SPARQL Update

Bernd Neumayr, Johannes Kepler University Linz

with contributions from Dieter Steiner

also see

http://www.w3.org/TR/sparql11-query/

http://www.w3.org/TR/sparql11-update/

Agenda

- SPARQL Update on RDF Graphs

```
DELETE { }
INSERT { }
WHERE  { }
```

- SPARQL Query over RDF Datasets (multiple named graphs)

```
WHERE { GRAPH ?g {  } }
```

- SPARQL Update on RDF Graph Store (updatable dataset)

```
DELETE { GRAPH ?g { } }
INSERT { GRAPH ?h { } }
WHERE  { GRAPH ?k { } }
```

Update on Single Graph

SPARQL Update on RDF Graphs

SPARQL Update on Single Graphs

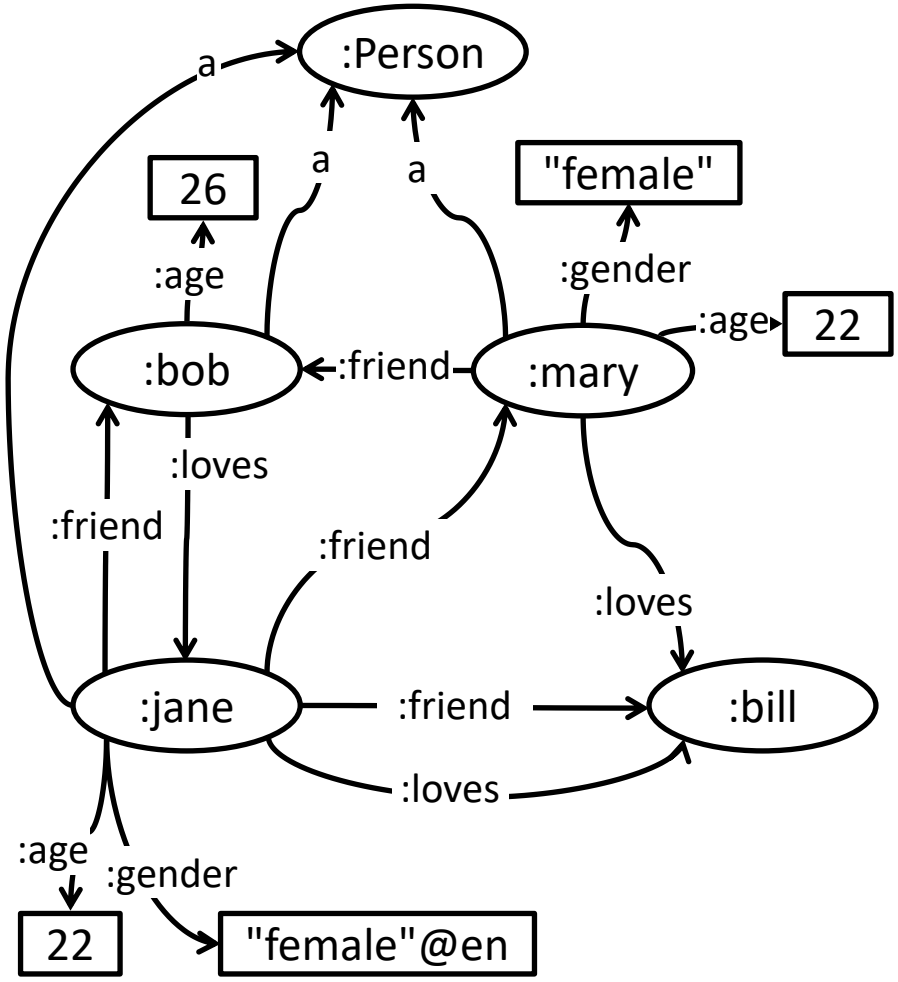
- The INSERT DATA operation adds some triples, given inline in the request, into a graph.
- The DELETE DATA operation removes some triples, given inline in the request, if the respective graph contains those.
- The fundamental pattern-based actions for graph updates are INSERT and DELETE (which can co-occur in a single DELETE/INSERT operation). These actions consist of groups of triples to be deleted and groups of triples to be added. The specification of the triples is based on query patterns.
- The difference between INSERT / DELETE and INSERT DATA / DELETE DATA is that INSERT DATA and DELETE DATA do not substitute bindings into a template from a pattern. The DATA forms require concrete data .

Insert Data (1/3)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://example.org/>

INSERT DATA {
  :jane  a :Person;
    :gender "female"@en; :age 22;
    :friend :mary, :bob, :bill;
    :loves :bill.
  :mary  a :Person;
    :gender "female"; :age 22;
    :friend :bob;
    :loves :bill.
  :bob  a :Person;
    :age 26;
    :loves :jane.
};

INSERT DATA{
  :mary :age 24.
  :bob a :Person;
    :age 28.
  :bill  a :Person;
    :gender "male";
    :friend :mary, :jane.
}
```

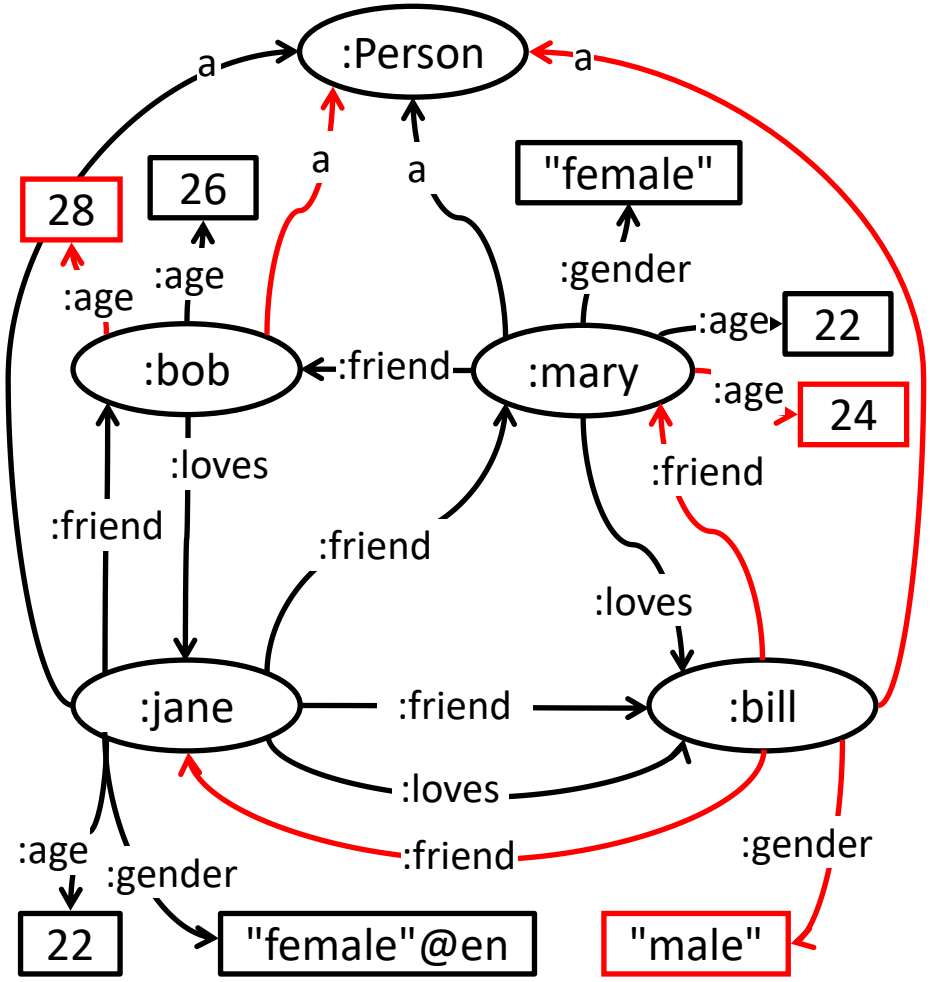
Insert Data (2/3)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://example.org/>

INSERT DATA {
  :jane  a :Person;
    :gender "female"@en; :age 22;
    :friend :mary, :bob, :bill;
    :loves :bill.
  :mary  a :Person;
    :gender "female"; :age 22;
    :friend :bob;
    :loves :bill.
  :bob  a :Person;
    :age 26;
    :loves :jane.
};

INSERT DATA{
  :mary :age 24.
  :bob a :Person;
    :age 28.
  :bill  a :Person;
    :gender "male";
    :friend :mary, :jane.
}
```



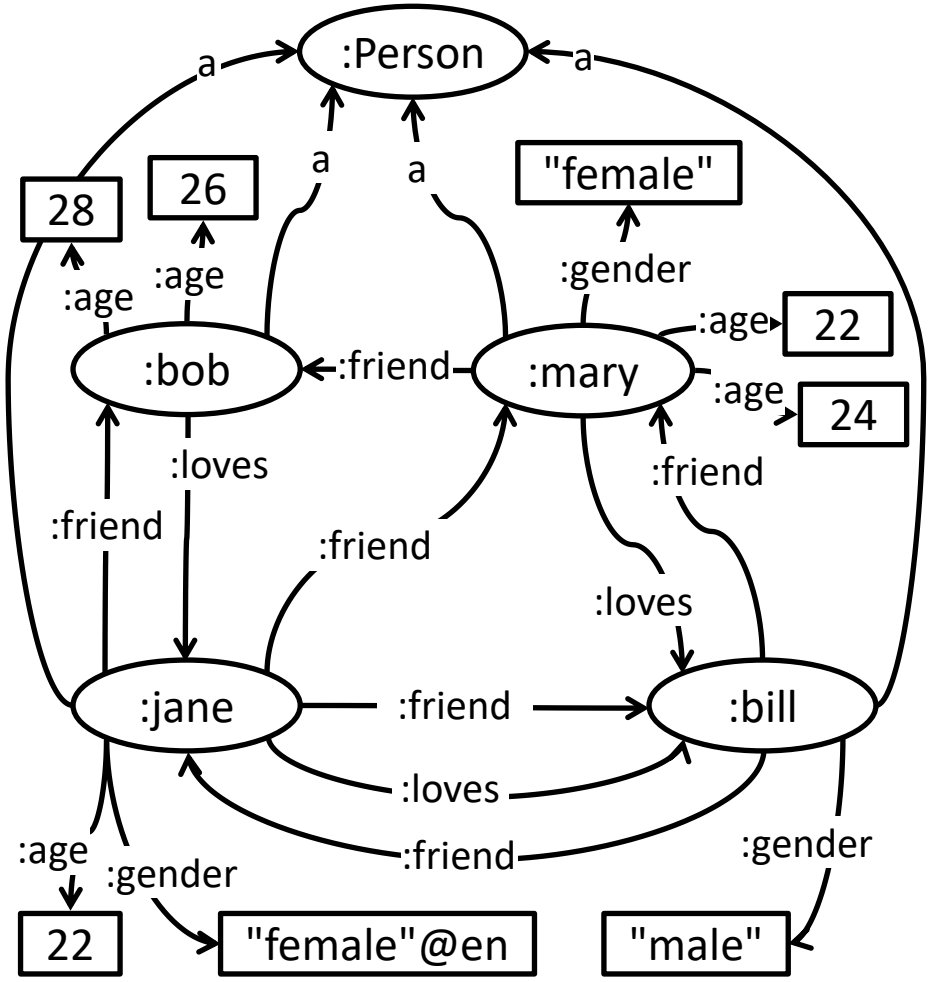
Insert Data (3/3)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://example.org/>

INSERT DATA {
  :jane  a :Person;
    :gender "female"@en; :age 22;
    :friend :mary, :bob, :bill;
    :loves :bill.
  :mary  a :Person;
    :gender "female"; :age 22;
    :friend :bob;
    :loves :bill.
  :bob  a :Person;
    :age 26;
    :loves :jane.
};

INSERT DATA{
  :mary :age 24.
  :bob a :Person;
    :age 28.
  :bill  a :Person;
    :gender "male";
    :friend :mary, :jane.
}
```



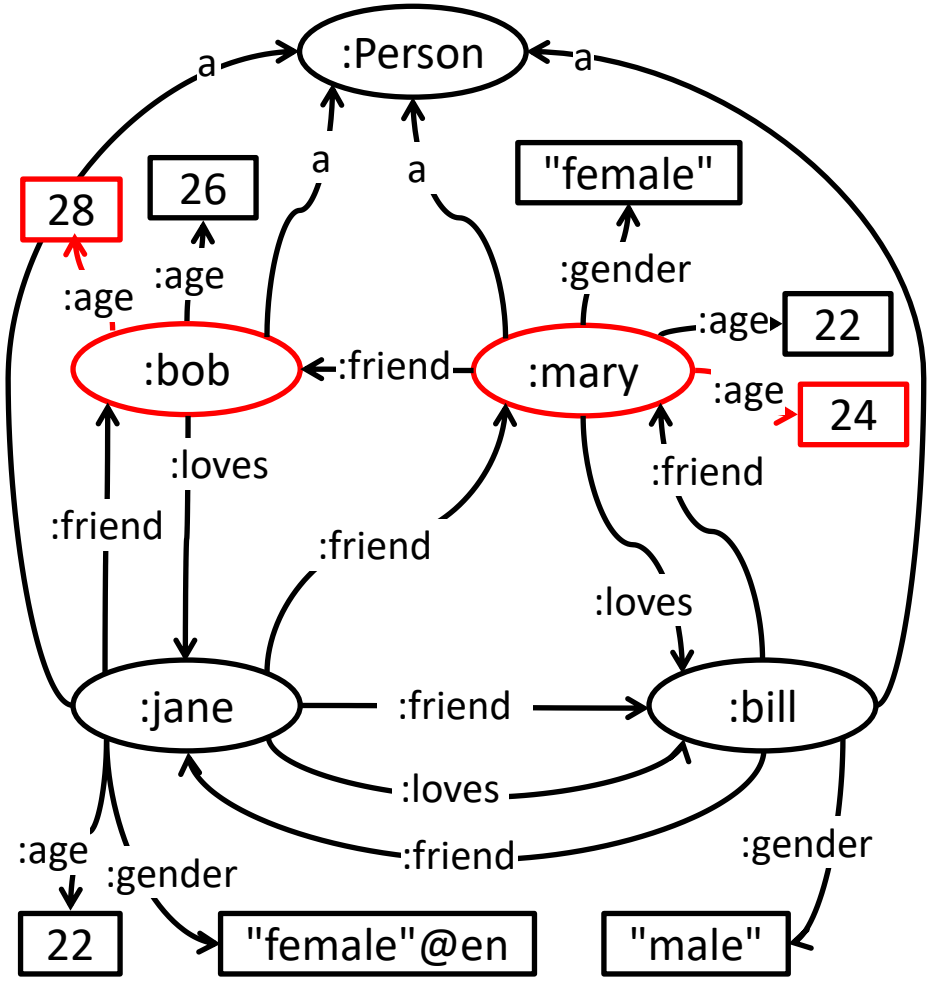
Delete Data (1/3)

```
DELETE DATA
  { :mary :age 24.
    :bob :age 28.
    :bob :age 43.
  }
```



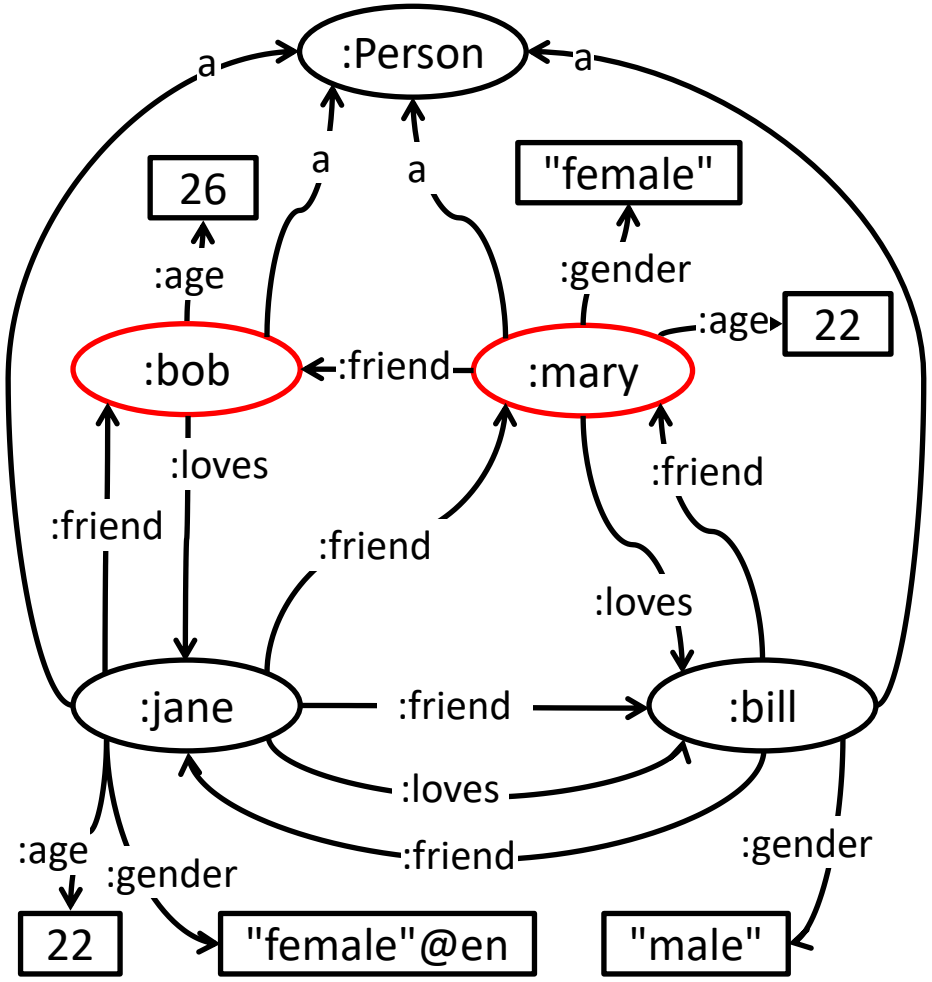
Delete Data (2/3)

```
DELETE DATA
  { :mary :age 24.
    :bob :age 28.
    :bob :age 43.
  }
```



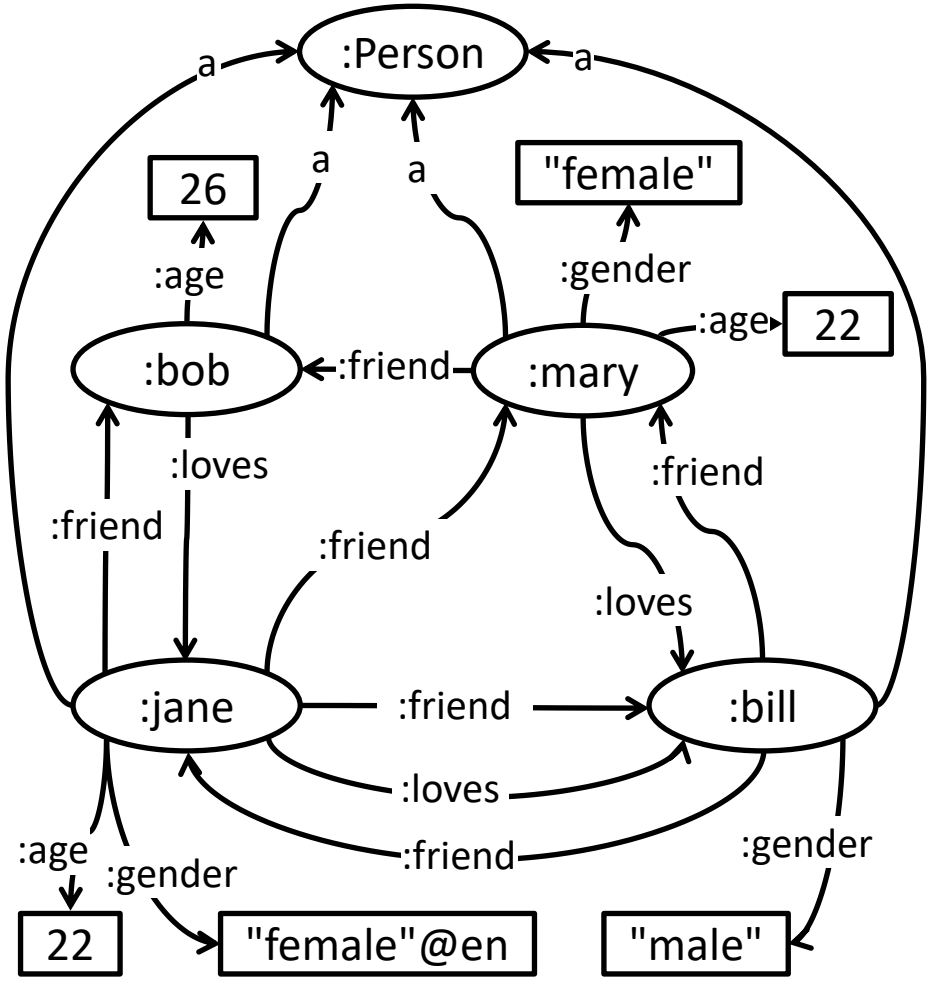
Delete Data (3/3)

```
DELETE DATA
  { :mary :age 24.
    :bob :age 28.
    :bob :age 43.
  }
```



Delete/Insert (1/8)

```
DELETE {?p :age ?age_old}
INSERT {?p :age ?age_new}
WHERE
  { ?p a :Person.
    ?p :age ?age_old.
    BIND(?age_old + 1 AS ?age_new)
  }
```

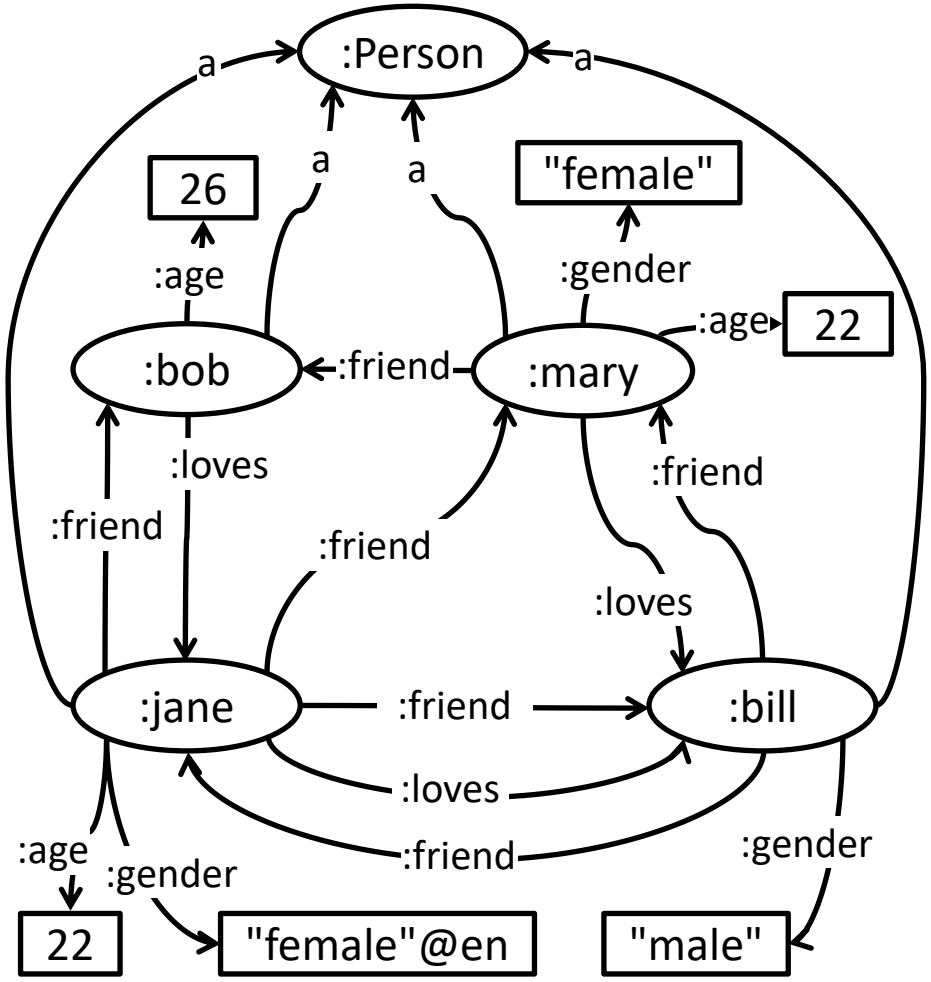


Delete/Insert (2/8)

```
DELETE {?p :age ?age_old}
INSERT {?p :age ?age_new}
WHERE
  { ?p a :Person.
    ?p :age ?age_old.
    BIND(?age_old + 1 AS ?age_new)
  }
```

bindings for the query pattern:

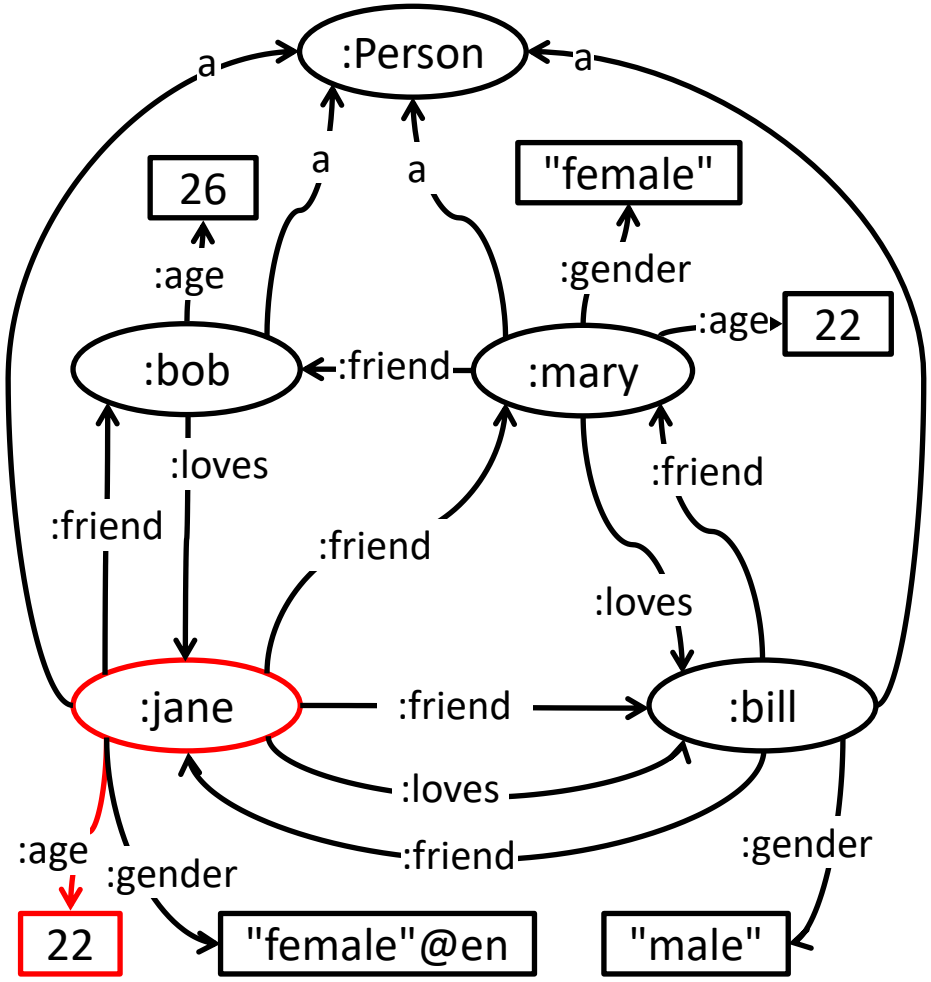
p	age_old	age_new
:jane	22	23
:mary	22	23
:bob	26	27



Delete/Insert (3/8)

```
DELETE {?p :age ?age_old}
INSERT {?p :age ?age_new}
WHERE
  { ?p a :Person.
    ?p :age ?age_old.
    BIND(?age_old + 1 AS ?age_new)
  }
```

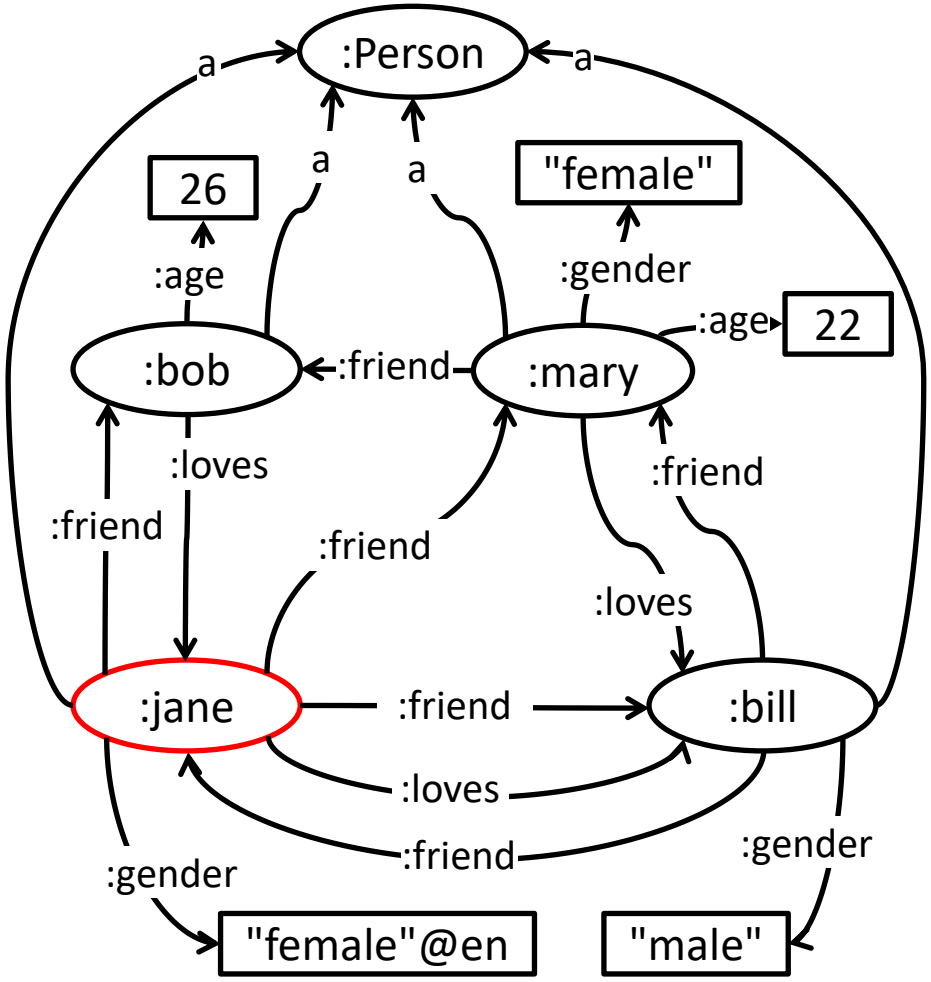
p	age_old	age_new
:jane	22	23
:mary	22	23
:bob	26	27



Delete/Insert (4/8)

```
DELETE {?p :age ?age_old}
INSERT {?p :age ?age_new}
WHERE
  { ?p a :Person.
    ?p :age ?age_old.
    BIND(?age_old + 1 AS ?age_new)
  }
```

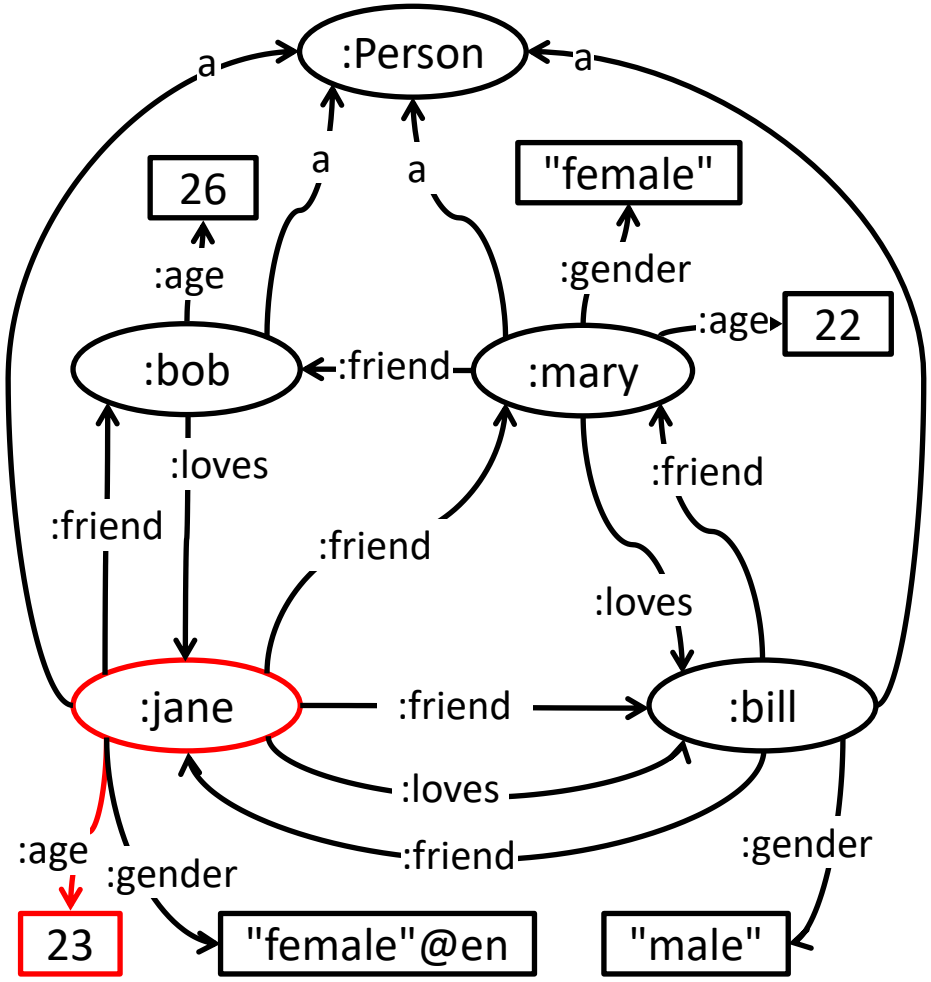
p	age_old	age_new
:jane	22	23
:mary	22	23
:bob	26	27



Delete/Insert (5/8)

```
DELETE {?p :age ?age_old}
INSERT {?p :age ?age_new}
WHERE
  { ?p a :Person.
    ?p :age ?age_old.
    BIND(?age_old + 1 AS ?age_new)
  }
```

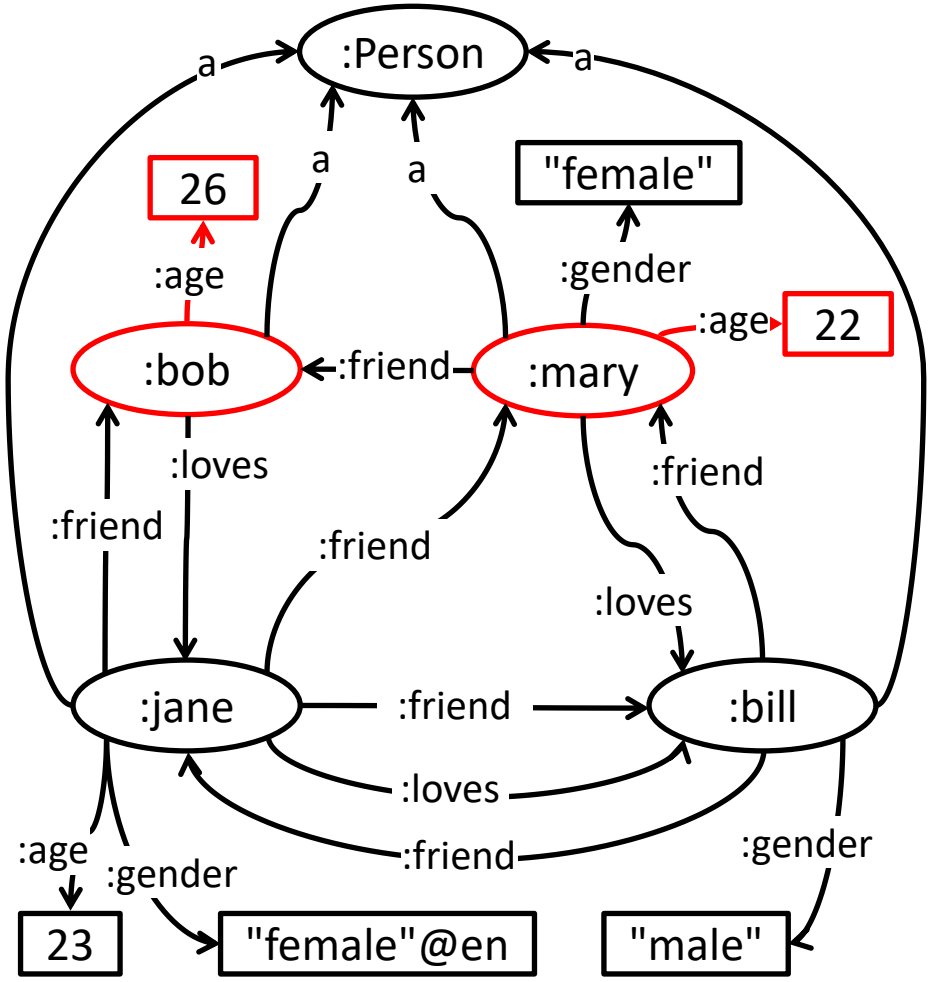
p	age_old	age_new
:jane	22	23
:mary	22	23
:bob	26	27



Delete/Insert (6/8)

```
DELETE {?p :age ?age_old}
INSERT {?p :age ?age_new}
WHERE
  { ?p a :Person.
    ?p :age ?age_old.
    BIND(?age_old + 1 AS ?age_new)
  }
```

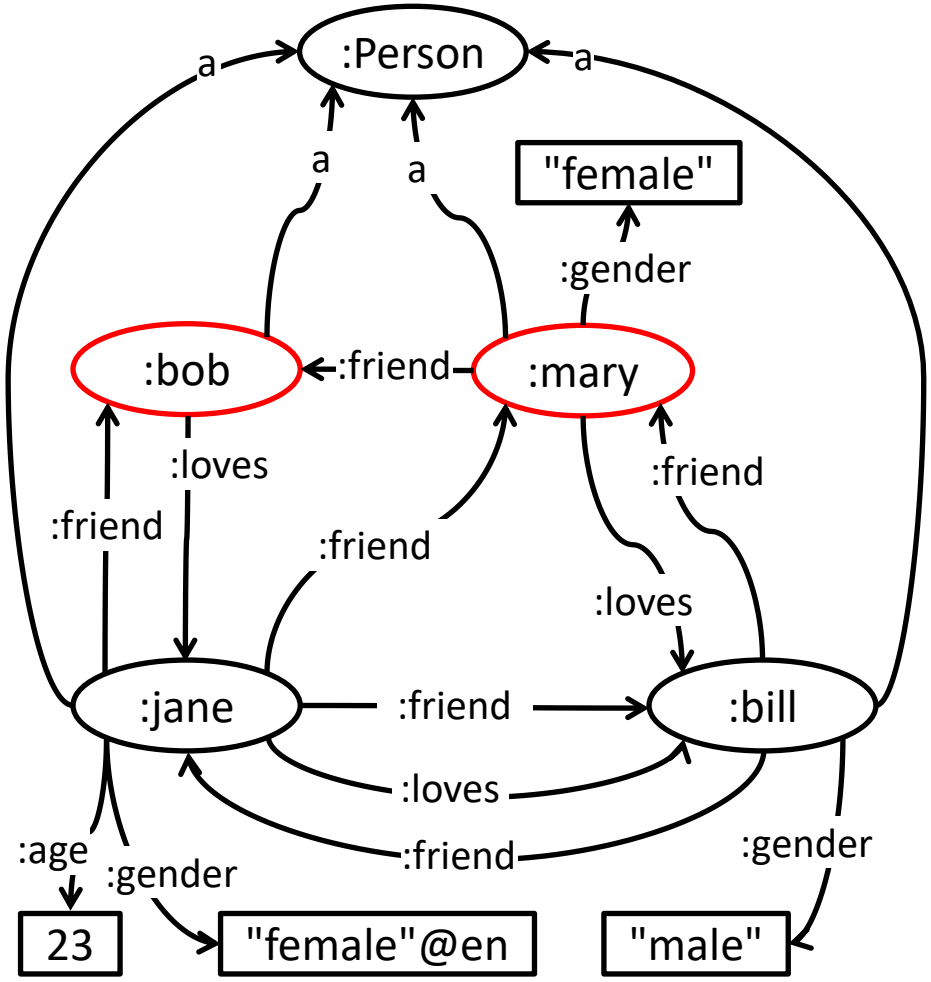
p	age_old	age_new
:jane	22	23
:mary	22	23
:bob	26	27



Delete/Insert (7/8)

```
DELETE {?p :age ?age_old}
INSERT {?p :age ?age_new}
WHERE
  { ?p a :Person.
    ?p :age ?age_old.
    BIND(?age_old + 1 AS ?age_new)
  }
```

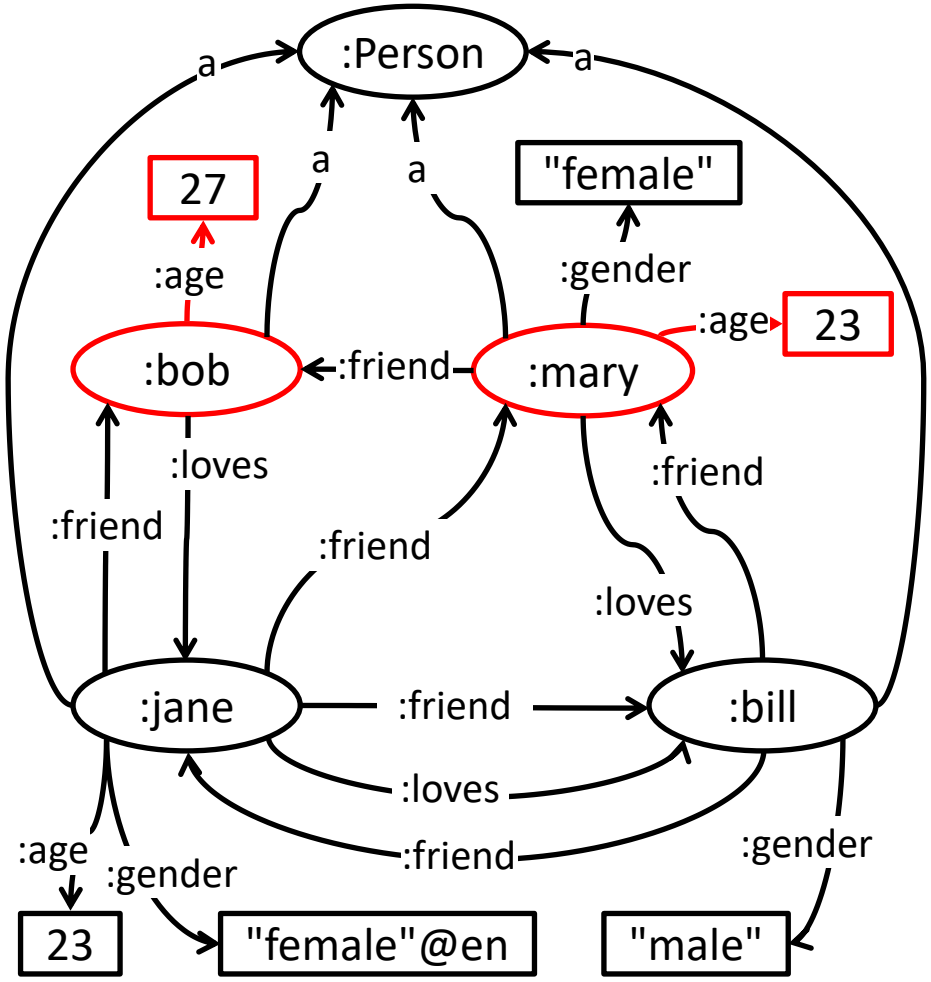
p	age_old	age_new
:jane	22	23
:mary	22	23
:bob	26	27



Delete/Insert (8/8)

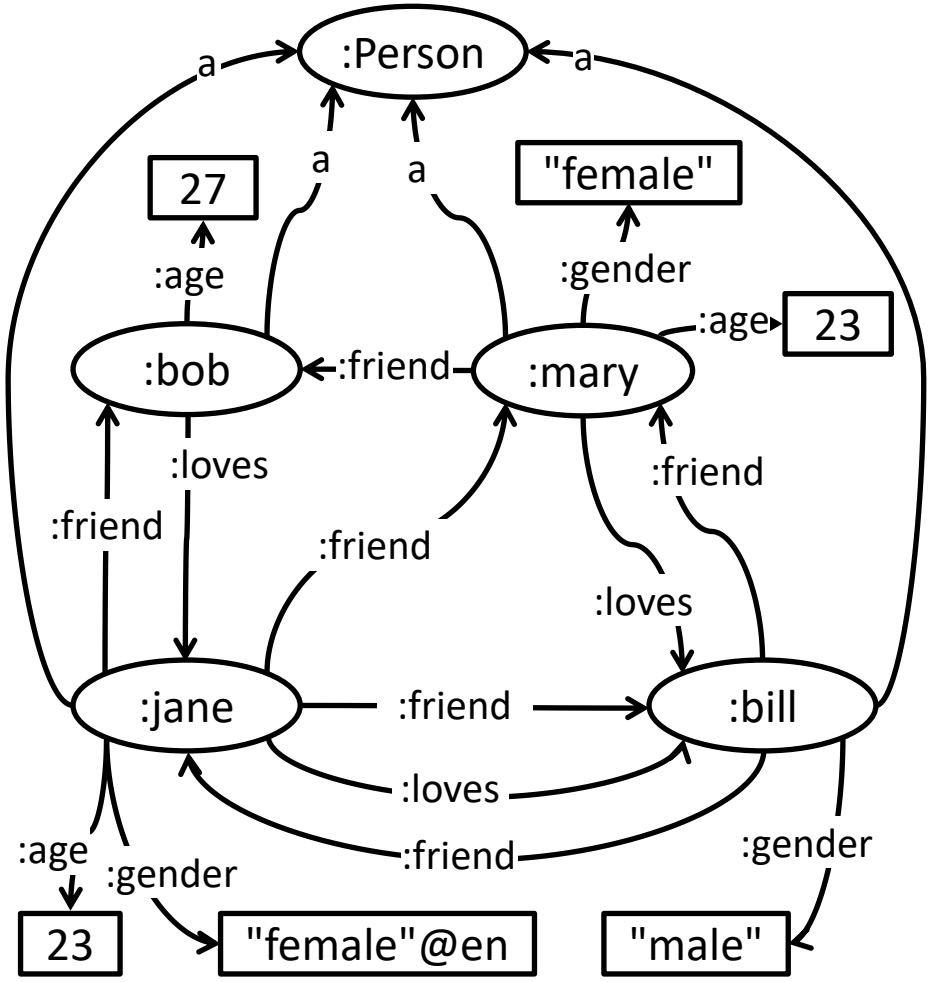
```
DELETE {?p :age ?age_old}
INSERT {?p :age ?age_new}
WHERE
  { ?p a :Person.
    ?p :age ?age_old.
    BIND(?age_old + 1 AS ?age_new)
  }
```

p	age_old	age_new
:jane	22	23
:mary	22	23
:bob	26	27



Insert with Subquery in WHERE clause (1/3)

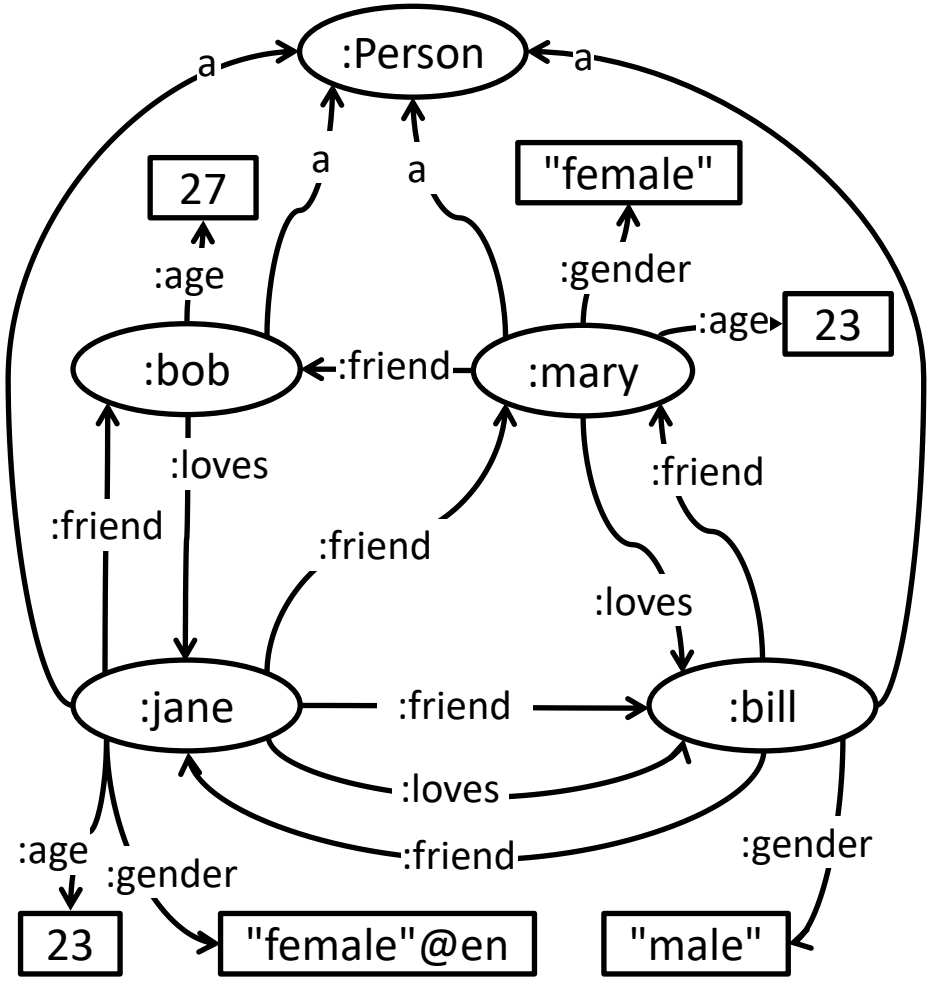
```
INSERT {?p :nrOfFriends ?nr}
WHERE
  { SELECT ?p (COUNT(?f) AS ?nr)
    WHERE
      { ?p a :Person.
        ?p :friend ?f.
      }
    GROUP BY ?p
  }
```



Insert with Subquery in WHERE clause (2/3)

```
INSERT {?p :nrOfFriends ?nr}
WHERE
  { SELECT ?p (COUNT(?f) AS ?nr)
    WHERE
      { ?p a :Person.
        ?p :friend ?f.
      }
    GROUP BY ?p
  }
```

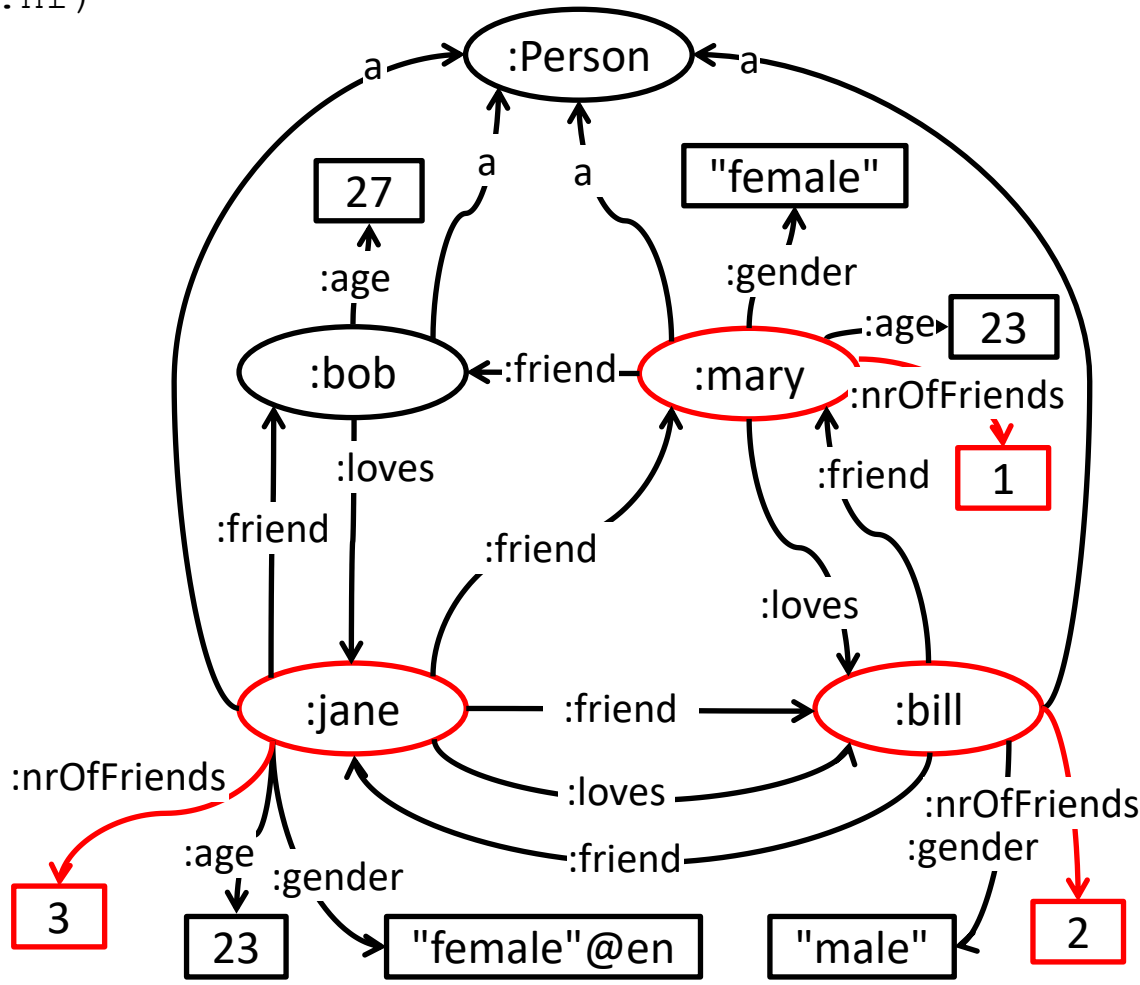
p	nr
:jane	3
:mary	1
:bil	2



Insert with Subquery in WHERE clause (3/3)

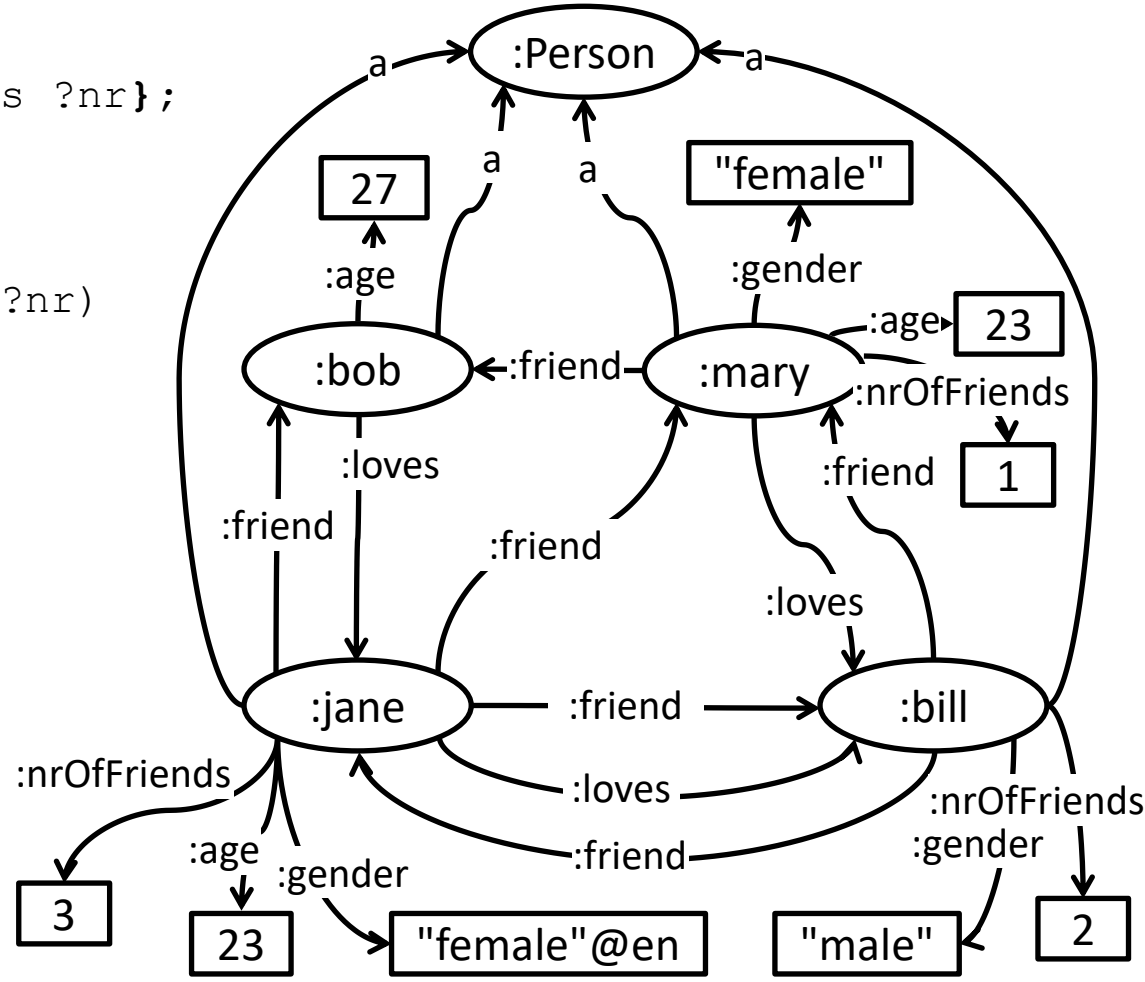
```
INSERT {?p :nrOfFriends ?nr}
WHERE
  { SELECT ?p (COUNT(?f) AS ?nr)
    WHERE
      { ?p a :Person.
        ?p :friend ?f.
      }
    GROUP BY ?p
  }
```

p	nr
:jane	3
:mary	1
:bill	2



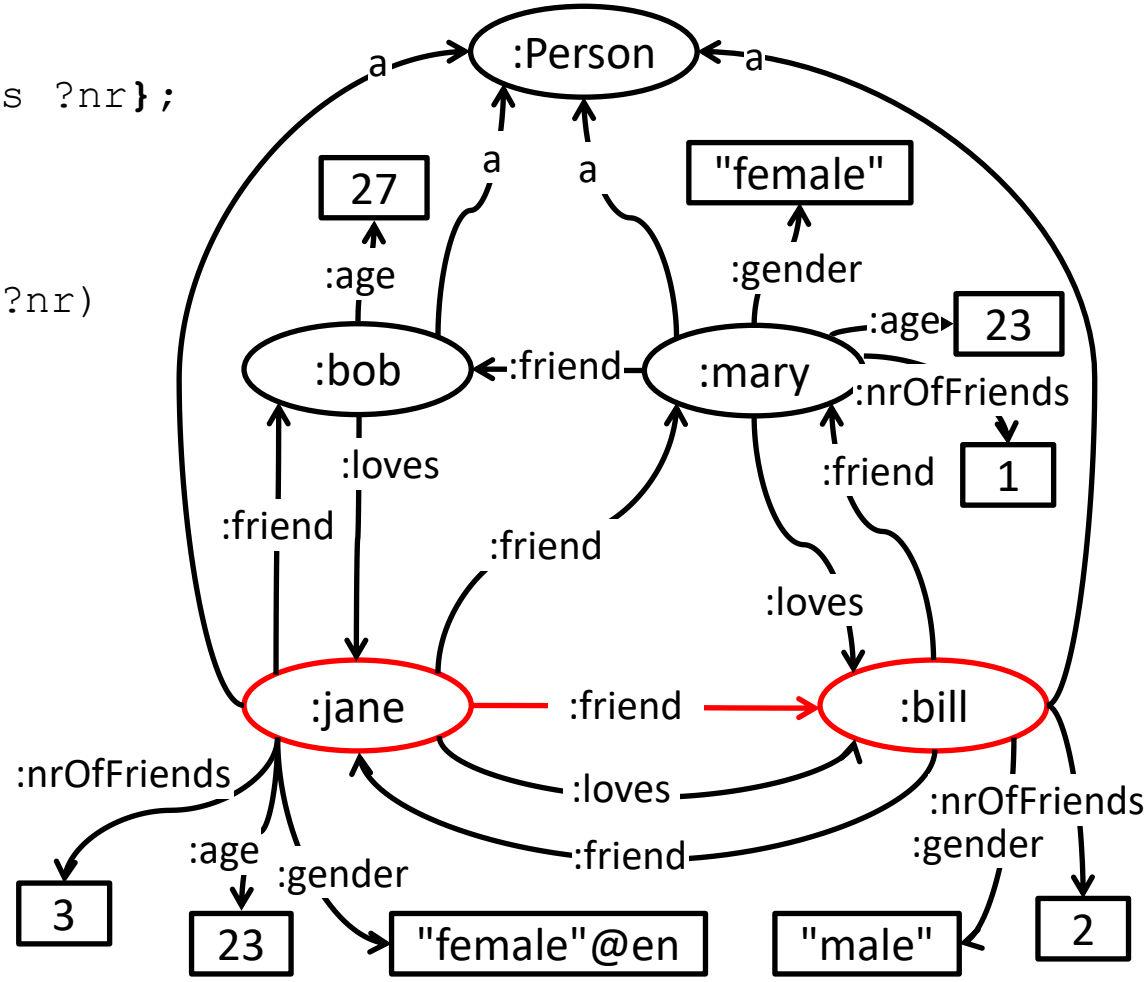
Update and Re-Calculation (1/6)

```
DELETE DATA
  { :jane :friend :bill };

DELETE WHERE {?p :nrOfFriends ?nr};

INSERT {?p :nrOfFriends ?nr}
WHERE
  { SELECT ?p (COUNT(?f) AS ?nr)
    WHERE
      { ?p a :Person.
        ?p :friend ?f.
      }
    GROUP BY ?p
  }
```



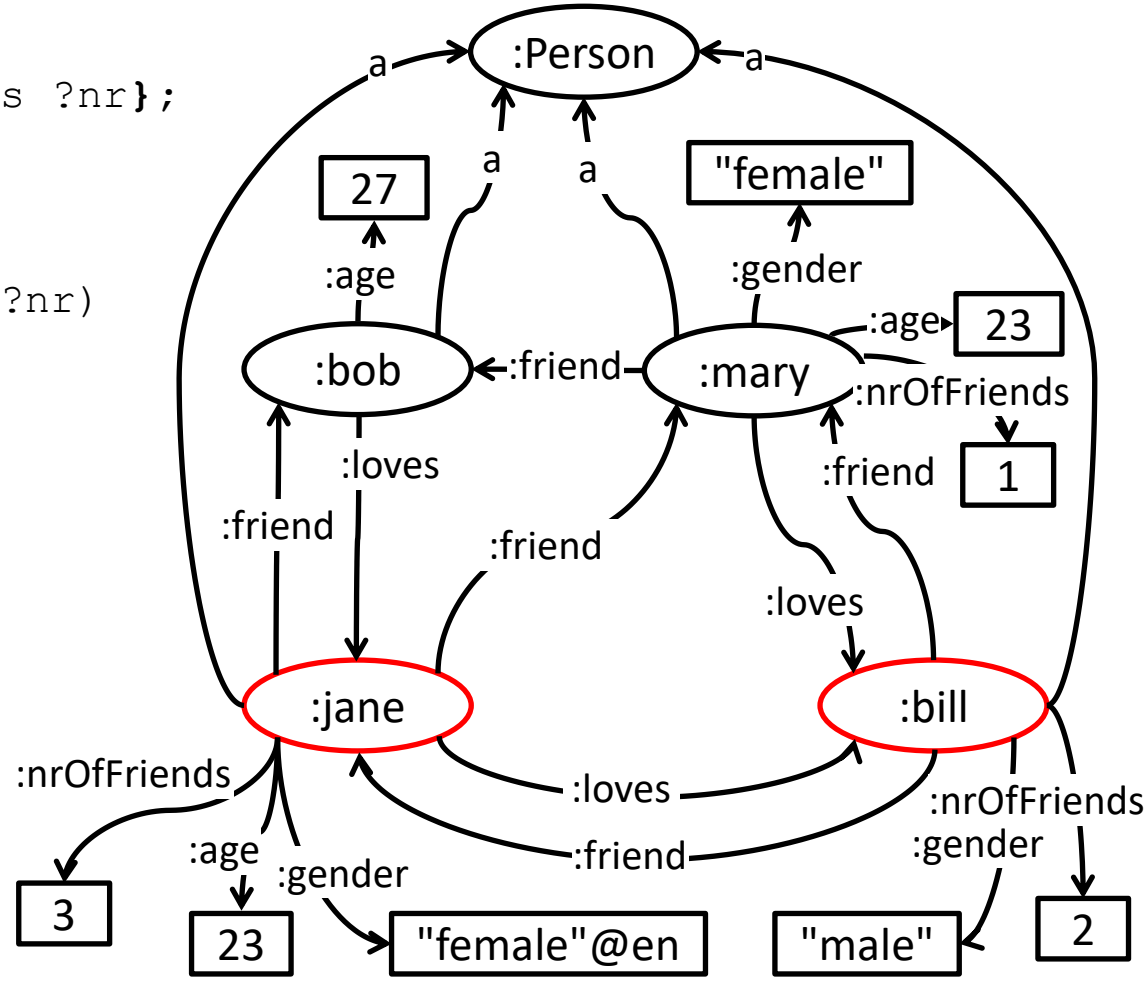
Update and Re-Calculation (2/6)

```
DELETE DATA
  { :jane :friend :bill };

DELETE WHERE {?p :nrOfFriends ?nr};

INSERT {?p :nrOfFriends ?nr}
WHERE
  { SELECT ?p (COUNT(?f) AS ?nr)
    WHERE
      { ?p a :Person.
        ?p :friend ?f.
      }
    GROUP BY ?p
  }
```



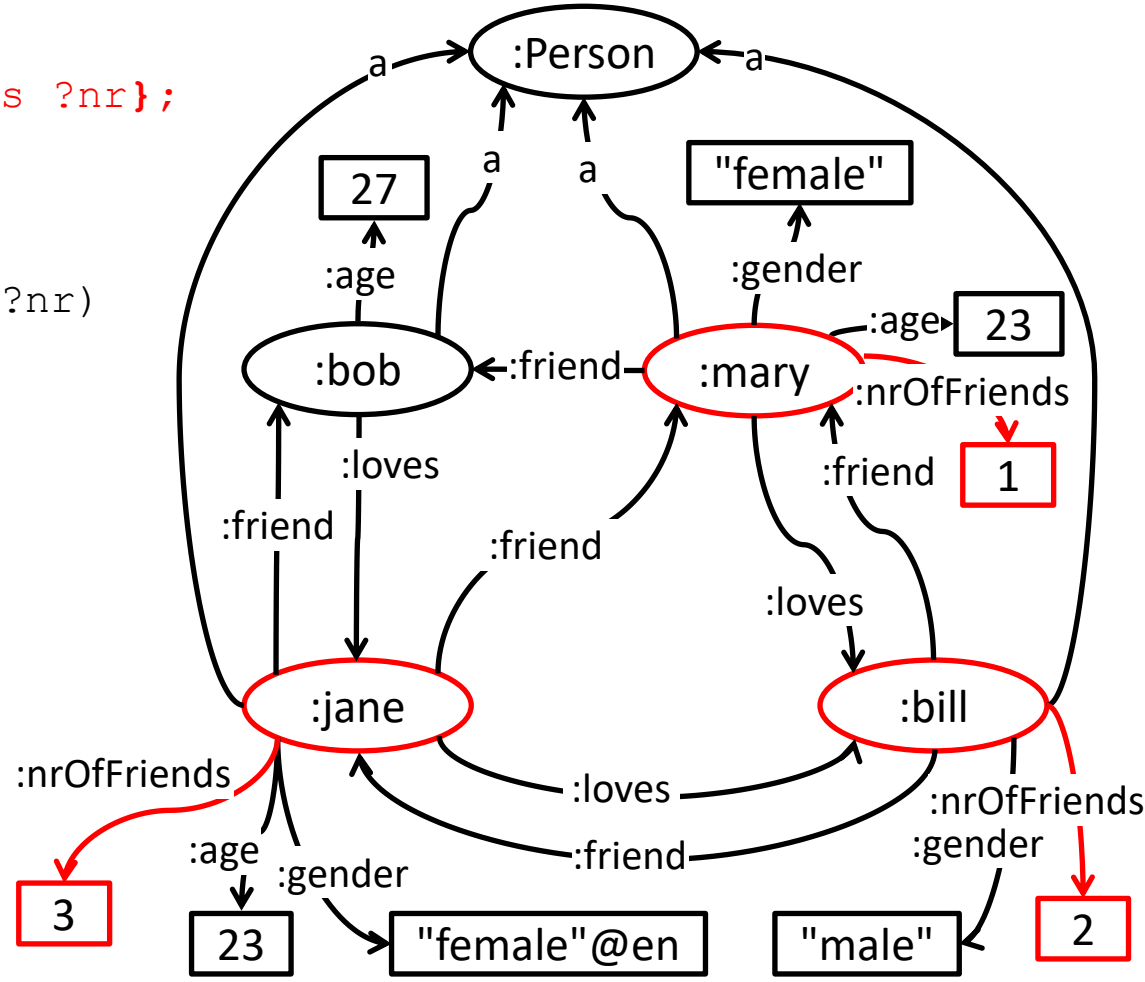
Update and Re-Calculation (3/6)

```
DELETE DATA
  { :jane :friend :bill };

DELETE WHERE {?p :nrOfFriends ?nr};

INSERT {?p :nrOfFriends ?nr}
WHERE
  { SELECT ?p (COUNT(?f) AS ?nr)
    WHERE
      { ?p a :Person.
        ?p :friend ?f.
      }
    GROUP BY ?p
  }
```



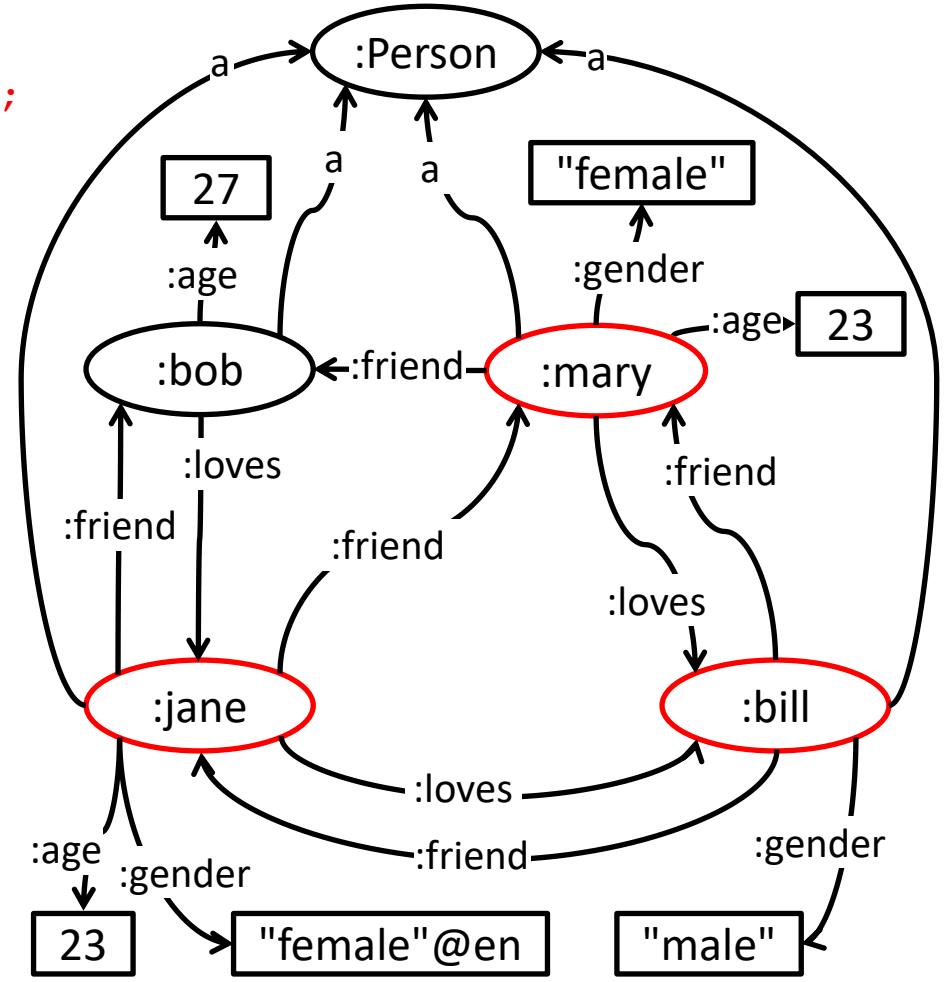
Update and Re-Calculation (4/6)

```
DELETE DATA
  { :jane :friend :bill };

DELETE WHERE {?p :nrOfFriends ?nr};

INSERT {?p :nrOfFriends ?nr}
WHERE
  { SELECT ?p (COUNT(?f) AS ?nr)
    WHERE
      { ?p a :Person.
        ?p :friend ?f.
      }
    GROUP BY ?p
  }
```



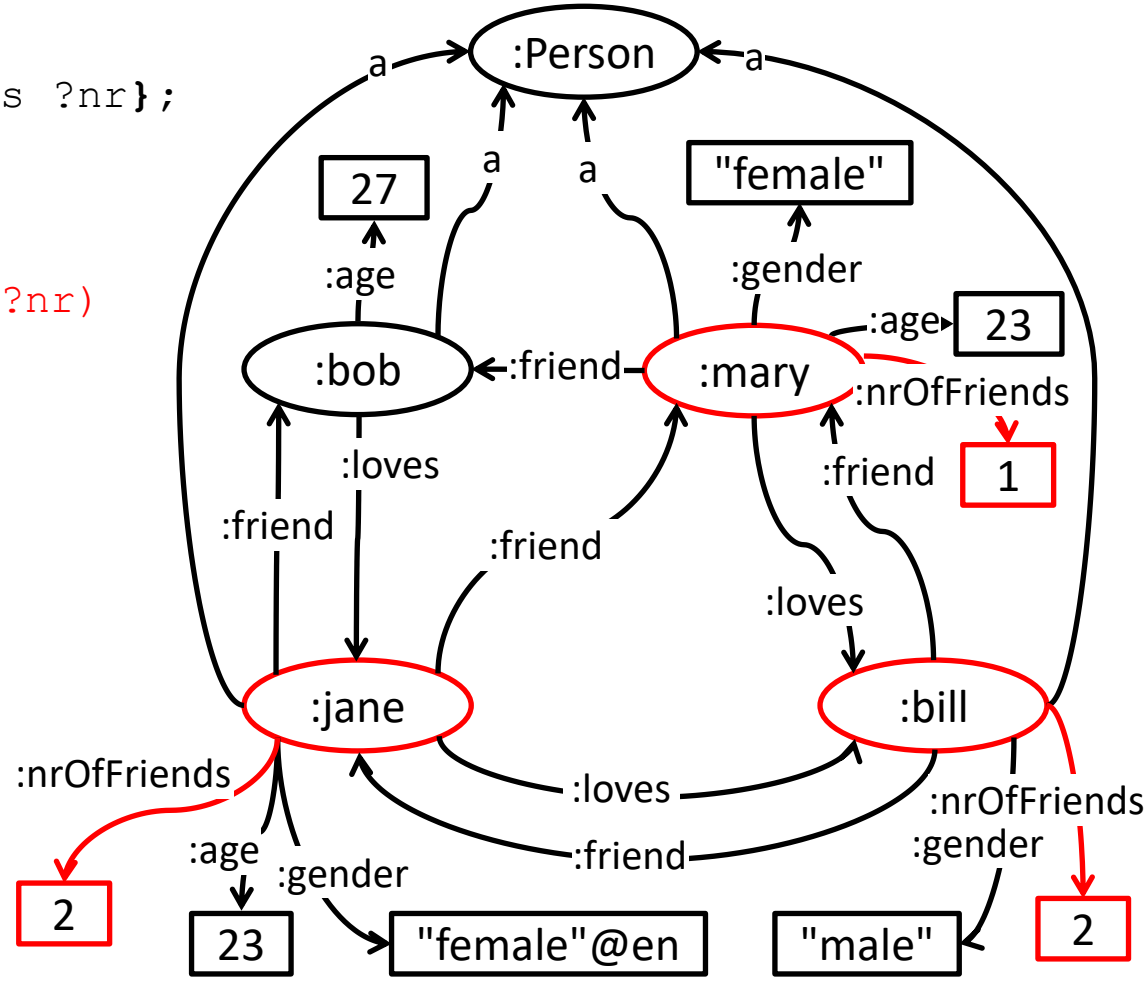
Update and Re-Calculation (5/6)

```
DELETE DATA
  { :jane :friend :bill };

DELETE WHERE {?p :nrOfFriends ?nr};

INSERT {?p :nrOfFriends ?nr}
WHERE
  { SELECT ?p (COUNT(?f) AS ?nr)
    WHERE
      { ?p a :Person.
        ?p :friend ?f.
      }
    GROUP BY ?p
  }
```



Update and Re-Calculation (6/6)

```
DELETE DATA
  { :jane :friend :bill };

DELETE WHERE {?p :nrOfFriends ?nr};

INSERT {?p :nrOfFriends ?nr}
WHERE
  { SELECT ?p (COUNT(?f) AS ?nr)
    WHERE
      { ?p a :Person.
        ?p :friend ?f.
      }
    GROUP BY ?p
  }
```



Query on Multiple Graphs

Querying RDF Datasets

RDF Datasets

An ***RDF dataset*** is a collection of RDF graphs, and comprises:

- Exactly **one *default graph***, being an RDF graph. The default graph does not have a name and *MAY* be empty.
- **Zero or more *named graphs***. Each named graph is a pair consisting of an IRI or a blank node (the *graph name*), and an RDF graph. Graph names are unique within an RDF dataset.
- Blank nodes can be shared between graphs in an RDF dataset.

see: http://www.w3.org/TR/rdf11-concepts/#section-dataset

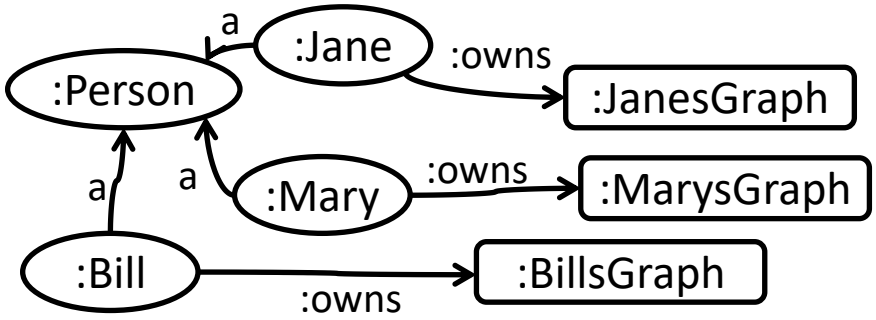
see: http://www.w3.org/TR/sparql11-query/#rdfDataset

RDF Dataset / Graph Store - Running Example

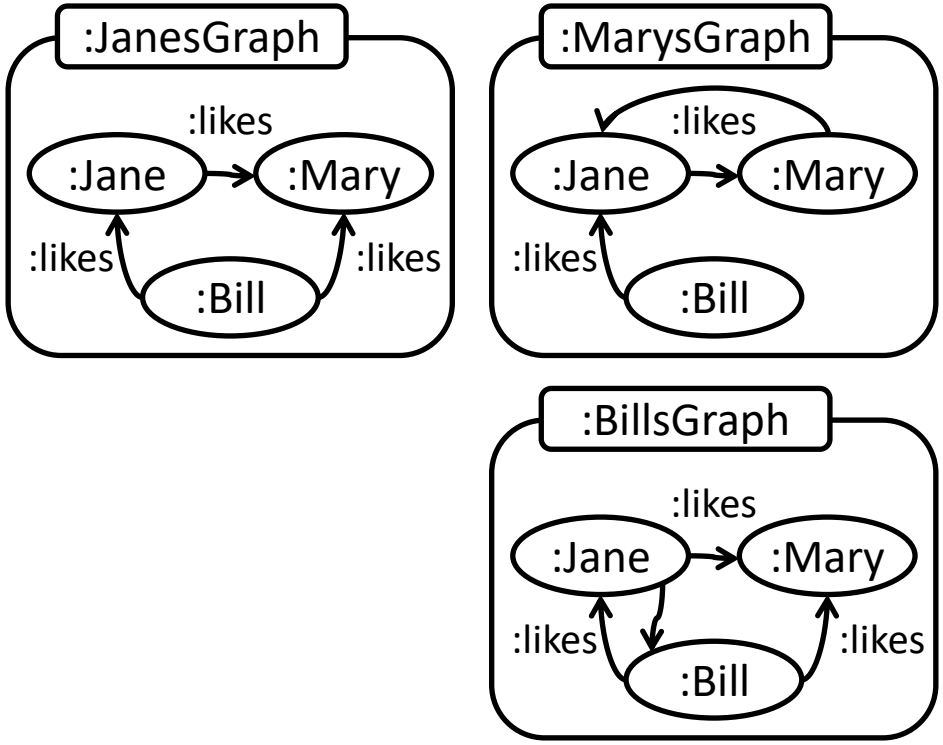
```
:Jane a :Person;
  :owns :JanesGraph.

:Mary a :Person;
  :owns :MarysGraph.

:Bill a :Person;
  :owns :BillsGraph.

:JanesGraph {
  :Jane :likes :Mary.
  :Bill :likes :Jane, :Mary.
}

:MarysGraph {
  :Jane :likes :Mary.
  :Bill :likes :Jane.
  :Mary :likes :Jane.
}

:BillsGraph {
  :Jane :likes :Mary, :Bill.
  :Bill :likes :Mary, :Jane.
}
```

RDF Dataset (aka Graph Store)

Default Graph

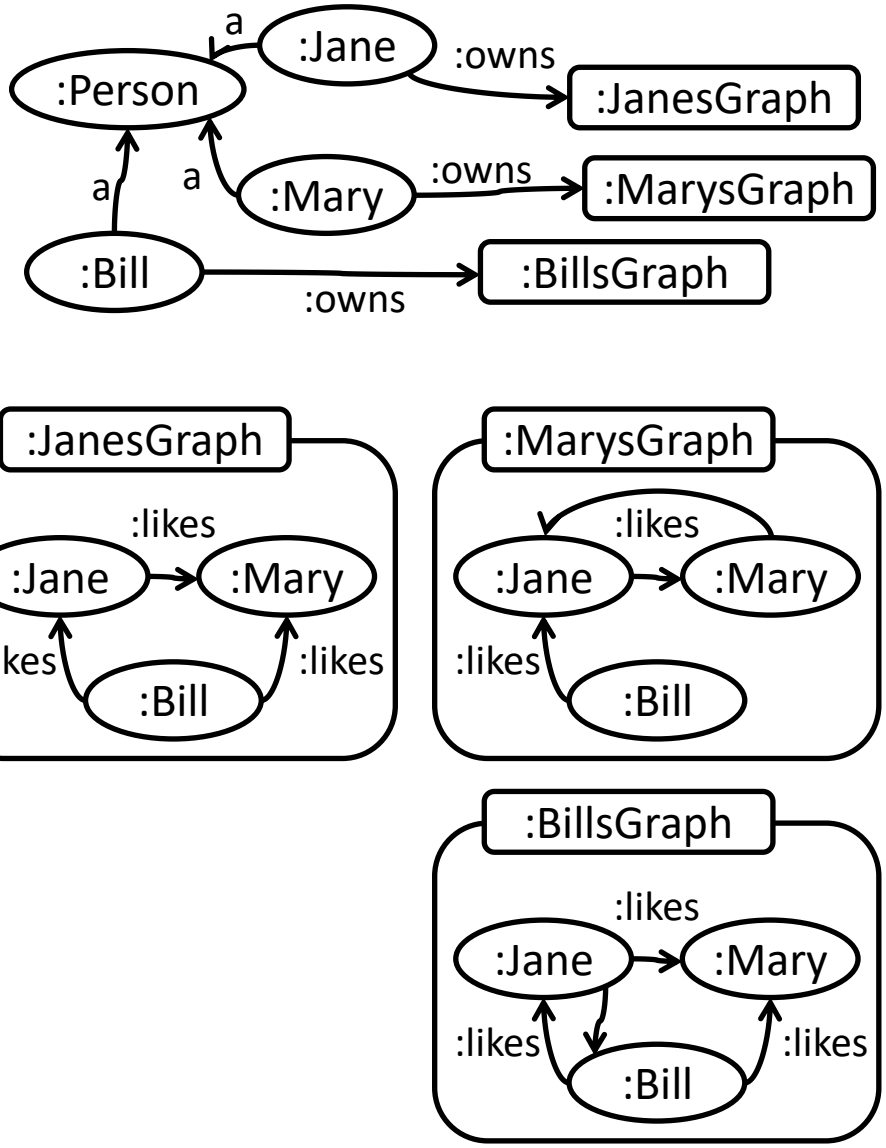


Named Graphs



Querying all Triples and **Quadruples** in a Dataset (1/2)

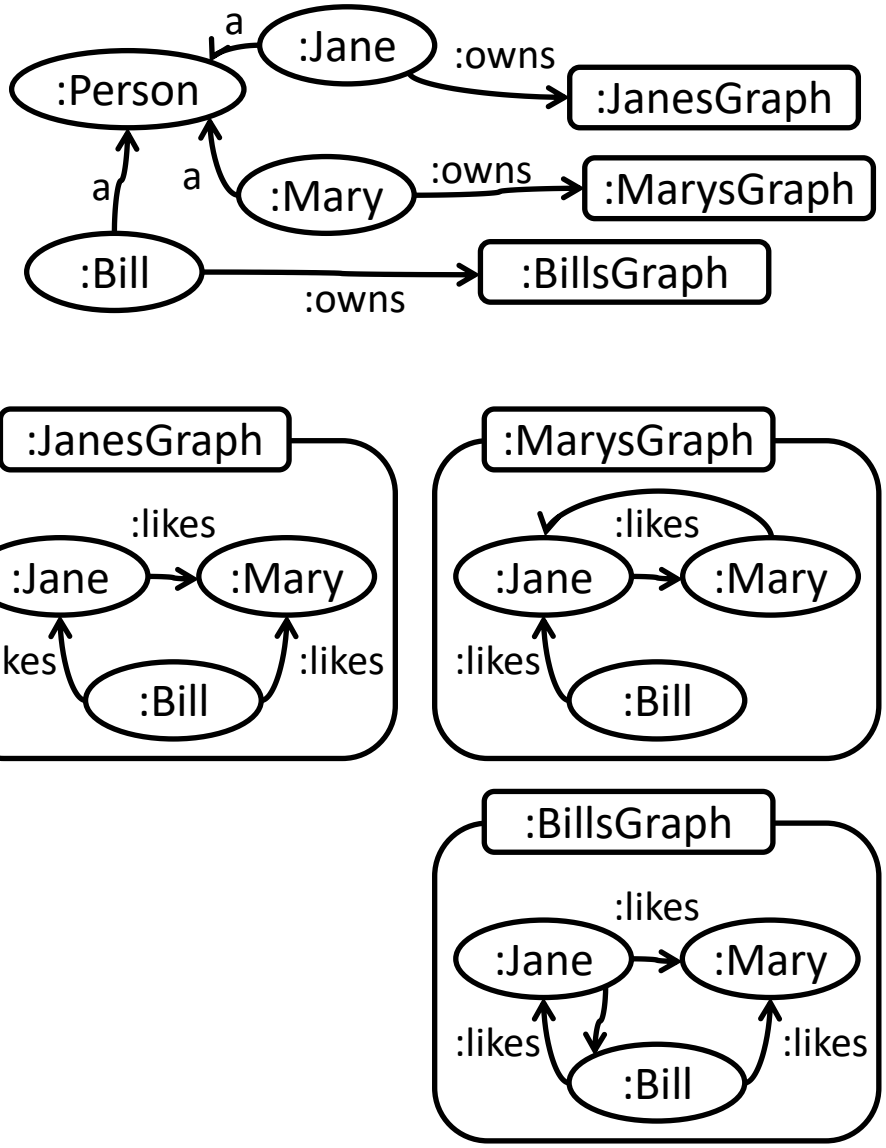
```
SELECT *
WHERE
  {  {?s ?p ?o}
    UNION
     { GRAPH ?g
         {?s ?p ?o}
     }
  }
```



Querying all Triples and **Quadruples** in a Dataset (2/2)

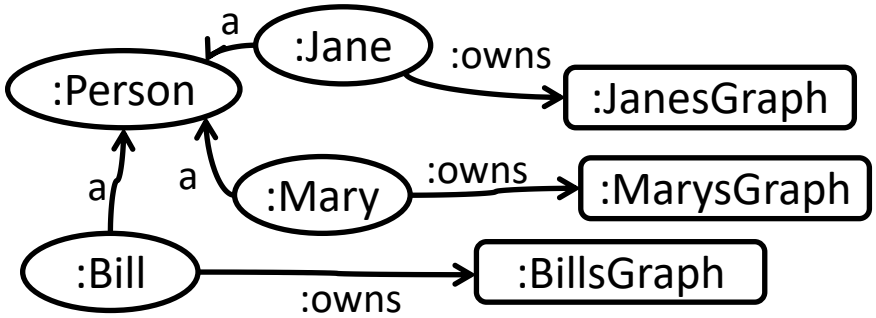
```
SELECT *
WHERE
  {  {?s ?p ?o}
    UNION
     { GRAPH ?g
         {?s ?p ?o}
     }
  }
```

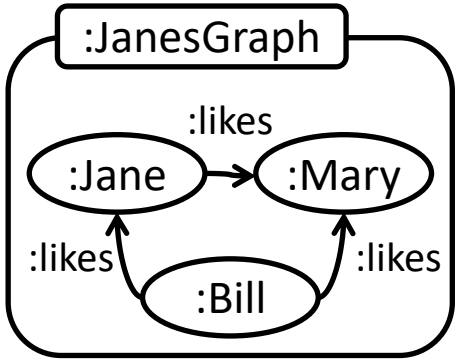
s	p	o	g
:Mary	:owns	:MarysGraph	
:Mary	rdf:type	:Person	
:Jane	:owns	:JanesGraph	
:Jane	rdf:type	:Person	
:Bill	:owns	:BillsGraph	
:Bill	rdf:type	:Person	
:Jane	:likes	:Mary	:JanesGraph
:Bill	:likes	:Mary	:JanesGraph
:Bill	:likes	:Jane	:JanesGraph
:Mary	:likes	:Jane	:MarysGraph
:Jane	:likes	:Mary	:MarysGraph
:Bill	:likes	:Jane	:MarysGraph
:Jane	:likes	:Bill	:BillsGraph
:Jane	:likes	:Mary	:BillsGraph
:Bill	:likes	:Jane	:BillsGraph
:Bill	:likes	:Mary	:BillsGraph

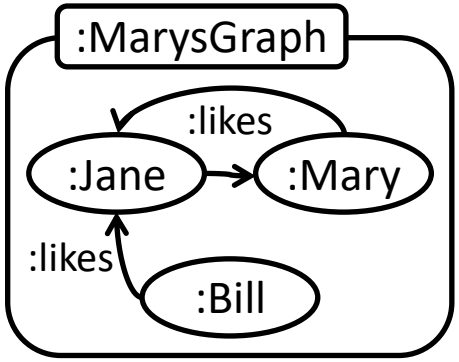


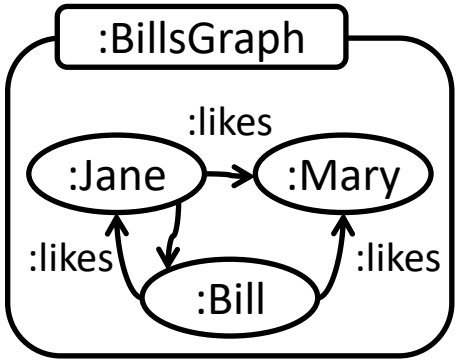
Querying a specific Named Graph (1/2)

```
SELECT *
WHERE
  {  GRAPH :JanesGraph
        {:Bill :likes ?o}

  }
```





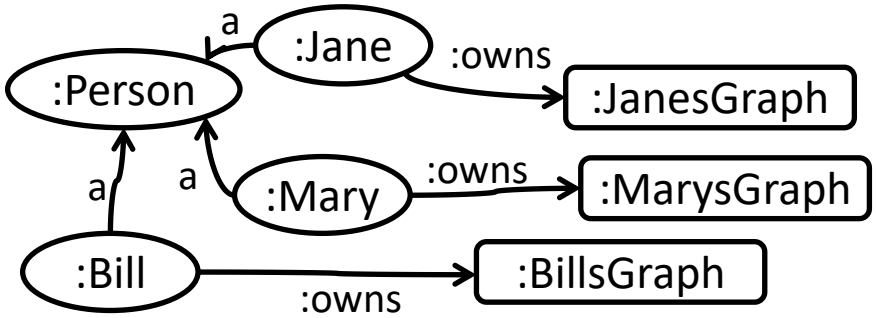


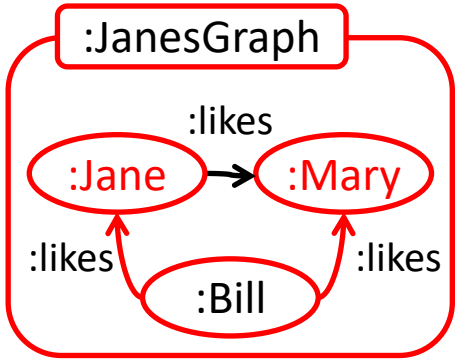


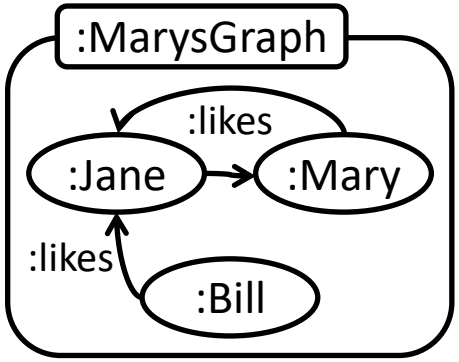
Querying a specific Named Graph (2/2)

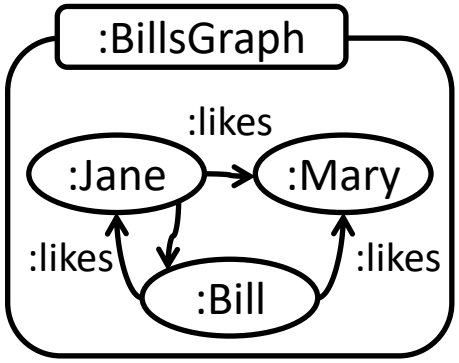
```
SELECT *
WHERE
  {  GRAPH :JanesGraph
       {:Bill :likes ?o}

  }
```

```
---------
| o     |
=========
| :Mary |
| :Jane |
---------
```



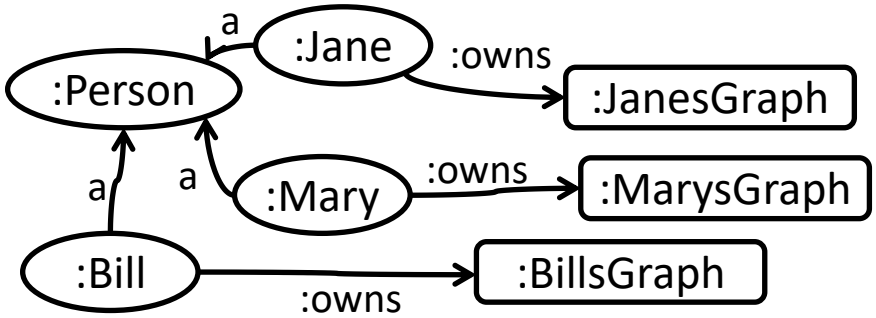


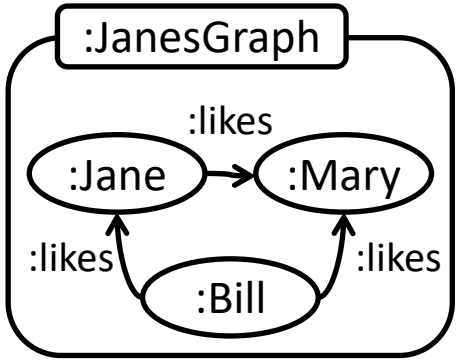


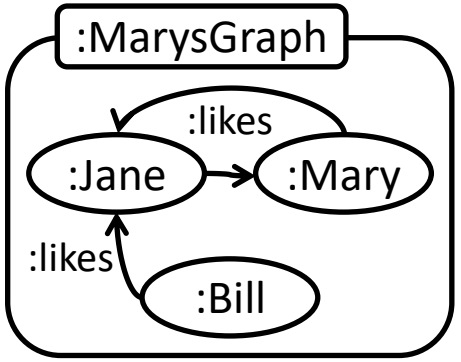


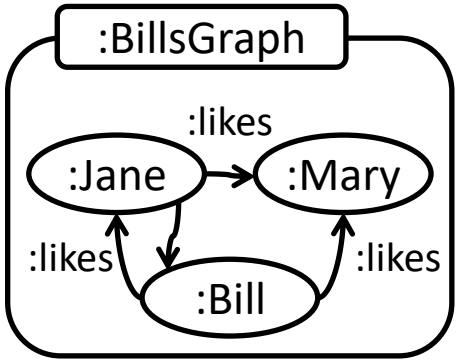
Intersection/Join of Named Graphs (1/2)

```
SELECT DISTINCT *
WHERE
  {  GRAPH :JanesGraph
        {?s ?p ?o}
     GRAPH :MarysGraph
        {?s ?p ?o}
  }
```





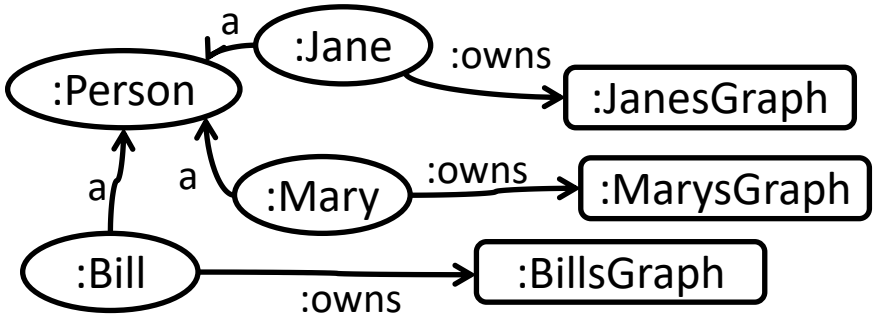


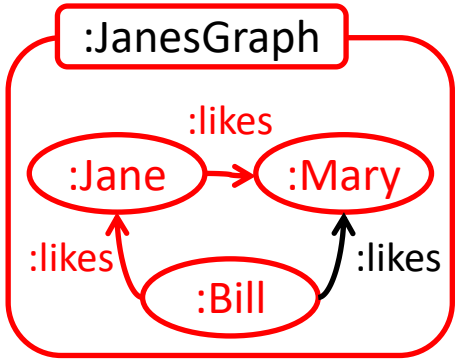


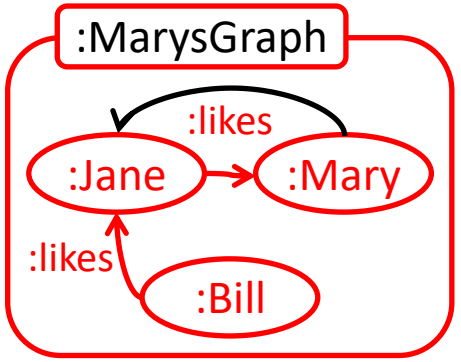
Intersection/Join of Named Graphs (2/2)

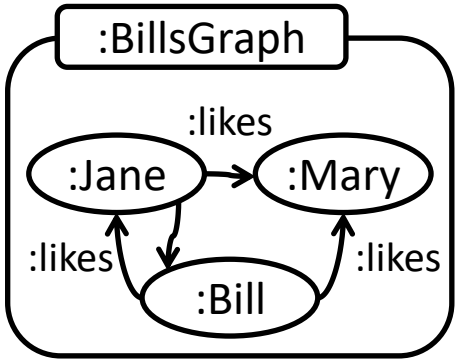
```
SELECT DISTINCT *
WHERE
  {  GRAPH :JanesGraph
         {?s ?p ?o}
     GRAPH :MarysGraph
         {?s ?p ?o}
  }
```

s	p	o
:Jane	:likes	:Mary
:Bill	:likes	:Jane



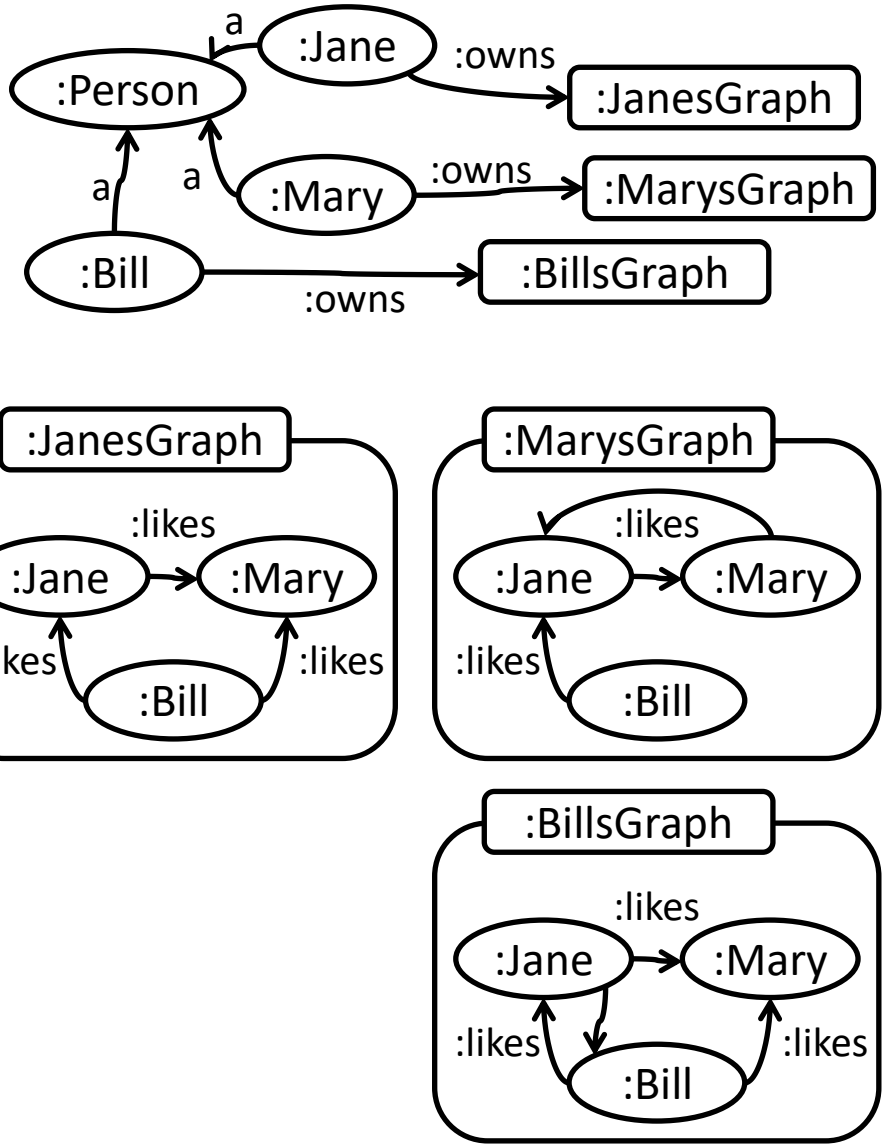






Union of Named Graphs (1/3)

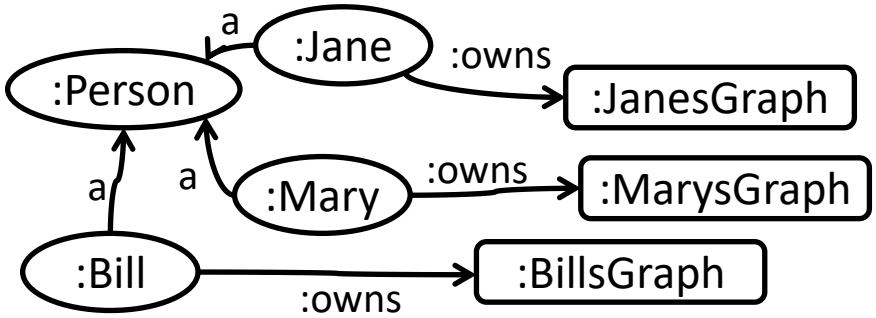
```
SELECT DISTINCT *
WHERE
  {   { GRAPH :JanesGraph
          {?s ?p ?o}
      }
      UNION
      { GRAPH :MarysGraph
          {?s ?p ?o}
      }
  }
```

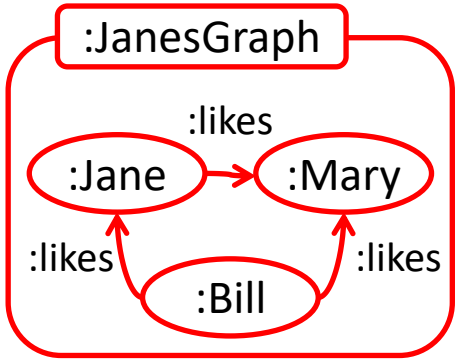


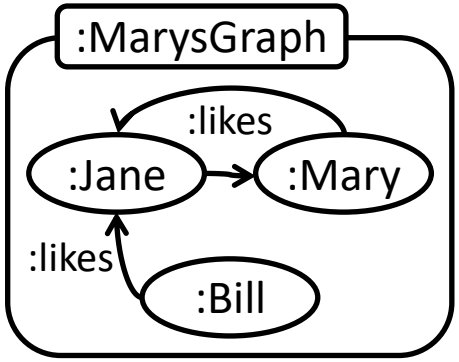
Union of Named Graphs (2/3)

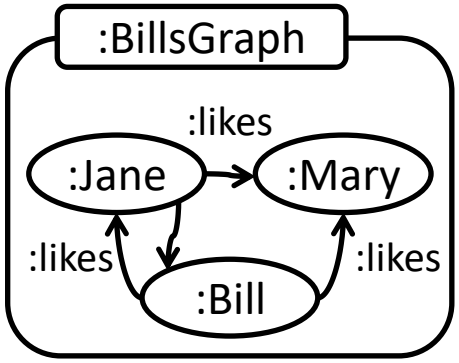
```
SELECT DISTINCT *
WHERE
  {   { GRAPH :JanesGraph
          {?s ?p ?o}
      }
      UNION
      { GRAPH :MarysGraph
          {?s ?p ?o}
      }
  }
```

```
-------------------------
| s     | p       | o     |
=========================
| :Jane | :likes  | :Mary |
| :Bill | :likes  | :Mary |
| :Bill | :likes  | :Jane |
-------------------------
```





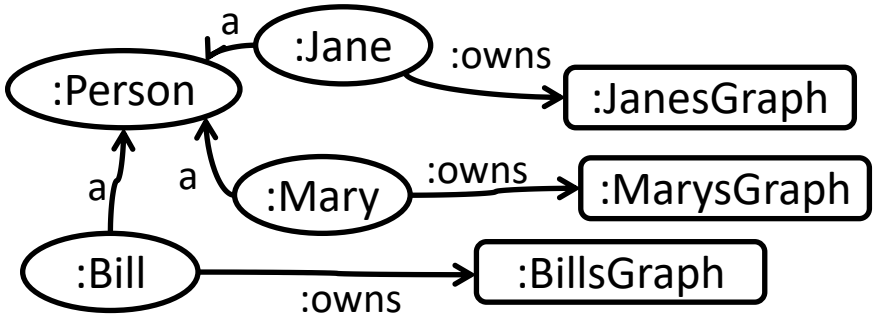


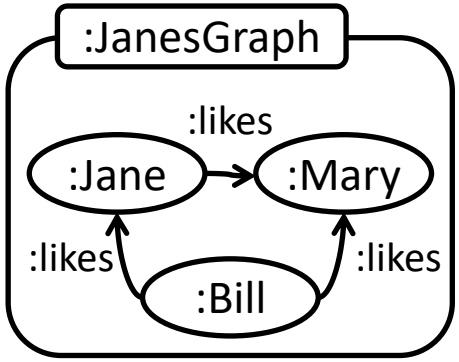


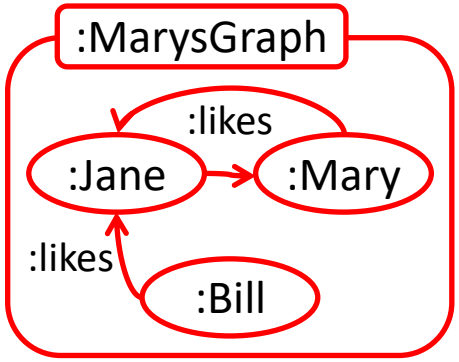
Union of Named Graphs (3/3)

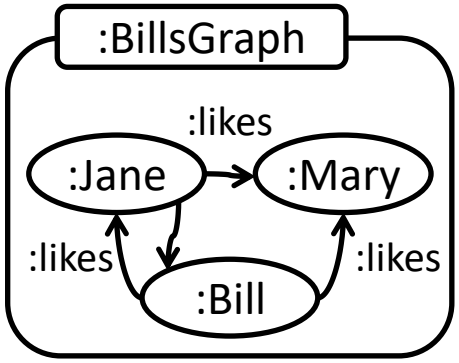
```
SELECT DISTINCT *
WHERE
  {   { GRAPH :JanesGraph
          {?s ?p ?o}
      }
      UNION
      { GRAPH :MarysGraph
          {?s ?p ?o}
      }
  }
```

s	p	o
:Jane	:likes	:Mary
:Bill	:likes	:Mary
:Bill	:likes	:Jane
:Mary	:likes	:Jane



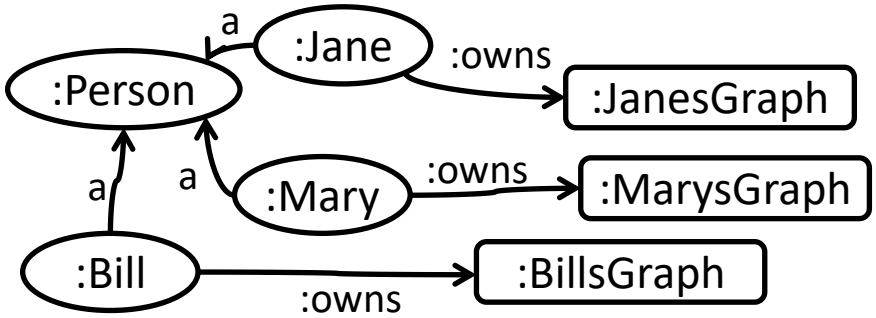


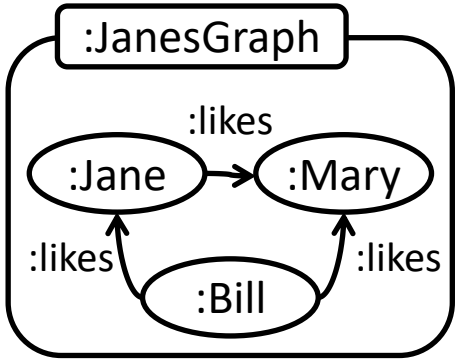


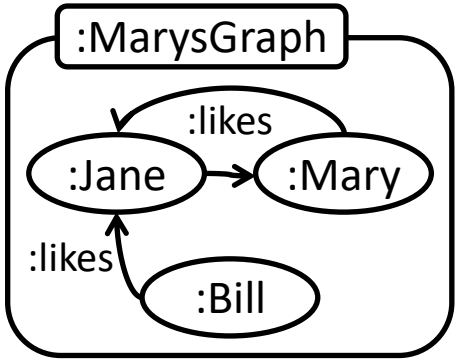


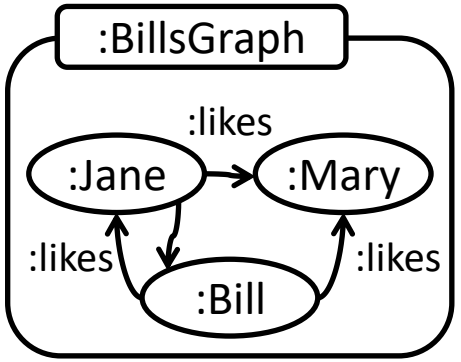
Querying graphs with a description fulfilling a given condition (1/3)

```
SELECT *
WHERE
  { :Jane :owns ?g.
    GRAPH ?g
          {:Bill :likes ?o}

  }
```



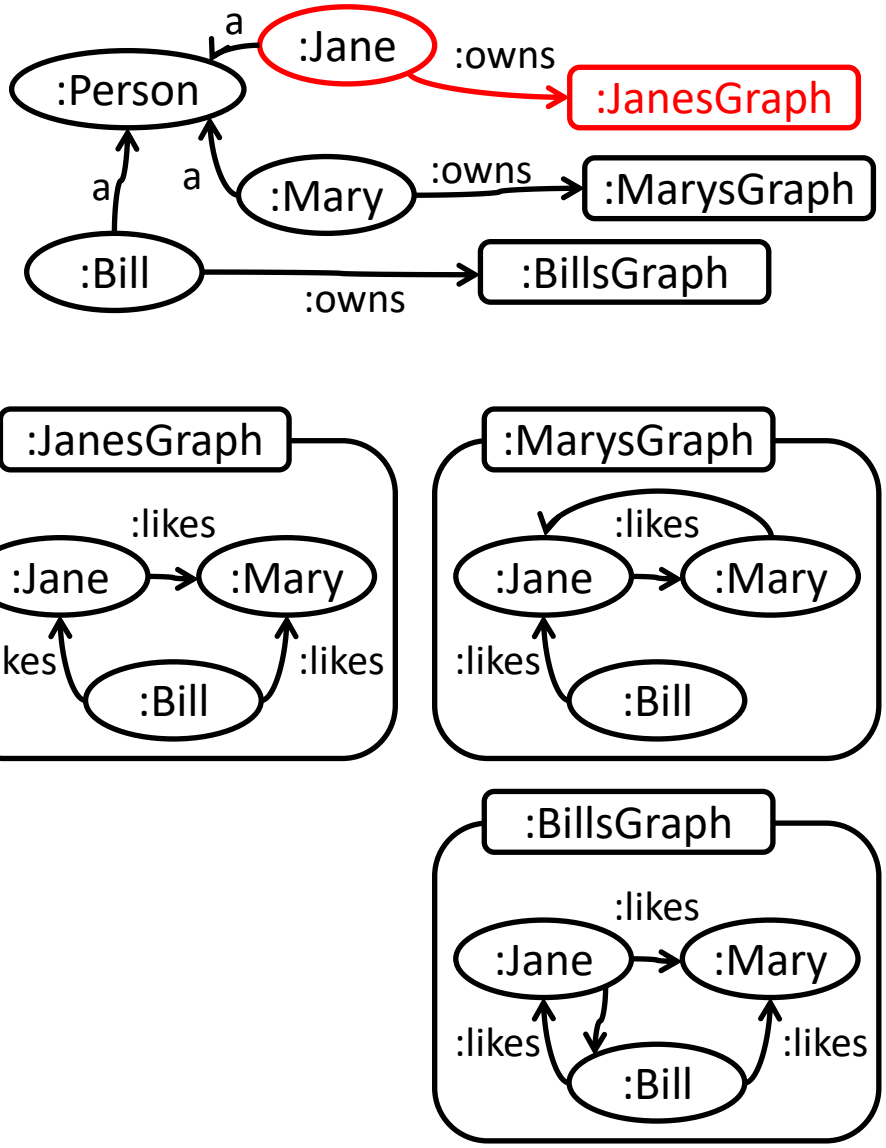






Querying graphs with a description fulfilling a given condition (2/3)

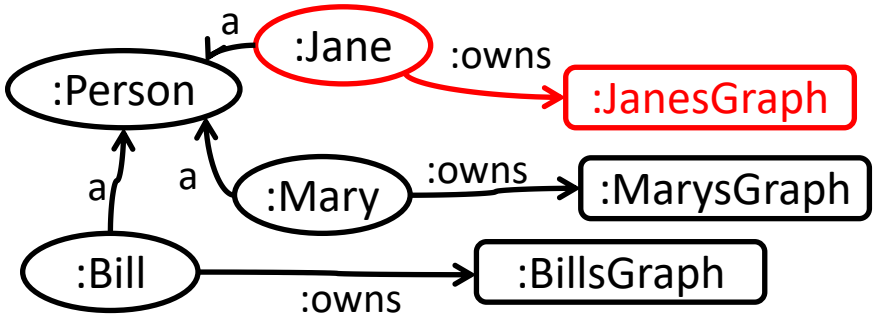
```
SELECT *
WHERE
  { :Jane :owns ?g.
    GRAPH ?g
          {:Bill :likes ?o}

  }
```

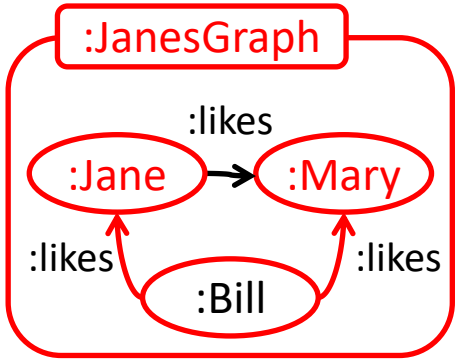


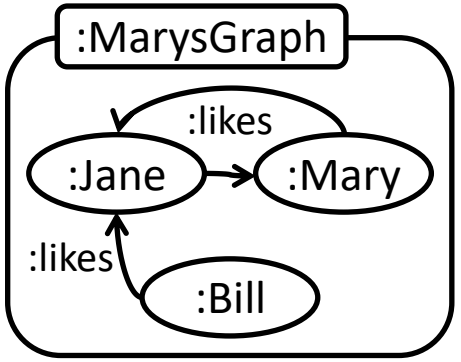
Querying graphs with a description fulfilling a given condition (3/3)

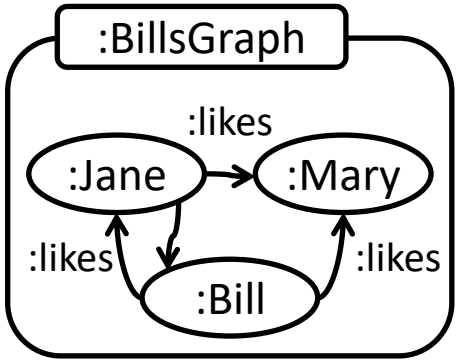
```
SELECT *
WHERE
  { :Jane :owns ?g.
    GRAPH ?g
          {:Bill :likes ?o}

  }
```

g	o
:JanesGraph	:Mary
:JanesGraph	:Jane





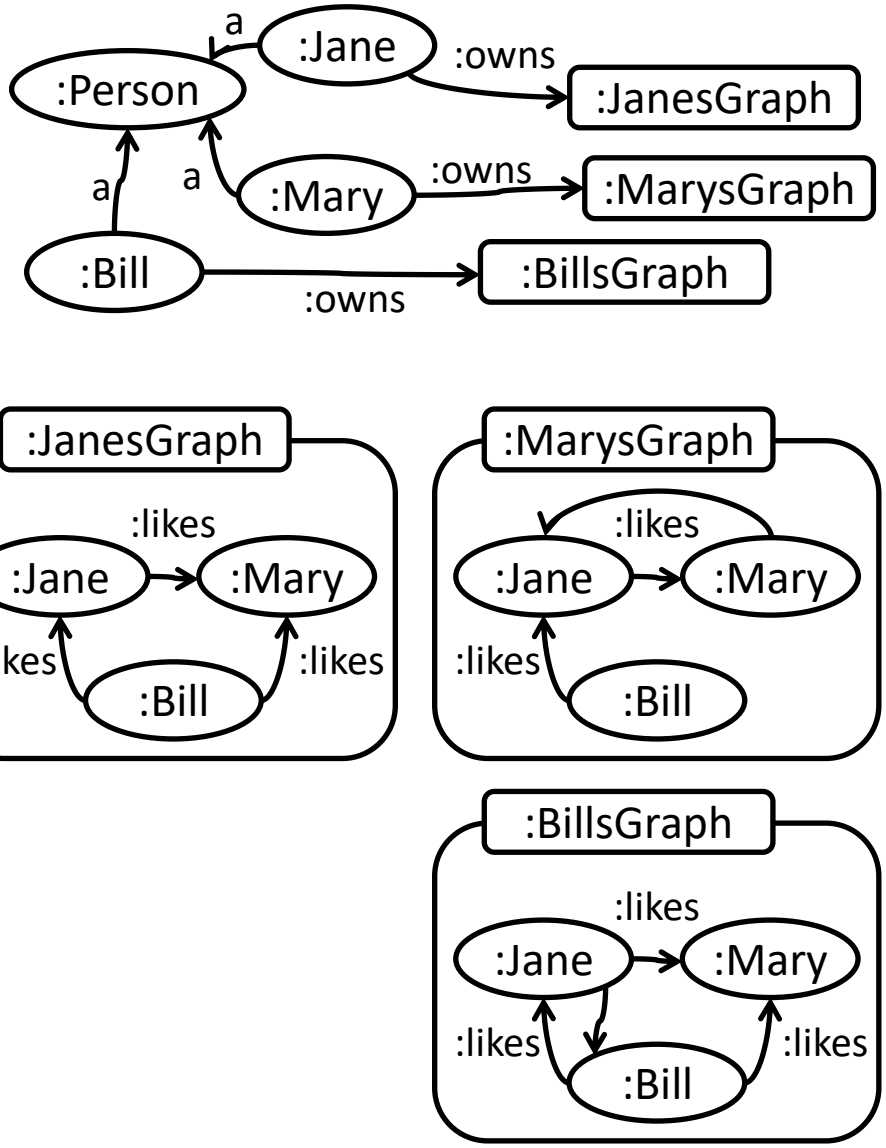




Querying graphs with a description fulfilling a given condition (1/3)

Union of all person-owned graphs

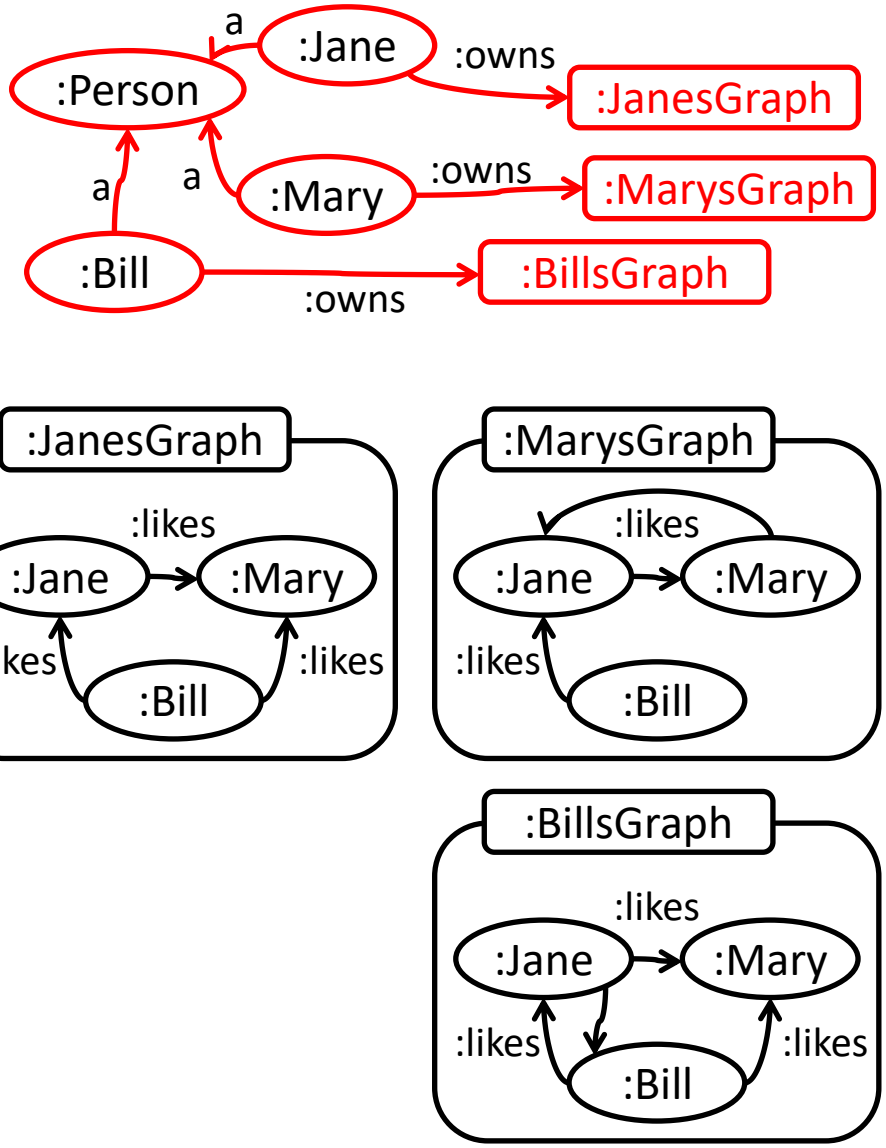
```
SELECT DISTINCT ?s ?p ?o
WHERE
  {   [] a :Person; :owns ?g.

      GRAPH ?g
          {?s ?p ?o}
  }
```



Querying graphs with a description fulfilling a given condition (2/3)

Union of all person-owned graphs

```
SELECT DISTINCT ?s ?p ?o
WHERE
  {   [] a :Person; :owns ?g.

      GRAPH ?g
          {?s ?p ?o}
  }
```



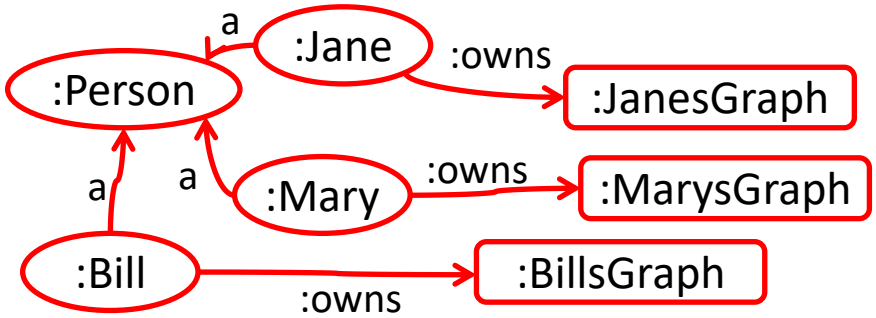
Querying graphs with a description fulfilling a given condition (3/3)

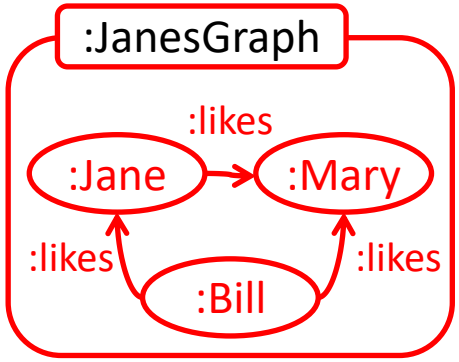
Union of all person-owned graphs

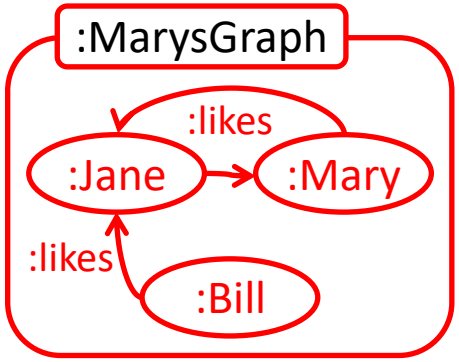
```
SELECT DISTINCT ?s ?p ?o
WHERE
  {   [] a :Person; :owns ?g.

      GRAPH ?g
          {?s ?p ?o}
  }
```

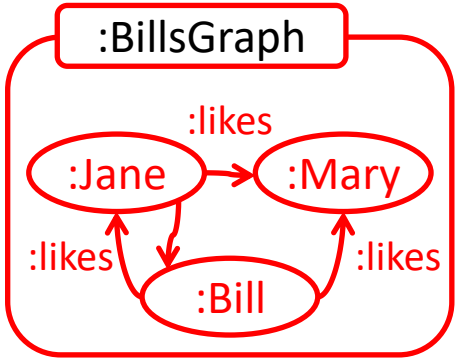
s	p	o
:Bill	:likes	:Jane
:Bill	:likes	:Mary
:Mary	:likes	:Jane
:Jane	:likes	:Mary

why not
:Jane :likes :Bill
?





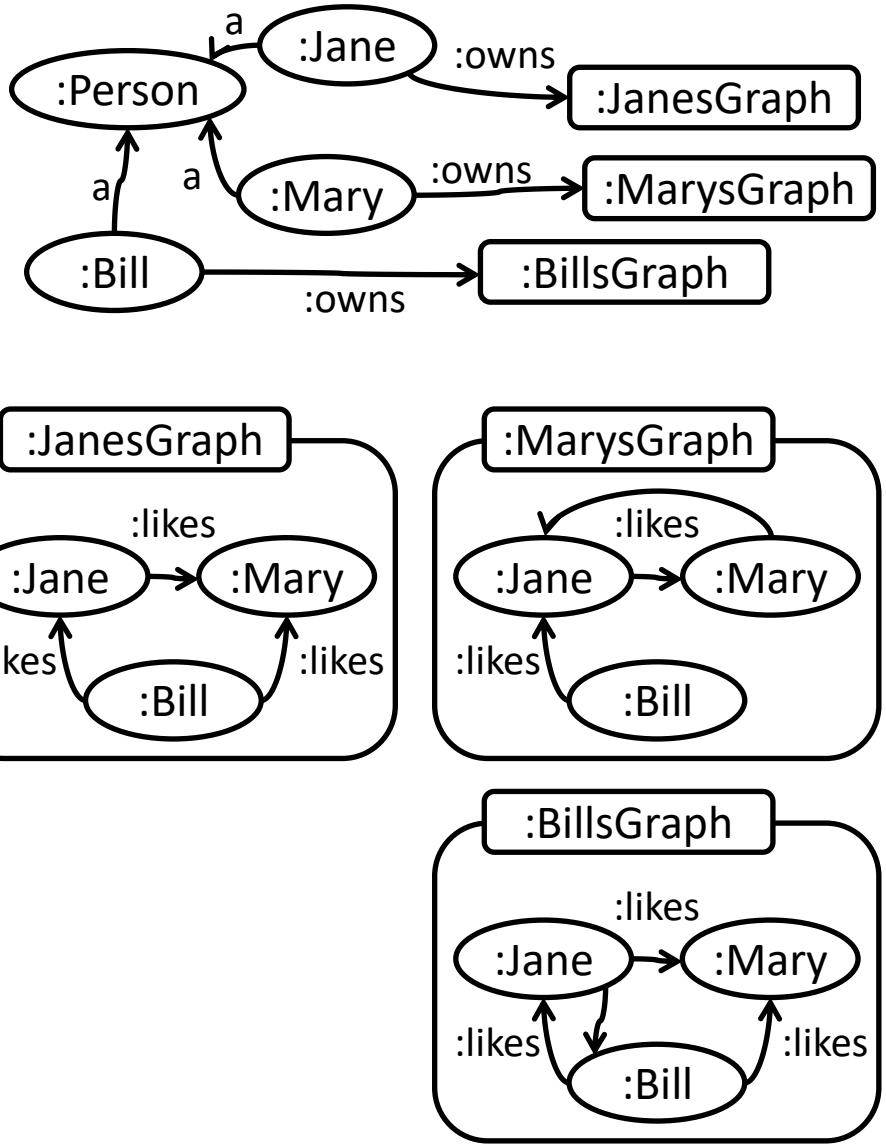




Correlating inner and outer queries (1/4)

How do the owners of graphs describe themselves?

```
SELECT ?s ?p ?o
WHERE
  {  ?s a :Person;
        :owns ?g.
     GRAPH ?g
        {?s ?p ?o}
  }
```

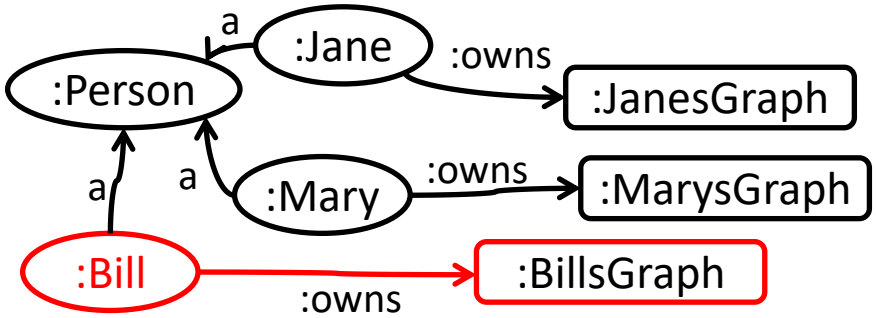


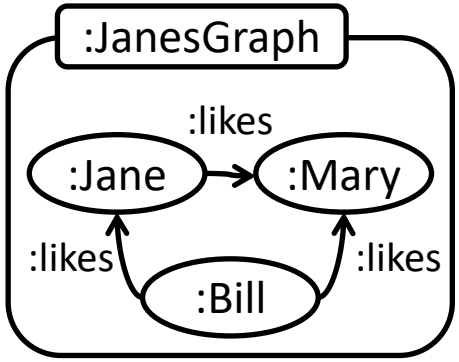
Correlating inner and outer queries (2/4)

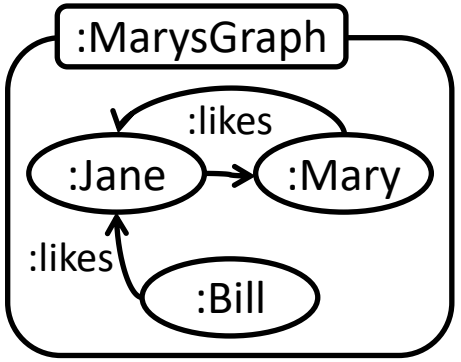
How do the owners of graphs describe themselves?

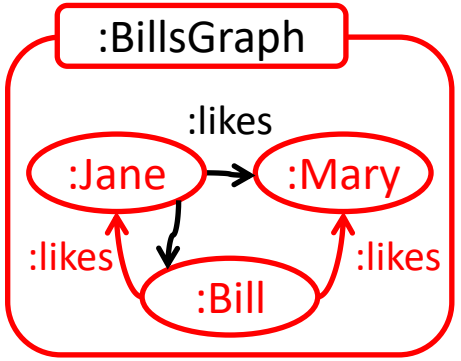
```
SELECT ?s ?p ?o
WHERE
  {  ?s a :Person;
        :owns ?g.
     GRAPH ?g
        {?s ?p ?o}
  }
```

s	p	o
:Bill	:likes	:Jane
:Bill	:likes	:Mary







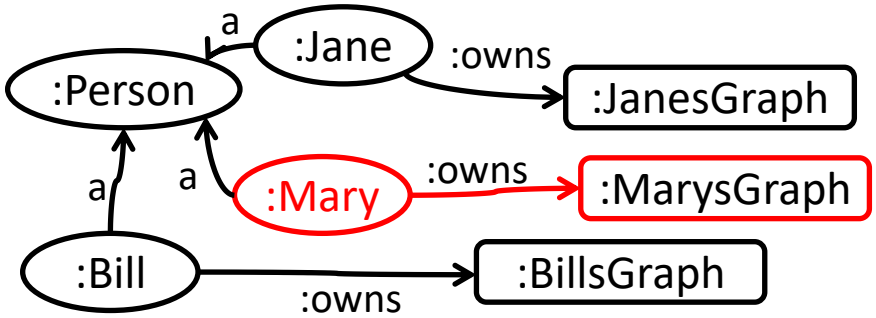


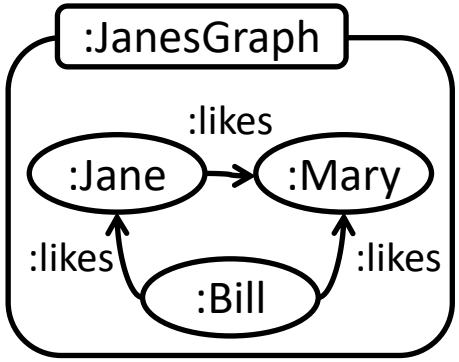
Correlating inner and outer queries (3/4)

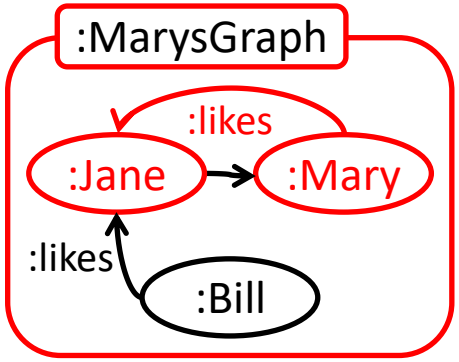
How do the owners of graphs describe themselves?

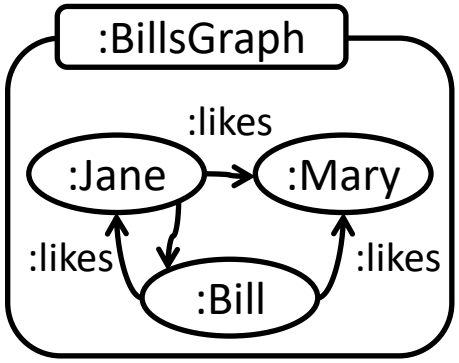
```
SELECT ?s ?p ?o
WHERE
  {  ?s a :Person;
        :owns ?g.
     GRAPH ?g
        {?s ?p ?o}
  }
```

s	p	o
:Bill	:likes	:Jane
:Bill	:likes	:Mary
:Mary	:likes	:Jane







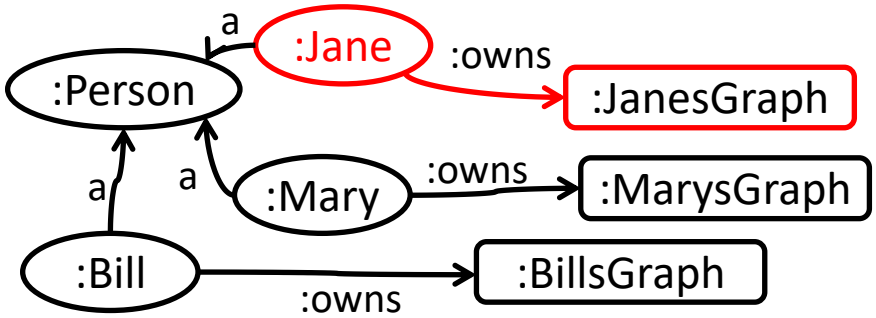


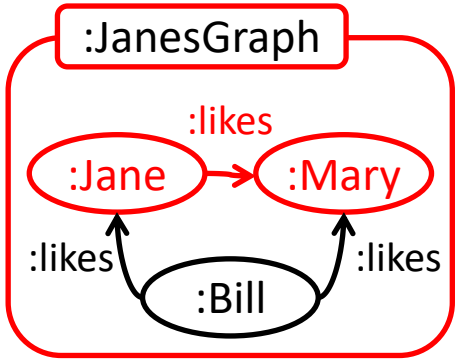
Correlating inner and outer queries (4/4)

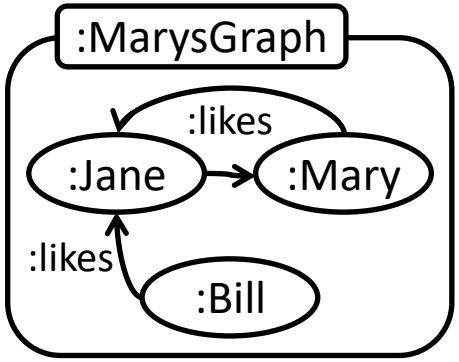
How do the owners of graphs describe themselves?

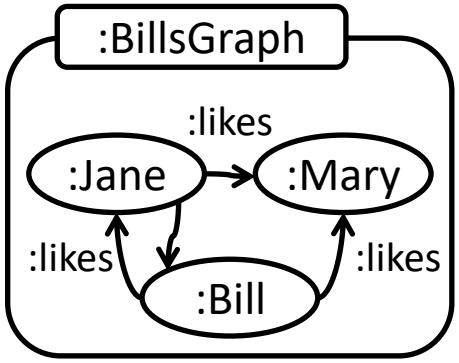
```
SELECT ?s ?p ?o
WHERE
  {  ?s a :Person;
        :owns ?g.
     GRAPH ?g
        {?s ?p ?o}
  }
```

s	p	o
:Bill	:likes	:Jane
:Bill	:likes	:Mary
:Mary	:likes	:Jane
:Jane	:likes	:Mary









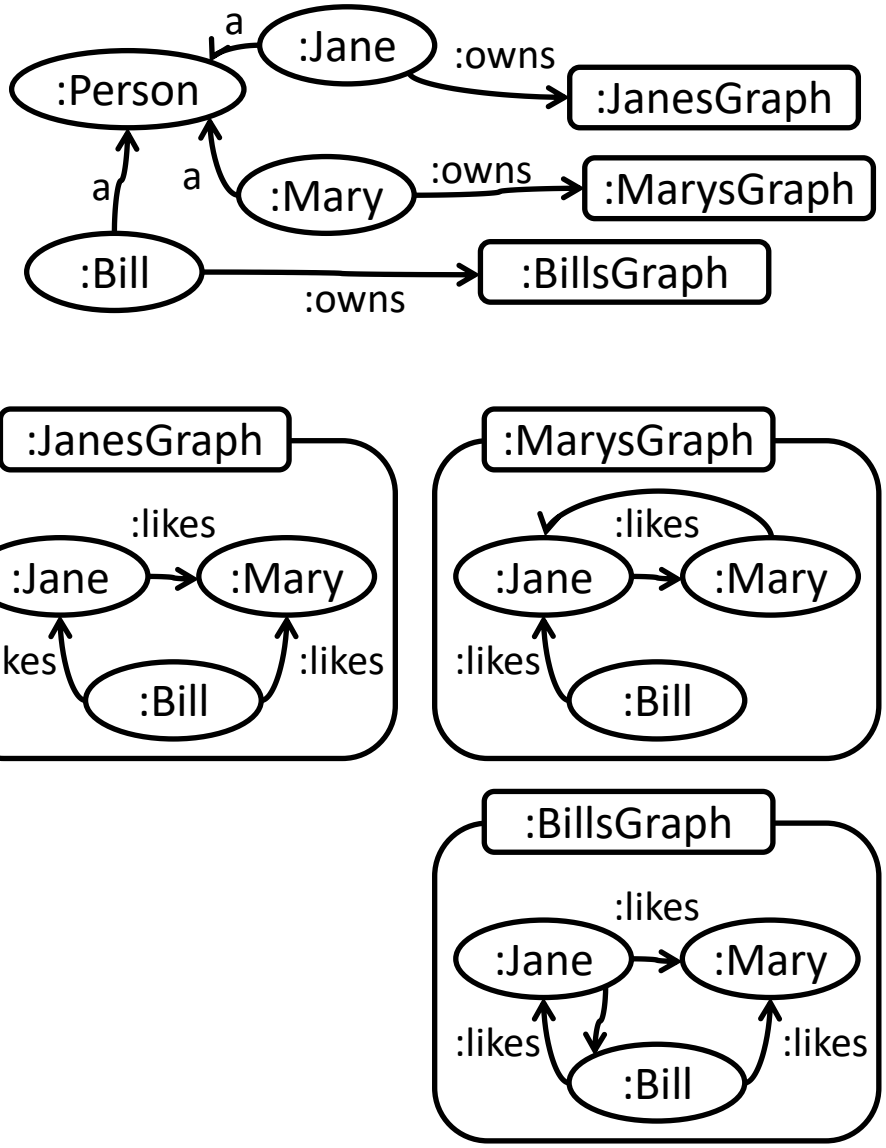
Update on Multiple Graphs

Update the Dataset (Graph Store)

see http://www.w3.org/TR/sparql11-update/

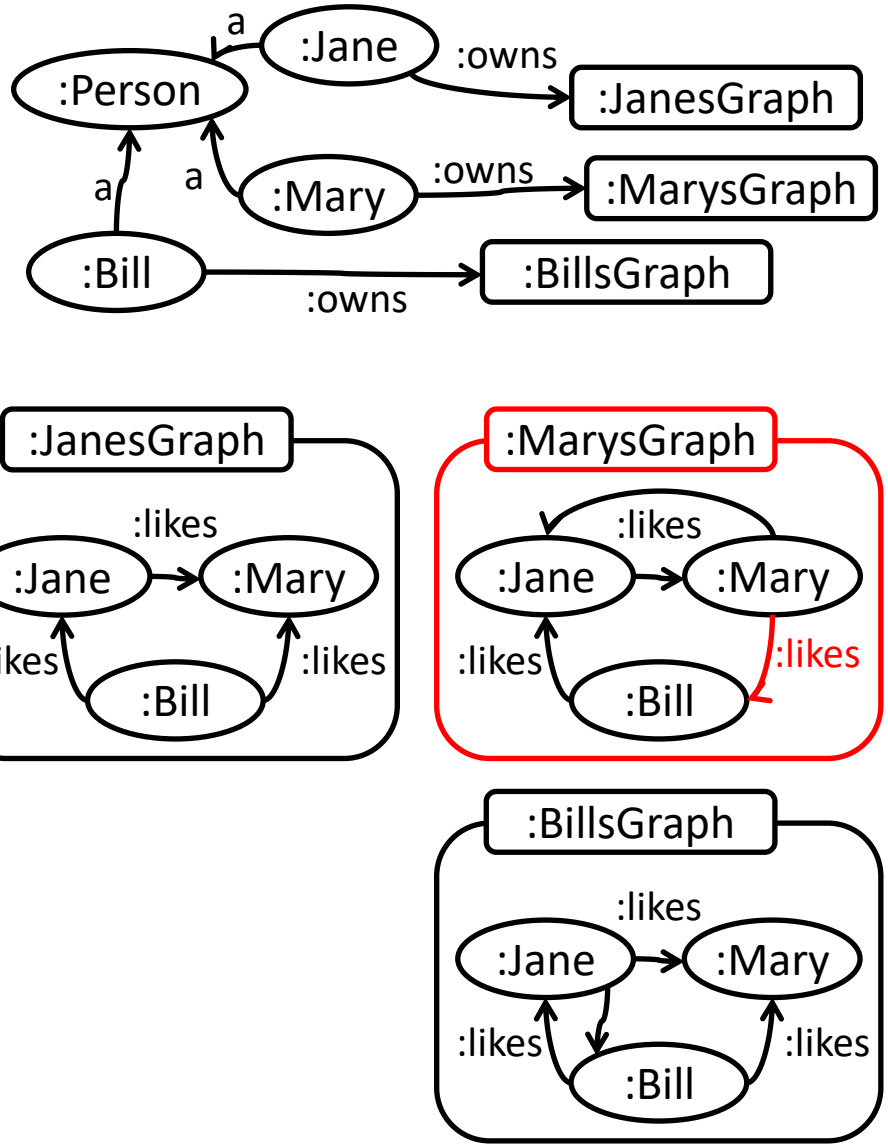
Insert Data into a Named Graph (1/2)

```
INSERT DATA {
  GRAPH :MarysGraph {
    :Mary :likes :Bill.
  }
}
```



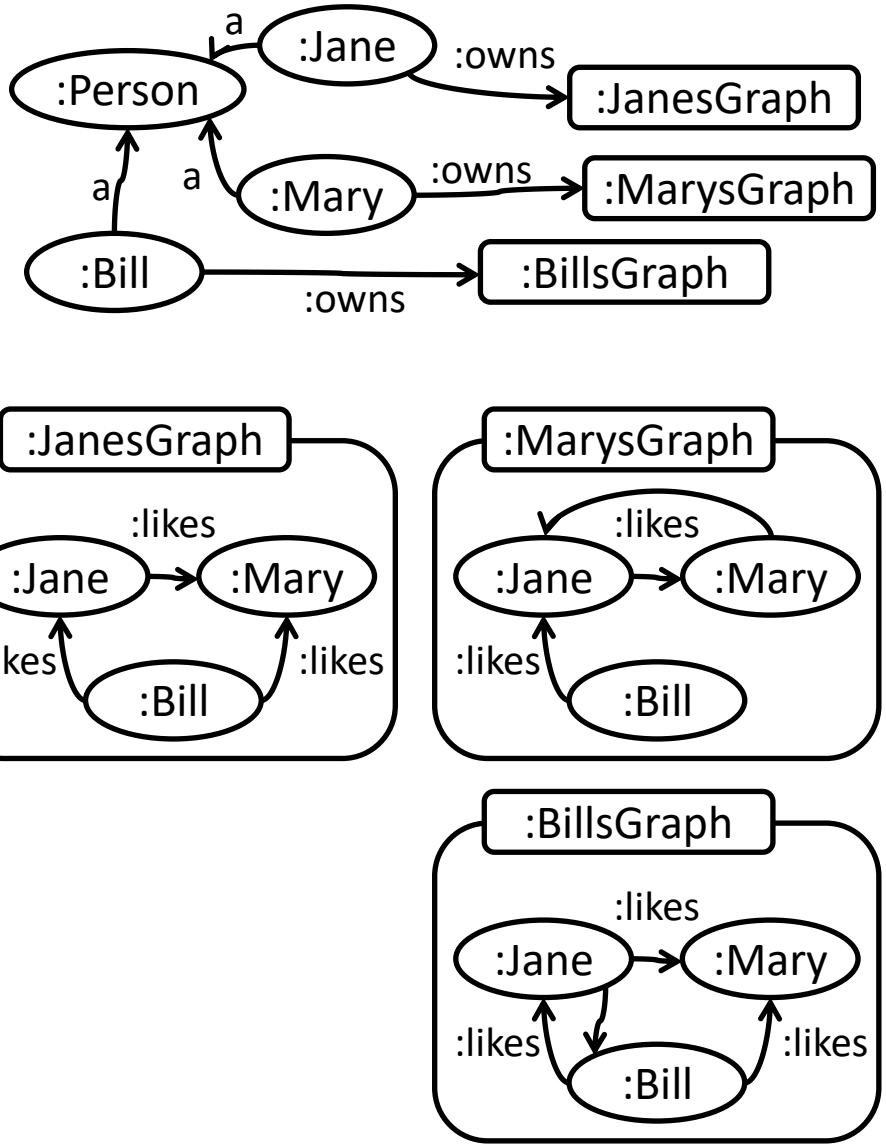
Insert Data into a Named Graph (2/2)

```
INSERT DATA {
  GRAPH :MarysGraph {
    :Mary :likes :Bill.
  }
}
```



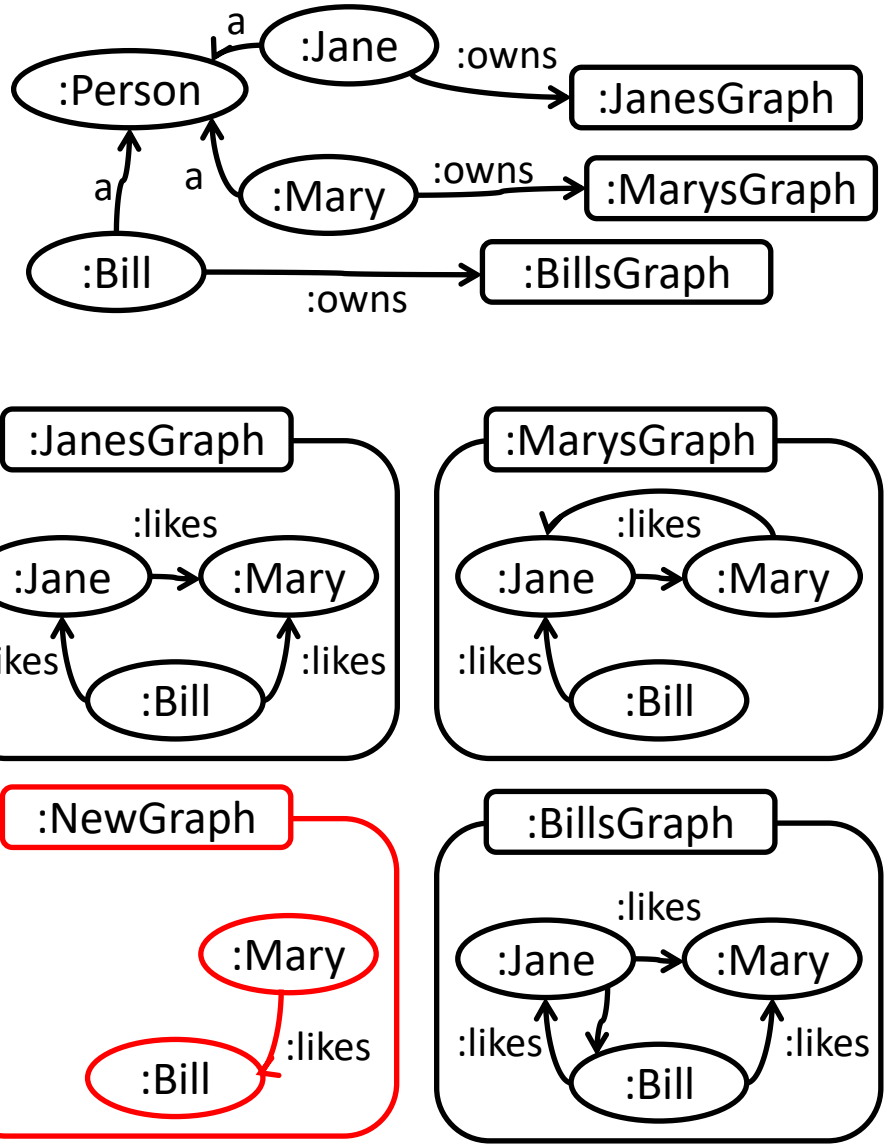
Insert Data into a new Named Graph (1/2)

```
INSERT DATA {
  GRAPH :NewGraph {
    :Mary :likes :Bill.
  }
}
```



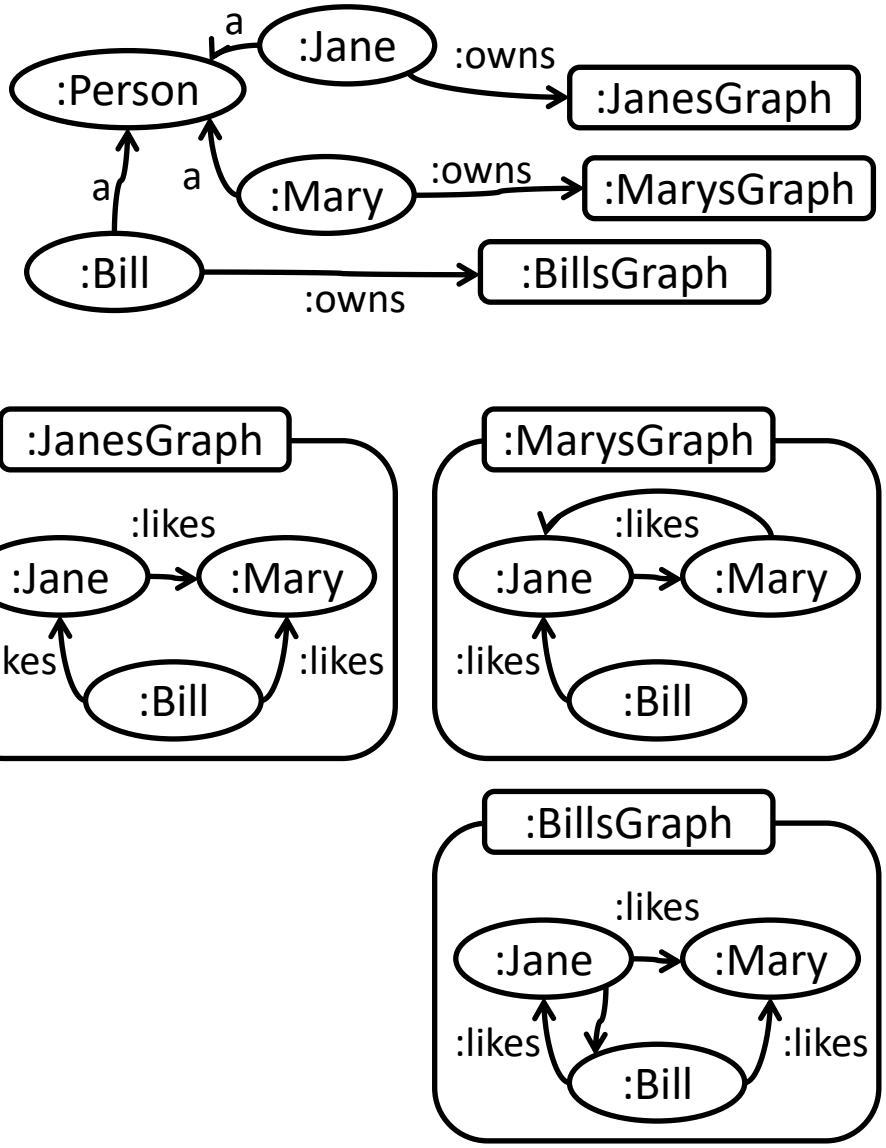
Insert Data into a new Named Graph (2/2)

```
INSERT DATA {
  GRAPH :NewGraph {
    :Mary :likes :Bill.
  }
}
```



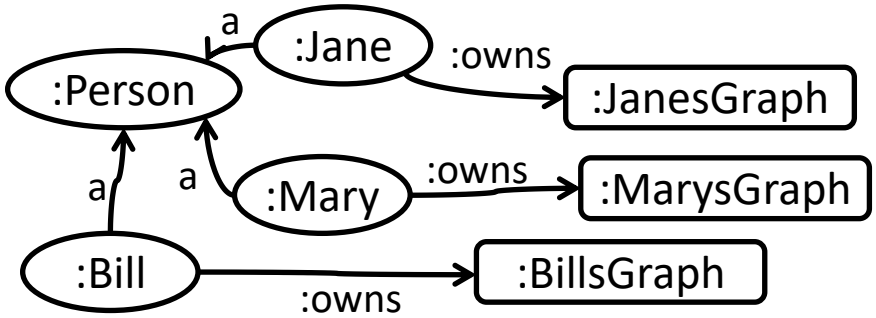
"Cut and Paste" (1/5)

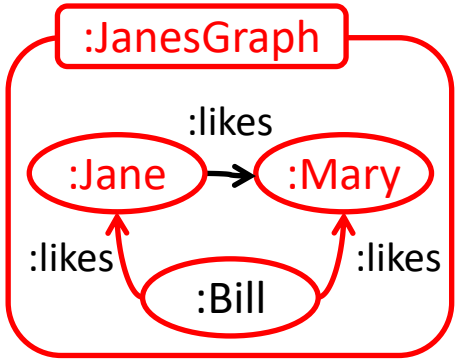
```
DELETE {
  GRAPH ?g
    { :Bill :likes ?o. } }
INSERT {
  GRAPH :BGraph
    { :Bill :likes ?o. } }
WHERE {
  GRAPH ?g
    { :Bill :likes ?o. } }
```

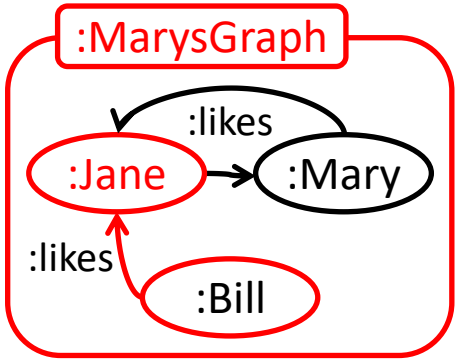


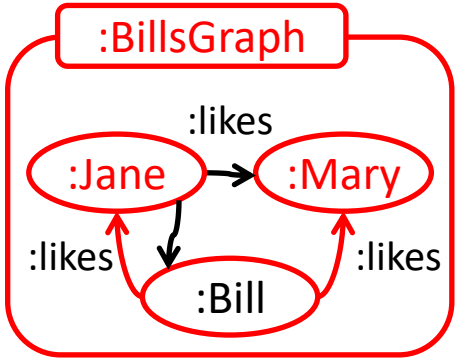
"Cut and Paste" (2/5)

```
DELETE {
  GRAPH ?g
    { :Bill :likes ?o. } }
INSERT {
  GRAPH :BGraph
    { :Bill :likes ?o. } }
WHERE {
  GRAPH ?g
    { :Bill :likes ?o. } }
```



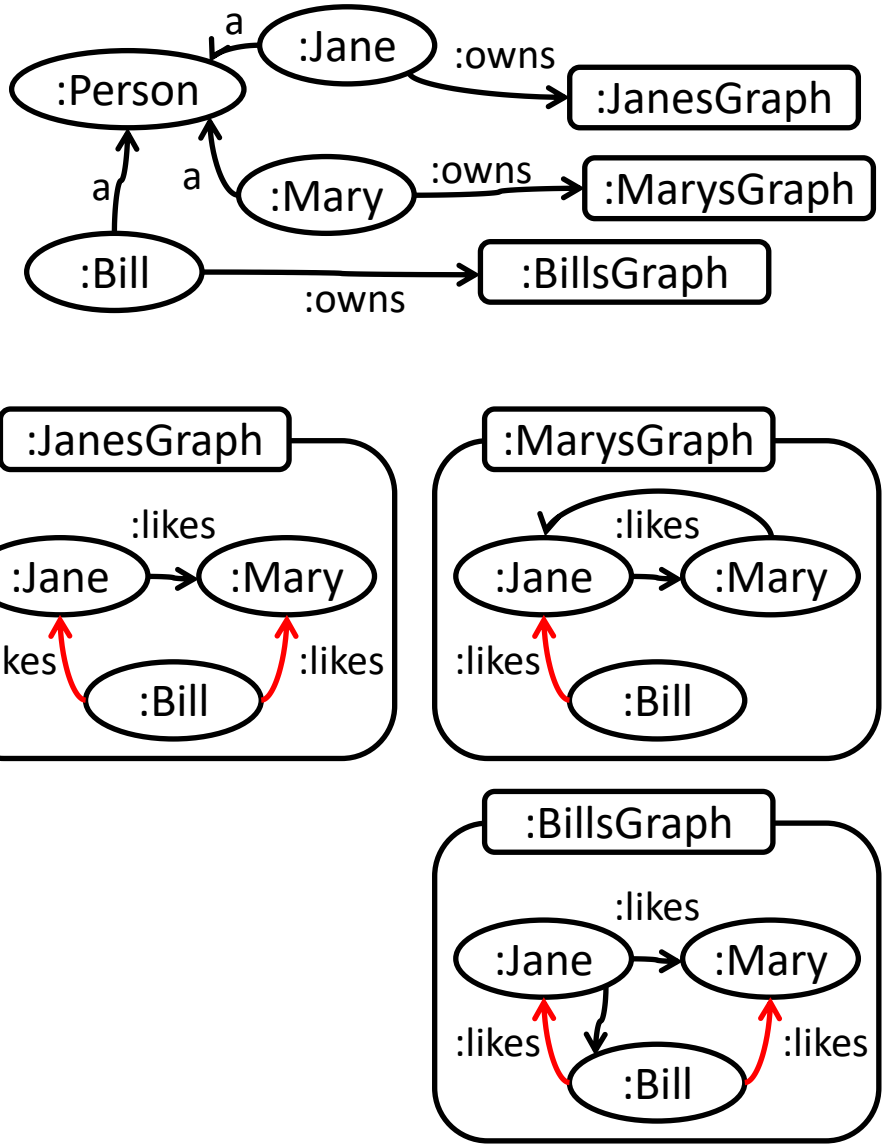






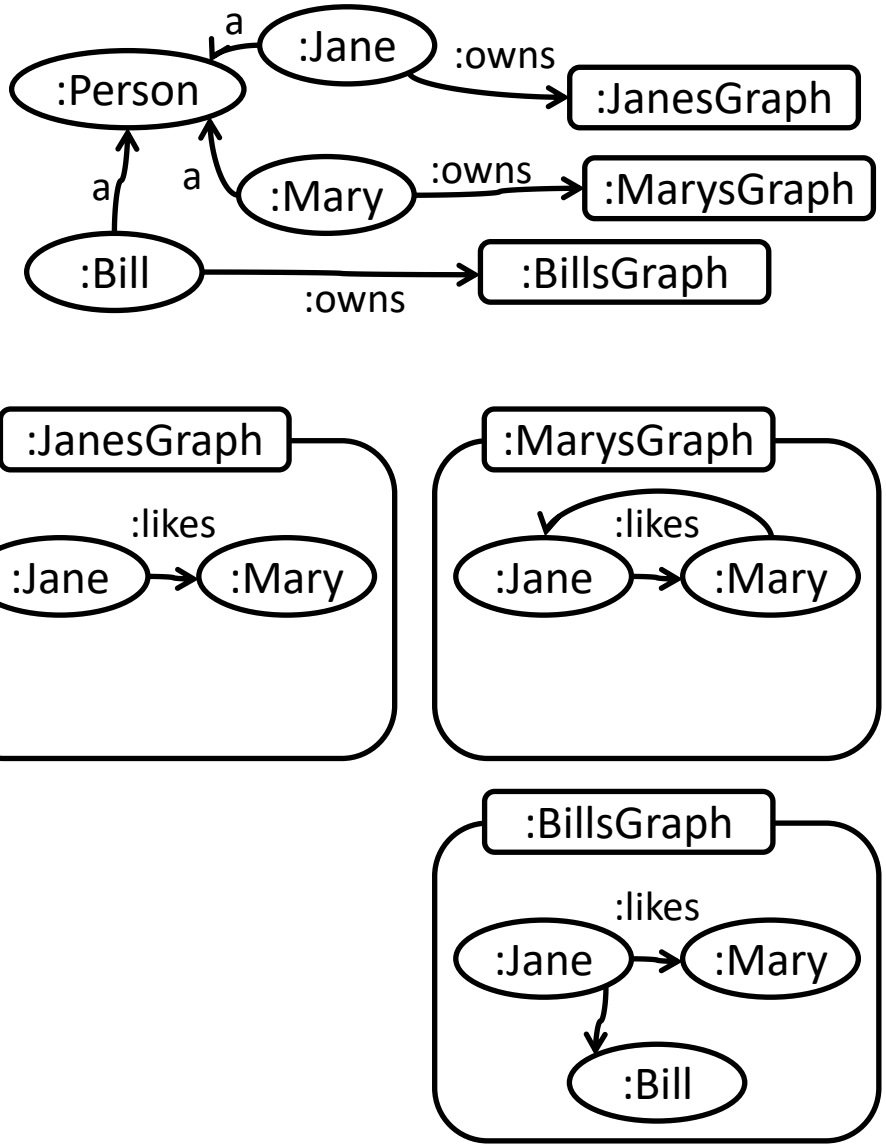
"Cut and Paste" (3/5)

```
DELETE {
  GRAPH ?g
    { :Bill :likes ?o. } }
INSERT {
  GRAPH :BGraph
    { :Bill :likes ?o. } }
WHERE {
  GRAPH ?g
    { :Bill :likes ?o. } }
```



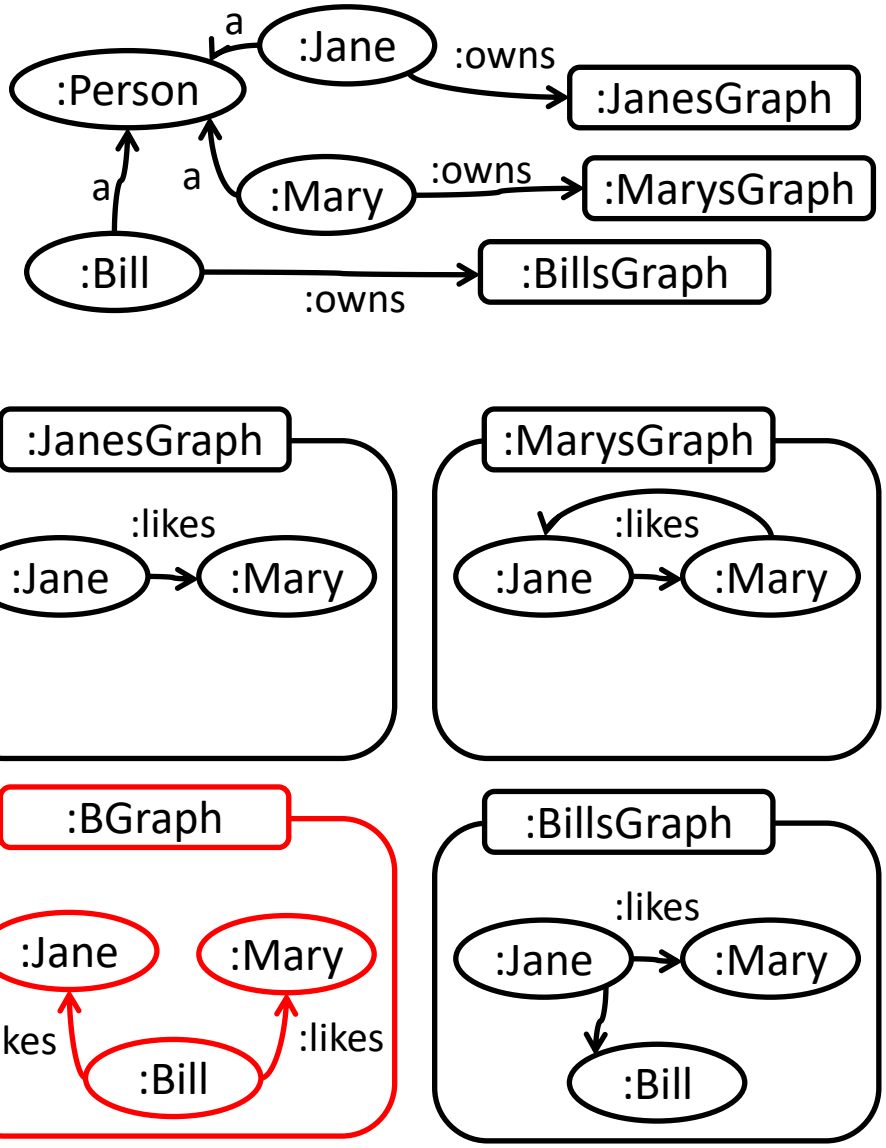
"Cut and Paste" (4/5)

```
DELETE {
  GRAPH ?g
    { :Bill :likes ?o. } }
INSERT {
  GRAPH :BGraph
    { :Bill :likes ?o. } }
WHERE {
  GRAPH ?g
    { :Bill :likes ?o. } }
```



"Cut and Paste" (5/5)

```
DELETE {
  GRAPH ?g
    { :Bill :likes ?o. } }
INSERT {
  GRAPH :BGraph
    { :Bill :likes ?o. } }
WHERE {
  GRAPH ?g
    { :Bill :likes ?o. } }
```



Summary

- Today we covered the core features of the **SPARQL 1.1 Update** W3C Recommendation.
- See https://www.w3.org/TR/2013/REC-sparql11-update-20130321/ for further details

  INSERT / DELETE: pattern-based

  INSERT DATA / DELETE DATA: explicit
- We also covered RDF Dataset queries of the **SPARQL 1.1 Query Language** W3C Recommendation.
- SPARQL Update in Jena**, Jena TDB**