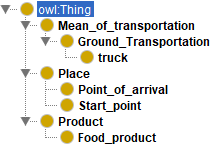
***ONTOLOGIA***

Per modellare l'ontologia è stato utilizzato il programma Protégé attraverso il quale sono state definite tutte le entità utili per il nostro scopo. Inoltre, grazie al reasoner HermiT è stato possibile interrogare l'ontologia, selezionando il tab "DL query", e ottenere le informazioni in merito.

Tra le entità definite abbiamo:

* ***Le classi***, un insieme di attuali e potenziali individui, e le sottoclassi



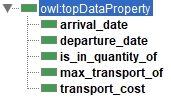
* ***Gli individui***, entità concrete o astratte, membri di una determinata classe.



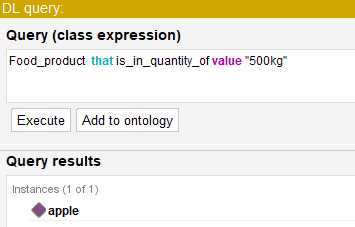
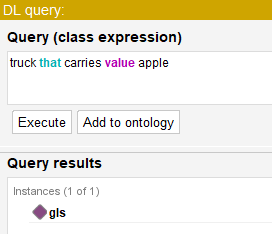
* ***Le relazioni tra oggetti*** che permettono di mettere in relazione due individui della stessa classe o di classi diverse.

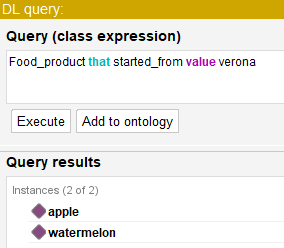


* ***Le relazioni tra dati*** che permettono di mettere in relazione un individuo con un tipo primitivo.

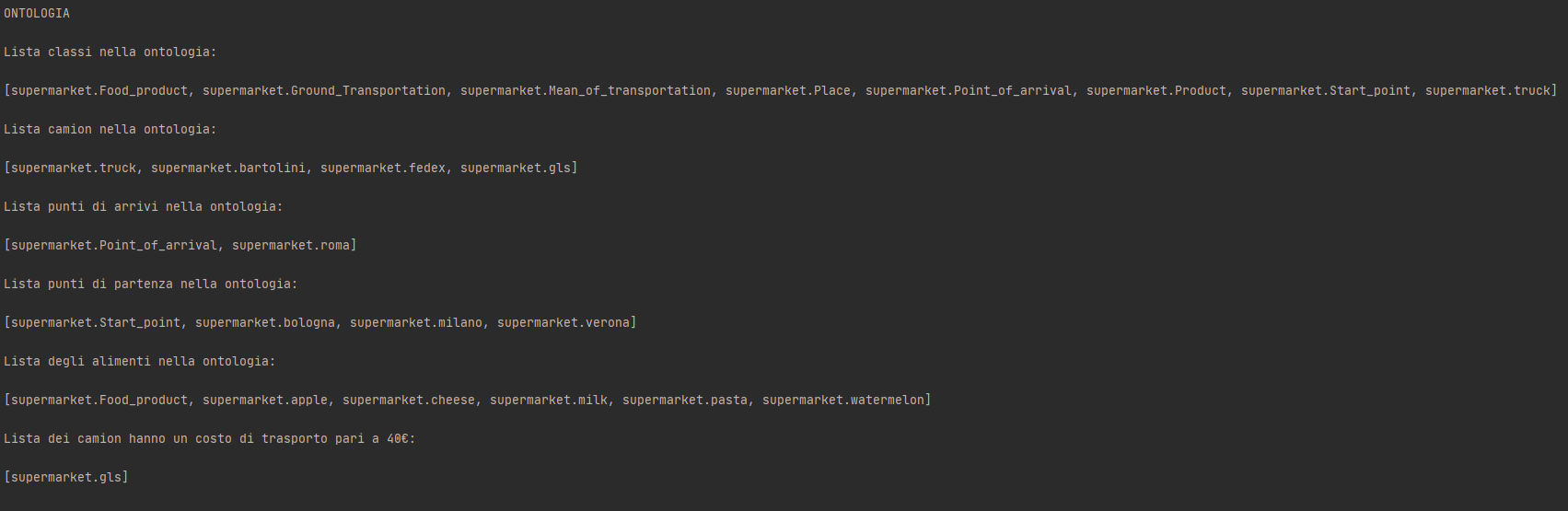


Dopo la creazione dell'ontologia sono state formulate delle query per interrogarla e verificare le risposte ottenute. Ecco alcuni esempi:



Infine, è possibile accedere all'ontologia anche tramite l'utlizzo della libreria OwlReady2 in Python permettendo anche la formulazione delle query. Essa, in base allo script definito, fornisce il seguente output:



***PROBLEMA DELLO ZAINO CON RISOLUZIONE BRANCH AND BOUND***

Il problema dello zaino è un problema di ottimizzazione combinatoria posta nel seguente modo:

*Sia dato uno zaino che possa sopportare un determinato peso e siano dati N oggetti, ognuno dei quali caratterizzato da un peso e un valore. Il problema si propone di scegliere quali di questi oggetti mettere nello zaino per ottenere il maggiore valore senza eccedere il peso sostenibile dallo zaino stesso.*

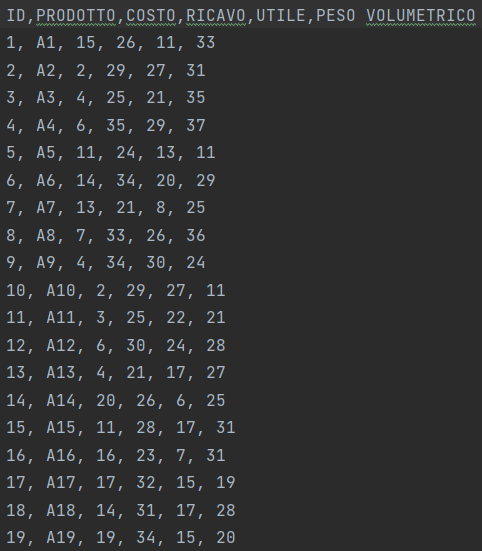
*La tecnica di risoluzione utilizzata per questo problema è quella del Branch and Bound, un metodo di risoluzione per problemi di ottimizzazione combinatoria applicabile principalmente quando esistono molti percorsi verso il nodo obiettivo e, soprattutto, quando si vuole il percorso migliore.*

Tale tecnica di risoluzione è stata utilizzata nel nostro problema dove dato il peso massimo che il camion può contenere, determinare la combinazione di oggetti che restituisce l’utile massimo senza eccedere il peso sostenibile del camion stesso.

In questo problema i dati seguenti sono:

* N=100 oggetti con un utile e un determinato peso volumetrico riportati nel file ‘Output\_.csv’
* W che corrisponde al peso massimo del camion dato in input dall’utente.

Di seguito è riportato un parte del file ‘Output\_.csv’



Alla fine si restituisce l’utile massimo ottenuto in base al peso massimo fornito in input. Ecco un esempio:

