. Authors: apellidos de los autores separados por “;”
 - b. ConferenceName: nombre de la conferencia
 - c. Year: año
 - d. City: ciudad
 - e. Reviewer: apellido del revisor
2. journals.csv contiene la lista con los artículos publicados en revistas. Contiene la siguiente información:
 - a. Authors: apellidos de los autores separados por “;”
 - b. JournalName: nombre de la revista
 - c. Year: año
 - d. Volume: numero del volumen
3. friendships.csv: contiene la lista con las amistades de cada revisor. Contiene la siguiente información:
 - a. Reviewer: apellido del revisor
 - b. Author: apellido del autor.

Se pide:

- 1) La creación y carga de la base de datos con Neo4j eligiendo la mejor forma de representar el modelo presentado.
- 2) La implementación de las siguientes consultas utilizando el lenguaje *Cypher*. Los resultados de las consultas se deberán escribir en un fichero de texto. El fichero de salida

solo contendrá los mensajes con los resultados de las consultas y deberá llamarse “nombreapellido.log”

1. Buscar los autores y revisores de un artículo dado.
 - Formado Salida: Q1: Titulo Paper, (Author1,...,AuthorN), Revisor
2. Buscar los artículos publicados en una edición particular de una conferencia.
 - Formado Salida: Q2: NombreConferencia, (Articulo1,..., ArticuloN)
3. Buscar los artículos publicados por un autor dado.
 - Formado Salida: Q3: ApellidoAutor, (Articulo1,..., ArticuloN)
4. Dado el volumen de una revista comprobar que para cada artículo el revisor no sea amigo de ningún de los autores.
 - Formado Salida: por cada amistad imprimir Q4: NombreArticulo, ApellidoAutor, ApellidoRevisor.

Entregas:

La práctica se realizará de forma individual. La práctica se entregará en el *Moodle* de la asignatura como un fichero “NombreApellido.zip”. El fichero zip debe contener:

- nombreapellido.jar: fichero jar con una clase master.neo4j.Neo4jApp que exponga los métodos:
 - constructor: recibe el path al fichero con la base de datos para inicializar Neo4j.
 - populate(String path): recibe el path a la carpeta con los 3 ficheros (conference.csv, journals.csv y friendships.csv) y los utiliza para popular la base de datos.
 - runQ1(String paperName)
 - runQ2(String conferenceName)
 - runQ3(String autorName)
 - runQ4(String journalName, int journalVolume)
- una carpeta src con el código fuente correspondiente al jar entregado.
- la carpeta graph.db de Neo4j correspondiente a la base de datos poblada.

La fecha límite de entrega es el día 18/01/2015.