```
@prefix swp: <http://www.semanticwebprimer.org/ontology/apartments.ttl#>.
@prefix dbpedia: <http://dbpedia.org/resource/>.
@prefix dbpedia-owl: <http://dbpedia.org/ontology/>.


swp:BaronWayApartment swp:hasNumberOfBedrooms 3;
                        swp:isPartOf swp:BaronWayBuilding.
swp:BaronWayBuilding dbpedia-owl:location dbpedia:Amsterdam,
                dbpedia:Netherlands.
```

We might like to ask a query over this data. For example: find the location of the building. How would we express this in SPARQL? We can build up this query as follows. We would want to match the following triple:

swp:BaronWayBuilding dbpedia-owl:location dbpedia:Amsterdam.

In SPARQL, we can just replace any element of the triple with a variable. Variables are denoted by a ? at their beginning. Introducing a variable for location, we would write the following:

swp:BaronWayBuilding dbpedia-owl:location ?location.

The triple store will take this *graph pattern* and try to find sets of triples that match the pattern. Thus, running this pattern over the original RDF, a triple store would return dbpedia:Amsterdam and dbpedia:Netherlands. Essentially, it finds all triples where swp:BaronWayBuilding is in the subject position and dbpedia-owl:location is in the predicate position.

To make this a complete SPARQL query, a couple of additions need to be made. First, all prefixes need to be defined. We also need to tell the triple store that we are interested in the results for a particular variable. Thus, a complete SPARQL query for the above query would be as follows: