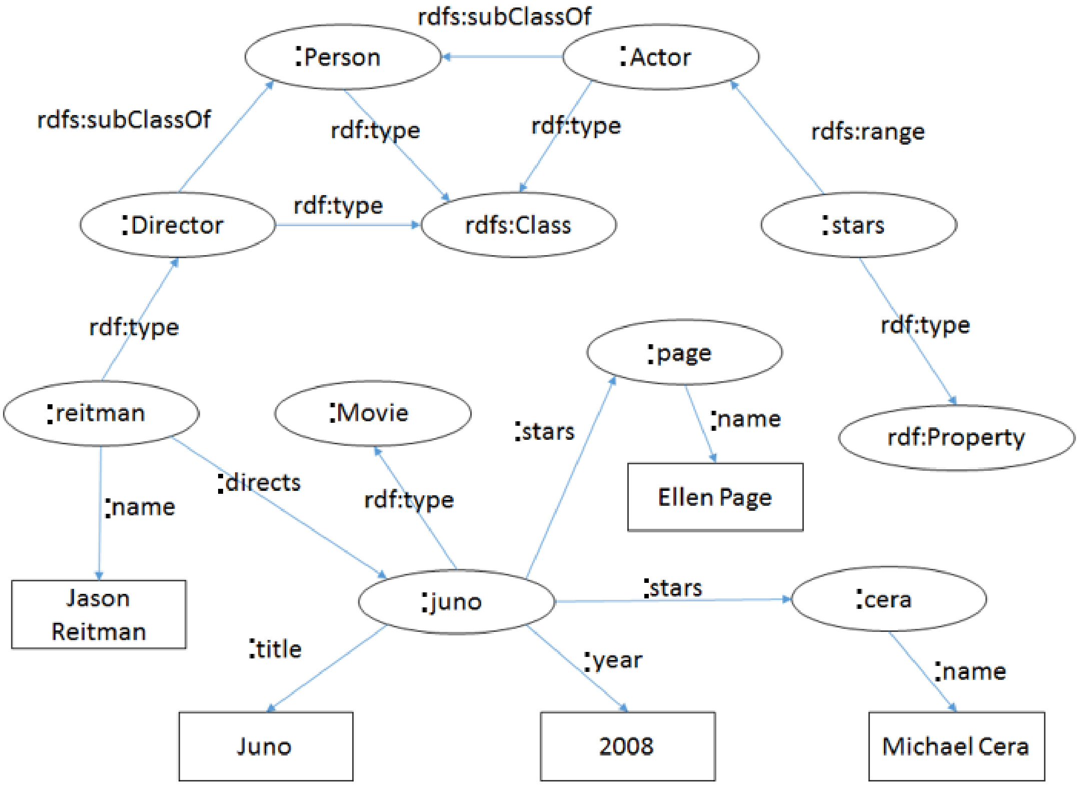
SPARQL EXERCISE

Name: ……….

Question 1.

Consider the following RDF model in Turtle notation:



*Note: Squared concepts represent literals. For the other concepts without an explicit URI assume the http://www.example.org/ namespace. Properly define **any other** prefix than rdf and rdfs you may want to use.*

INFERENCE DEACTIVATED:

a) Get the name of all actors that participated in Juno

```
PREFIX ex: <http://www.example.org/>
SELECT     ?name
WHERE
           ?X ex:stars ?Y .
           ?Y ex:name ?name .
           ?X ex:title ?title .
           FILTER regex(?title, “JUNO”)
```

Since inference is deactivated we rely on the semantics of :star. :title and :name properties to identify the precise movie and related actors. We could add the triple stating `?X a ex:movie` or `?Y a ex:actor` but the result might be then not complete. Realise that if I forget to state a move with rdf:type ex:movie (respectively, rdf:type ex:actor), this resource would not appear in the result **if precise typing is required in the pattern**. In general, without inference, we cannot assume people will be systematic type (with rdf:type) every resource, and we may lose results. You have an example in this graph. ex:page has not been typed as :actor (ex:page a ex:actor does not exist in this graph) and adding `?Y a ex:actor` to the pattern would make that :page or :cera would not appear in the result.

b) Get the name of all directors

```
PREFIX ex: <http://www.example.org/>
SELECT     ?name
WHERE
           ?X ex:directs ?Y .
           ?X ex:name ?name .
```

Like in the previous query, we cannot rely on typing. Therefore, we can only rely on the semantics of :directs. We could add `?X a ex:director` or `?Y a ex:movie`, but then, the same discussion as above applies, and we would depend on people adding the triples properly typing directors (which we cannot guarantee).

c) Get the name of all persons

```
PREFIX ex: <http://www.example.org/>
SELECT     ?name
WHERE {
           ?X ex:directs ?Y .
           ?X ex:name ?name . }
UNION {
           ?X ex:stars ?Y .
           ?Y ex:name ?name . }
```

Since we cannot rely on typing we need to look for specific resources participating in triples whose semantics (that of the predicate) states there is a person on the domain or range. If a person can be a director and an actor, we could add DISTINCT to the SELECT clause.

d) Get the title of all movies

```
PREFIX ex: <http://www.example.org/>
SELECT DISTINCT ?title
WHERE {
           ?X ex:directs ?Y .
           ?Y ex:title ?title . }
UNION {
           ?X ex:stars ?Y .
           ?X ex:title ?title . }
```

Again, we cannot rely on typing and we must identify those triples that identify a movie either in their range or domain. Note we do not consider year because it is not strongly related to the concept movie and we just focus on those with a strong implication.

Overall, without typing, we are left to interpret the semantics of the properties and query the explicit knowledge asserted.

INFERENCE ACTIVATED

a) Get the name of all actors that participated in Juno

```
PREFIX ex: <http://www.example.org/>
SELECT     ?name
WHERE
           ?X ex:title ?title .
           ?X ex:stars ?Y .
           ?Y a ex:actor .
           ?Y ex:name ?name .
           FILTER regex(?title, “JUNO”)
```

Now we can rely on typing for :stars, since there is a triple defining its :range. We still cannot rely on typing for movies because there is no triple typing the domain or range for directs or stars in the movie-side. Therefore, typing is only reliable for actor.

If we add the range and domain for directs and stars, respectively, for movies, then, we could safely add `?X a :movie .`

b) Get the name of all directors

Since there is no typing inference for directors, the query would remain the same. However, if we add to the schema information the following triple: `ex:directs rdfs:domain ex:director`, the query would become much simpler:

```
PREFIX ex: <http://www.example.org/>
SELECT     ?name
WHERE
           ?X a :director .
           ?X ex:name ?name .
```

c) Get the name of all persons

Even if actor and director are properly identified as subsets of person, we still rely on identifying actors and directors depending on their participation in triples whose property is directs or stars. Therefore, the query would remain the same. However, if we would have defined the range and domain for directs and stars, plus the current triples defining actor and director as rdfs:subClassOf, we could have then largely simplified the query to:

```
PREFIX ex: <http://www.example.org/>
SELECT     ?name
WHERE{
           ?X a ex:person .
           ?X ex:name ?name .
```

d) Get the title of all movies

Same as before. Since the typing inference for movie is not guaranteed with the current schema information, we would still rely on the same query as before. However, if we properly define the domain and range of stars and directs so movies are automatically asserted as such, we could largely simply the query to:

```
PREFIX ex: <http://www.example.org/>
SELECT     ?title
WHERE
           ?X a :movie .
           ?Y ex:title ?title .
```

Summing up, activating inference is not enough. We need to add the right schema triples carefully to generate the right inference. In general, the more you type, the easier to query. However, sometimes is not possible to infer the domain / range of a property in a univocal way and then, you may still need to rely on the semantics provided by the properties at the instance level.