W2. Check List — Junyi

- ▼ What are Popular Data Models

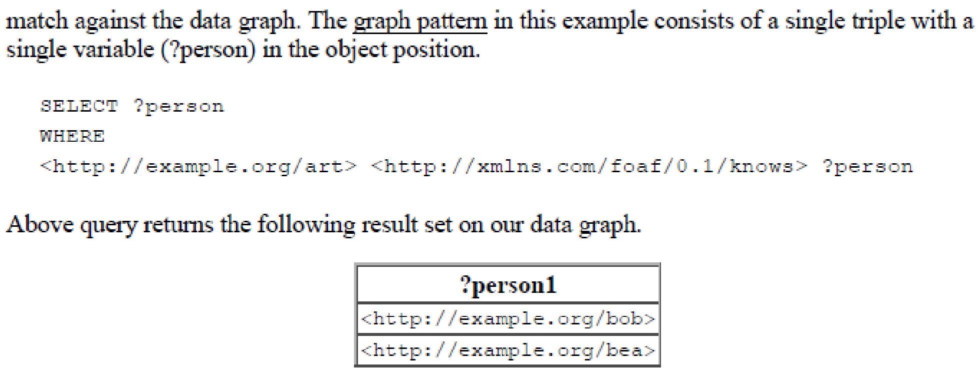
  RDF & Property Graph

- ▼ RDF Data model
  - ▼ Components?
    - ▼ a subject, a predicate, and
      an object.
  - ▼ types of node?
    - ▼

      An IRI is an Internationalized
      Resource Identifier used to uniquely identify resources on the web. A literal is a value of a certain
      data type, for example, string, integer, etc. A blank node is a node that does not have an
      identifier, and is similar to an anonmyous or an existential variable
  - ▼ RDF vocabulary (colletion of IRIs) → RDF graphs → RDF datasets
    - ▼ An RDF dataset is a collection of RDF graphs and comprises exactly one default graph that can be
      empty and does not need to have a name, and one or more named graphs. Each named graph
      consists of an IRI or a blank node that represents its name and the RDF graph.
  - ▼ SPARQL queries
    - ▼ Capacity
      - ▼ querying required and optional graph patterns along with their conjunctions and disjunctions.
    - ▼ Results Form

be sets or RDF graphs

        - ▼ The query consists of two parts
            - ▼ SELECT clause identifies the variables to appear in the query results
            - ▼ WHERE clause provides the graph pattern to match against the data graph.



match against the data graph. The graph pattern in this example consists of a single triple with a single variable (?person) in the object position.

```
SELECT ?person
WHERE
<http://example.org/art> <http://xmlns.com/foaf/0.1/knows> ?person
```

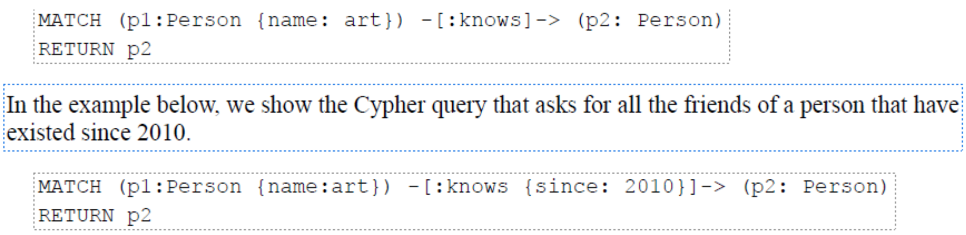
Above query returns the following result set on our data graph.

?person1
<http://example.org/bob>
<http://example.org/bea>

        - ▼ queries have various forms
            - ▼ SELECT form that we have considered until now returns the variable bindings
            - ▼ CONSTRUCT form can be used to create results that define an RDF graph

- ▼ PG model
    - ▼ components
        - ▼ nodes, relationships, properties
            - ▼ node components
                - ▼ a label, a set of properties
                - ▼ keys: string; values: arbitrary data
    - ▼ define a peoperty graph
        - ▼ node, edges, properties
    - ▼ Cypher queries
        - ▼ capacity
            - ▼ query, create, update, delete data from a graph databases

    - ▼ query clauses
        - ▼ MATCH: specifies a graph pattern that should match against the data graph
        - ▼ RETURN: what should be returned
    - ▼ Notation
        - ▼ specified in an ASCII notation for graphs: each node is written in parentheses, and each edge is written as an arrow.
        - ▼



```
MATCH (p1:Person {name: art}) -[:knows]-> (p2: Person)
RETURN p2
```

In the example below, we show the Cypher query that asks for all the friends of a person that have existed since 2010.

```
MATCH (p1:Person {name:art}) -[:knows {since: 2010}]-> (p2: Person)
RETURN p2
```

- ▼ Comparison of Data Models
    - ▼ Comparison of RDF and Property Graph Data Models
        - ▼ RDF requires IRIs and blank nodes
        - ▼ existence of properties
        - ▼ → To support the edge properties, the RDF model supports an extension known as reification.
        - ▼ How does reifiction work?
            - ▼ type rdf:statement, property rdf: (subject, predate, object)
    - ▼ Convertion beween RDF and GD
        - ▼ With the above **reification** vocabulary, it becomes possible to mechanically translate the data in the property graph model to RDF.
        - ▼ A possible refinement is that we create new property nodes only for those nodes that are either IRIs or blanks nodes. For any triple in RDF in which the target is a literal, we make it a property of the node in the property graph data.
        - ▼ converting the syntactic form of data and the queries.
    - ▼ Comparison of Graph Models and Relational Data Model

- ▼ little reliance on a predefined schema, and the optimization of operations that involve graph traversals
- ▼ → Alternative: represent the relational data in a schema free manner

▼ a graph data model offers significant advantages for application that have rich relationships between objects, and require extensive traversal of those relationships.
BHOWEVER, limitation of GDM: unproperiate if

- ▼ contains primarily numeric data, e.g., timeseries → huge number of nodes
- ▼ many relationships that cannot be naturally represented using binary relations
- ▼ mathematical equations and chemical reactions