

attraversamento del grafo. [1]

Un GraphDB è stato usato in MuseoTorino per collegare schede attraverso distinte relazioni, come illustrato in **figura 1**.

In questo esempio si vede come la scheda di riferimento (Camillo Benso Conte di Cavour) sia direttamente collegata ad altre schede:

- una doppia relazione con la scheda *Torino*;
- una relazione con la scheda *Castello di Cavour* di tipo *Tumulato*;
- una relazione con la scheda *Palazzo Cavour* di tipo *Vive*;
- altre relazioni con schede di tipo evento (i vari Governi).

È interessante notare come il *Castello di Cavour* e il *Palazzo Cavour* siano direttamente correlati a Cavour, ma siano anche indirettamente collegati fra di loro tramite una *correlazione di secondo livello*. Questa correlazione non è stata definita da un input dell'utente, ma, in un GraphDB, questa informazione è per definizione "implicita" e pertanto è una informazione facile (e rapida) da recuperare (per esempio nel momento in cui si sta accedendo ai dati che riguardano la scheda del *Palazzo Cavour*).

Il GraphDB è molto veloce a cercare le schede in esso memorizzate: così, nell'esempio di cui sopra, si è in grado di caricare tutte le relazioni di livello 2 per proporre, filtrando i risultati nel modo migliore in base a rapporti derivanti o al tipo di scheda, quella che potrebbe essere di interesse per un utente nel momento in cui ne stia analizzando una specifica. Per esempio la scheda *Castello di Cavour* potrebbe essere di interesse per un utente che sta valutando la scheda *Palazzo Cavour*, poiché il sistema riesce ad analizzare in tempi rapidi tutte le relazioni fra sé e quelle direttamente o indirettamente correlate a più livelli di profondità.

Questa è una delle migliori caratteristiche in un GraphDB, poiché il grafo delle relazioni è libero da schemi e può essere evoluto o raffinato in un secondo momento. Questa funzionalità è un *must-to-have* nella gestione di un catalogo e, a livello tecnico, è lo stesso sistema che sta alla base dei social network più famosi (Facebook, Twitter ecc), che implementano questi

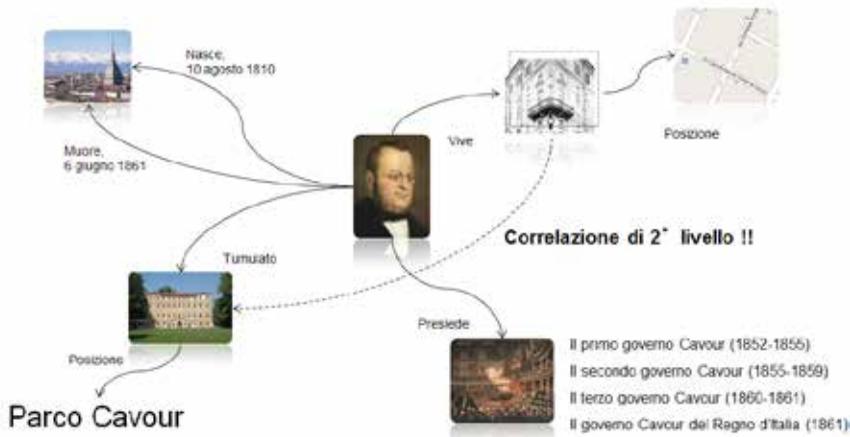


Figura 1. Esempio di rappresentazione in un GraphDB.

prevedibili o predefinitibili, e RDF DB risolve le esigenze di ricerca di oggetti attraverso le loro relazioni (ad esempio, un grafico sociale);

- *il geo-posizionamento* è gestito al meglio in un GIS (Sistema informativo geografico), adatto a gestire migliaia di oggetti geo-posizionati e a permettere le ricerche di tipo spaziale (ad esempio quali oggetti risiedono all'interno di un'area geografica definita attraverso coordinate).

Tutti questi strumenti permettono di risolvere problemi specifici, ma in MuseoTorino si è cercato uno strumento che unisse tutte queste caratteristiche in una sola soluzione. Per questo è stato utilizzato un *GraphDB*.

Un GraphDB ha tutte le caratteristiche precedentemente descritte: è schema-less, come un DocumentDB; gestisce milioni di oggetti e relazioni, come un RDF; dispone di integrazione spaziale come un GIS.

Un GraphDB è la soluzione migliore per la gestione di un catalogo eterogeneo e le molteplici relazioni fra i dati.

III. IL GRAPHDB (DATABASE A GRAFI)

Un GraphDB è un sistema software (database) che memorizza i dati rappresentandoli come un insieme di vertici (nodi, ovvero punti) collegati fra di loro da un insieme di archi (collegamenti, ovvero linee). Questi database sono ottimizzati per eseguire un tipo di processo noto come un