

Estructura ejercicio Neo4j

Nodos

region (n_name, r_name, min_supplycost)
order (o_key, c_mktsegment, o_orderdate, o_shippriority)
LineItem (return_flag, linestatus, shipdate, quatity, extendedprice, discount, tax)
supplier (s_name, s_acctbal, s_address, s_phone, s_comment)
part (p_partkey, p_mfgr, p_size, p_type)

Porque juntamos region-nacion?

Como hemos podido observar, siempre que necesitemos una región también es requerida su nación, por lo tanto hemos decidido redundar la nación y incorporarla dentro de sus regiones, para así evitar un salto de nodos y hacer más eficientes las consultas. Aunque en ciertas ocasiones redundar los datos nos supone un coste alto en memoria, en nuestro caso el escaso número de naciones diferentes y regiones hacen que compense una parte irrisoria de memoria en favor de evitar un salto entre estos.

Porque juntamos order-customer?

Como en el caso anterior siempre que consultemos uno necesitamos el otro, en este caso solo necesitamos un único atributo de customer, por tanto hemos decidido incorporar entre atributo en la order, en este caso la redundancia es mayor, pero seguimos evitando un salto entre nodos. En este caso asumimos que se quiere como objetivo principal la eficiencia en frente del ahorro de memoria, en caso que fuera a la inversa no se podría redundar esta información.

Porque materializamos el min_supplycost en el nodo de región?

Para optimizar la segunda query hemos decidido materializar un nuevo atributo para el nodo región. De esta forma cuando accedemos para conseguir los nodos que tiene un supplycost inferior no tenemos que acceder a todos los nodos, ya que solo accedemos a aquellos que corresponden con el atributo determinado previamente. De esta forma evitamos tener que hacer un pre-recorrido de todos los nodos para detectar los nodos con el mínimo supplycost.

