

Altri modelli dei dati basati su grafi – Property graph

1 Nuove applicazioni dei modelli basati su grafi

Negli ultimi anni (a partire circa dal 2015), si è avuto un ritorno di interesse riguardo ai modelli dei dati basati su grafi, motivato dall'esigenza di analisi dei dati provenienti, ad esempio, dalle reti sociali.

Ciò ha portato allo sviluppo di DBMS nativi basati su grafi, ciascuno con un proprio linguaggio di interrogazione. Al momento, è in corso un processo di standardizzazione finalizzato a cercare di uniformare tutti questi linguaggi di interrogazione: per ora, il linguaggio che sembra prevalere è Cypher, che è associato al DBMS Neo4j.

Come già detto, i precursori di queste tecnologie sono i modelli e linguaggi per la rappresentazione della conoscenza, introdotti intorno agli anni '70 nell'ambito dell'intelligenza artificiale, e poi ripresi per formare le basi del semantic web (con la definizione di RDF, ecc.).

2 Elementi dei linguaggi di interrogazione

In generale, i linguaggi di interrogazione per i modelli dei dati basati su grafi si fondano su due concetti principali:

- *graph pattern* (pattern di grafi)
- *path expression*.

Ad esempio, considerando un ipotetico database bibliografico (contenente autori, articoli, ecc.) rappresentato come grafo,

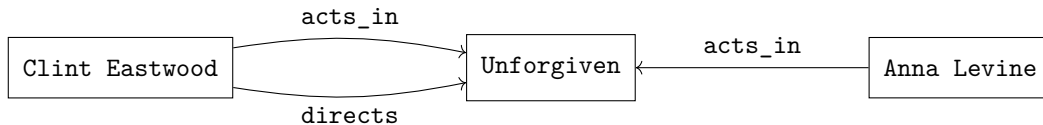
- un esempio di interrogazione basata su graph pattern sarebbe: “trovare tutte le coppie di autori di articoli pubblicati in conferenze”;
- un esempio di interrogazione basata su path expression sarebbe invece: “trovare i cammini più corti tra due autori”.

3 Tipi di grafi

In RDF, il modello dei dati è un **edge-labelled multigraph**:

- *edge labelled* significa che (anche) gli archi (oltre ai nodi) hanno una label (etichetta);
- *multigraph* significa che sono ammessi più archi tra gli stessi due nodi.

Ad esempio:

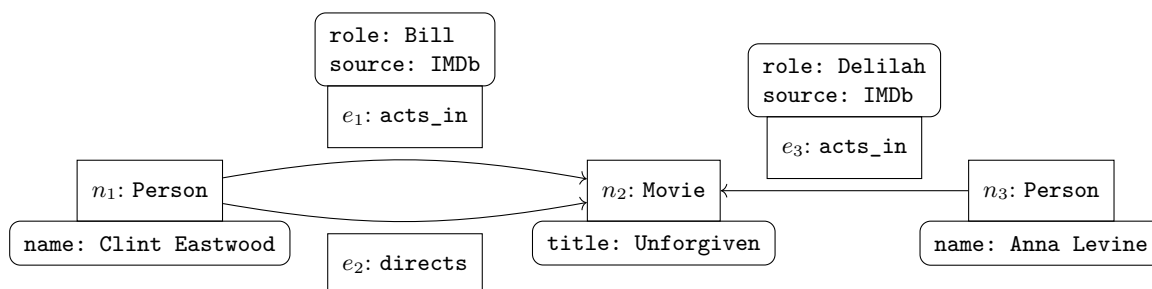


Tale approccio è abbastanza limitato nel caso in cui si vogliono aggiungere nuove informazioni sia ai nodi che agli archi: per fare ciò, bisogna infatti modificare la struttura del grafo. Per questo motivo, può essere conveniente associare alle label (dei nodi e degli archi) degli insiemi di coppie chiave–valore, ottenendo così un modello chiamato **property graph**.

Più in dettaglio, in un property graph ogni nodo e ogni arco è etichettato con un identificativo univoco, che associa al nodo / arco stesso un insieme di coppie *proprietà–valore* (in questo ambito, si usa prevalentemente il termine “proprietà”, piuttosto che “chiave”) chiamate **attributi**.

Quindi, attraverso l’aggiunta / modifica degli attributi dei nodi e degli archi, è possibile annotare il grafo *senza* doverne modificare la struttura.

Ad esempio, le informazioni rappresentate nel grafo mostrato sopra potrebbero essere riportate nel seguente property graph (nel quale sono annotate anche alcune informazioni aggiuntive):



Osservazione: In questo grafo:

- ogni nodo e ogni arco ha una singola label;
- ogni proprietà ha un singolo valore.

Rimuovendo parzialmente o completamente questi vincoli, si ottiene un modello dei dati ancora più generale, talvolta indicato con il termine *knowledge graph* (ma, in realtà, tale termine non ha una definizione formale, e viene usato per riferirsi a diversi tipi di grafi).