SPARQL Query Language

Aggregates

Mark S. Fox PhD FAAAI FIEEE FEIC LEL
Distinguished Professor of Urban Systems Engineering
Professor of Industrial Engineering and Computer Science
Associate Director (Research), School of Cities
University of Toronto

Outline (Slides based on F. Freitas, Cin/UFPE, Brazil)

1. Basic queries
2. Constraints
3. **Aggregation**
4. Graphs
5. Query Forms
6. SPARQL in Fuseki

Aggregates: Count Items

```
@prefix ex: <http://example.org/> .

ex:lemonade1    ex:price 3 ;
                rdf:type ex:Softdrink.
ex:beer1        ex:price 3;
                rdf:type ex:Beer.
ex:wine1        ex:price 3.50 ;
                rdf:type ex:Wine.
ex:wine2        ex:price 4 .
                rdf:type ex:Wine.
ex:wine3        ex:price "n/a";
                rdf:type ex:Wine.
```

```
PREFIX ex: <http://example.org/>
SELECT (Count(?Item) AS ?C)
WHERE { ?Item ex:price ?Pr }
```

Results:

?C
5

Aggregates: Count Categories

```
@prefix ex: <http://example.org/> .

ex:lemonade1    ex:price 3 ;
                rdf:type ex:Softdrink.
ex:beer1        ex:price 3;
                rdf:type ex:Beer.
ex:wine1        ex:price 3.50 ;
                rdf:type ex:Wine.
ex:wine2        ex:price 4 .
                rdf:type ex:Wine.
ex:wine3        ex:price "n/a";
                rdf:type ex:Wine.
```

```
PREFIX ex: <http://example.org/>
SELECT (Count(DISTINCT ?T) AS ?C)
WHERE { ?Item rdf:type ?T }
```

Results:

?C
3

Aggregates: Count Items per Categories

```
@prefix ex: <http://example.org/> .

ex:lemonade1    ex:price 3 ;
                rdf:type ex:Softdrink.
ex:beer1        ex:price 3;
                rdf:type ex:Beer.
ex:wine1        ex:price 3.50 ;
                rdf:type ex:Wine.
ex:wine2        ex:price 4 .
                rdf:type ex:Wine.
ex:wine3        ex:price "n/a";
                rdf:type ex:Wine.
```

```
PREFIX ex: <http://example.org/>
SELECT ?T (Count(?Item) AS ?C)
WHERE { ?Item rdf:type ?T }
GROUP BY ?T
```

Results:

?T	?C
Softdrink	1
Beer	1
Wine	3

Aggregates: Count Items per Categories

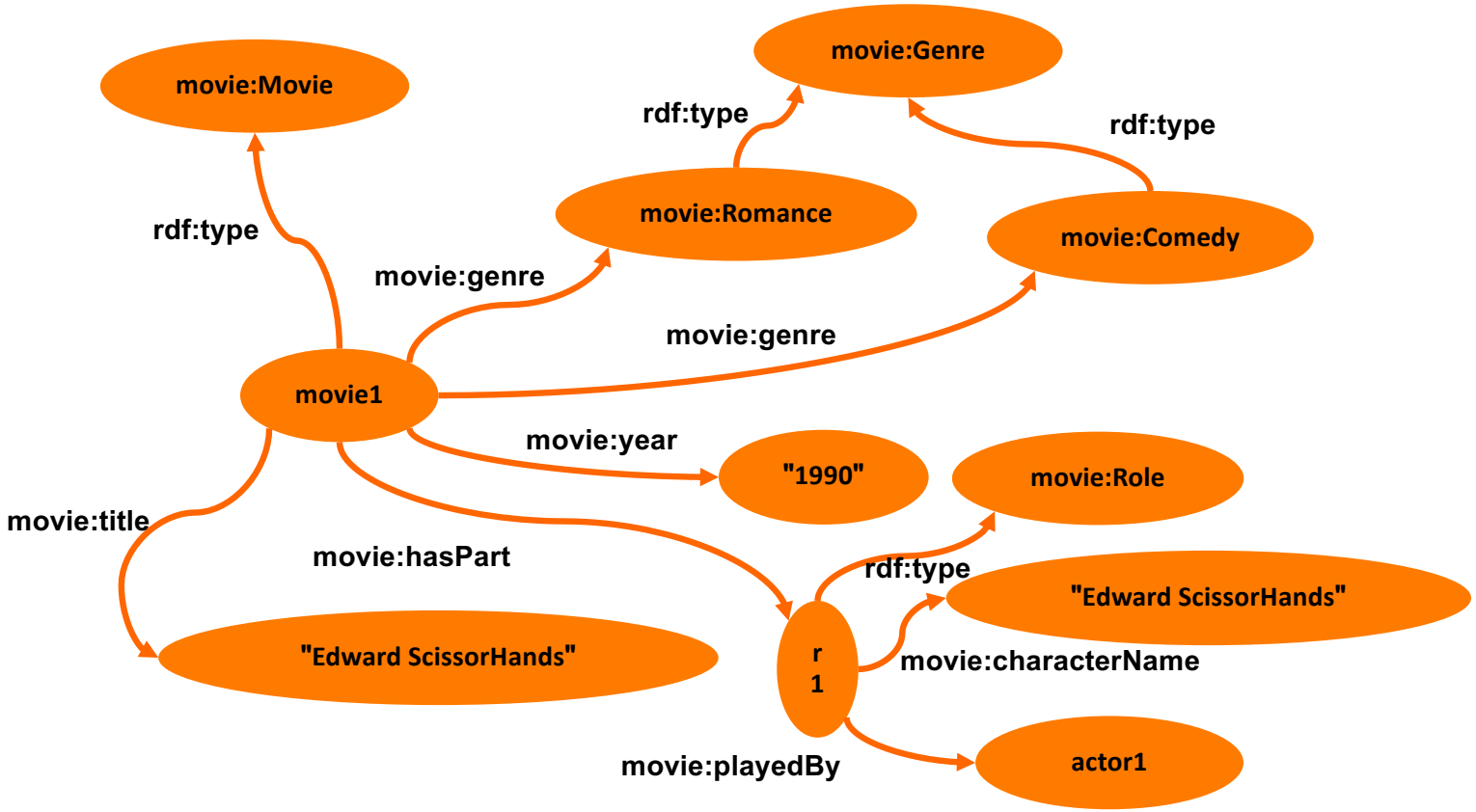
```
@prefix ex: <http://example.org/> .

ex:lemonade1    ex:price 3 ;
                rdf:type ex:Softdrink.
ex:beer1        ex:price 3;
                rdf:type ex:Beer.
ex:wine1        ex:price 3.50 ;
                rdf:type ex:Wine.
ex:wine2        ex:price 4 .
                rdf:type ex:Wine.
ex:wine3        ex:price "n/a";
                rdf:type ex:Wine.
```

```
PREFIX ex: <http://example.org/>
SELECT ?T (Count(?Item) AS ?C)
WHERE { ?Item rdf:type ?T }
GROUP BY ?T
HAVING Count(?Item) > 1
```

Results:

?T	?C
Wine	3

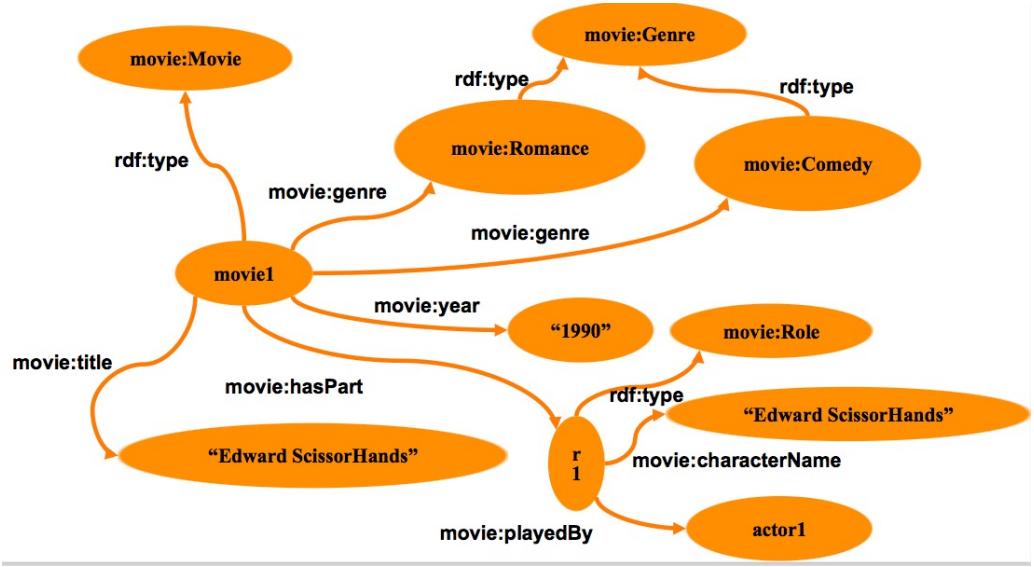
Movies Example



Example Query 1

- Select the movies that has a character called "Edward Scissorhands"

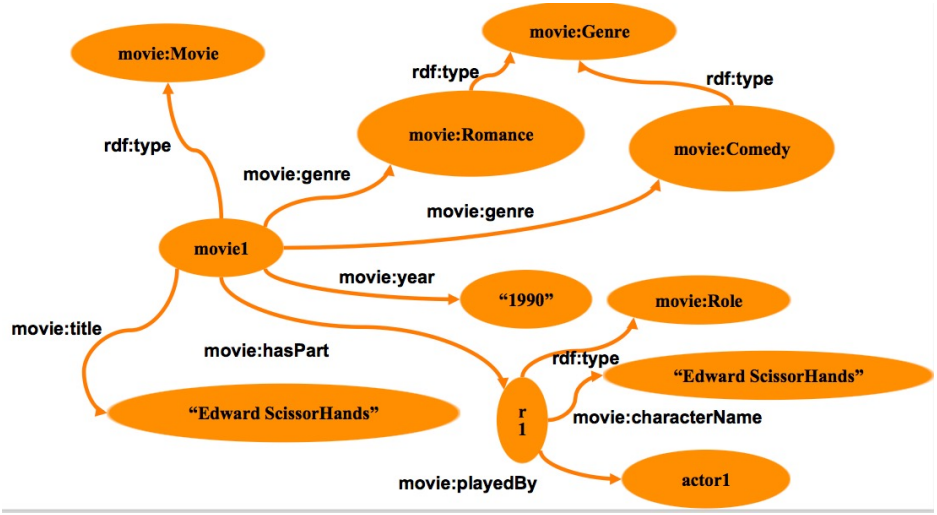
```
PREFIX movie: <http://example.org/movies/>
SELECT DISTINCT ?x ?t
WHERE { ?x movie:title ?t ; movie:hasPart ?y .?y  movie:characterName ?z .
        FILTER (?z = "Edward ScissorHands"@en)}
```



Example Query 2

- Find all movies which share at least one genre with "Gone with the Wind"

```
PREFIX movie: <http://example.org/movies/>
SELECT DISTINCT ?x2 ?t2
WHERE { ?x1 movie:title ?t1. ?x1 movie:genre ?g1. ?x2 movie:genre ?g2.
        ?x2 movie:title ?t2.
        FILTER (?t1 = "Gone with the Wind"@en && ?x1!=?x2 && ?g1==?g2)}
```



Example Query 3

- Create a graph of actors and relate them to the movies they play in (through a new ‘playsInMovie’ relation)

```
PREFIX movie: <http://example.org/movies/>
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>

CONSTRUCT { ?x movie:playInMovie ?m
          }
WHERE  {
          ?m movie:title ?t ;
             movie:hasPart ?y .
          ?y movie:playedBy ?x .
          ?x foaf:firstName ?fname.
          ?x foaf:lastName ?lname.
}
```

Other Aggregates

- SUM *… as usual*
- AVG *… as usual*
- MIN *… as usual*
- MAX *… as usual*
- SAMPLE *… "pick" one non-deterministically*
- GROUP_CONCAT *… concatenate values with a designated separator string*

…this list is extensible *… new built-ins will need to define error-behaviour, extra-parameters (like SEPARATOR in GROUP_CONCAT)*