DSE 203 Final Project

Schema Mapping Specification and Implementation

Neo4J Schema

The Neo4j datastore seems to have two distinct networks that do not share any nodes.

Schema A

(node) **:Organization**
- **.Name** (string, could be / delimited)

(node) **:Actor**
- **.Type** ["Individual" | "Group"]
- **.AliasList** (list of strings)
- **.Name** (string)

(node) **:Country**
- **.Name** (string)

(relationship) **:Affiliation**
- **.beginDate** (YYYY-MM-DD or "beginning of time")
- **.endDate** (YYYY-MM-DD or "end of time")

(relationship) **:From**

```
(:Actor)-[:Affiliation]->(:Organization)

(:Actor)-[:From]->(:Country)
```

Schema B

(node) **:AgentName**
.Name (string)

(node) **:Aliases**
.AliasList (list of strings)

(node) **:AgentType**
.Name [“Individual” | “Group”]

(node) **:Sector**
.Name (string)

(relationship) **:AgentAlias**

(relationship) **:AgentType**

(relationship) **:is-a**

(relationship) **:Sector**

```
(:AgentName)-[:AgentAlias]->(:Aliases)

(:AgentName)-[:AgentType]->(:AgentType)

(:AgentName)-[:Sector]->(:Sector)

(:Sector)-[:is-a]->(:Sector)
```

Information Provided by Neo4J Schema

1. Group/individual country
2. Group/individual organization
3. Group/individual dates of association with organization
4. Group/individual aliases
5. Agent (perpetrator/victim) name
6. Agent aliases
7. Agent type (individual/group)
8. Agent category

Global Schema

Changes to the global schema may necessitate changes to the mappings.

```
Events(IEID, SID, Date, Country, State, LocationType,
     Location,  Summary, EventType )

Participants(IEID, SID, PID, PType, PName, PRole, )

ParticipantDetail(PID, Category, Organization, Country, Name,
     Description)

EventDetails(IEID, SID, Description)

RelatedEvents(IEID, IEID2)
```

Graph-to-Relational

In order to write datalog against the Neo4j schema, I think we need to convert it from a graph schema to a relational schema.

Schema A

```
Actor(id, ptype, pname)

AliasList(id, alias)

From(id, country)

Affiliation(id, org, start, end)
```

Schema B

```
AgentName(id, pname)

AgentType(id, ptype)

Aliases(id, alias)

AgentSector(id, sector_id)

SectorName(sector_id, sector_name)

SectorIs-A(sector_id1, sector_id2)
```

Mapping

Since the mappings below are a bit confusing, here's a quick legend for reading them:

```
GlobalSchema(_, return_variable, "Literal value" )

      NEO4J QUERY THAT SATISFIES ABOVE GLOBAL SCHEMA

      Datalog definition of the above schema mapping (source-to-global)
```

(In the `GlobalSchema`, the _ is an ignored field, `return_variable` is an output we expect, and `"literal value"` is an input that is required.)

```
Participants(_, sid, _, ptype, "Some Name", _)

      MATCH (a:Actor)
      WHERE a.AliasList CONTAINS "Some Name"
      RETURN a.Type AS ptype, ID(a) AS sid

      Participants(_, "NEO4J:"+sid, _, ptype, "Some Name", _) :-
            AliasList( sid, "Some Name" ),
            Actor( sid, ptype, name)

      MATCH (a:AgentName)-[:AgentAlias]->(n:Aliases),
            (a:AgentName)-[:AgentType]->(t:AgentType)
      WHERE n.AliasList CONTAINS "Some Name"
      RETURN t.Name AS ptype, ID(a) AS sid

      Participants(_,"NEO4J:"+sid, _, ptype, "Some Name", _) :-
            Aliases( sid, "Some Name" ),
            AgentType( sid, ptype )

Participants(_, sid, _, "Group", pname, _)

      MATCH (a:Actor)
      WHERE a.Type = "Group"
      return a.Name AS pname, ID(a) AS sid

      Participants(_,"NEO4J:"+ sid, _, "Group", pname, _) :-
            Actor( sid, "Group", pname )

      MATCH (a:AgentName)-[:AgentType]->(t:AgentType)
      WHERE t.Name = "Group"
      RETURN a.Name AS pname, ID(a) AS sid

      Participants(_,"NEO4J:"+ sid, _, "Group", pname, _) :-
            AgentType( sid, "Group" ),
            AgentName( sid, pname )

Participants(_, 12345, _, ptype, pname, _)

      MATCH (a:Actor)
```

```
WHERE ID(a) = 12345
RETURN a.Type AS ptype, a.Name AS pname
```

Participants(_,12345, _, ptype, pname, _) :-
Actor(12345, _, pname),
Type(12345, ptype)

```
MATCH (a:AgentName)-[:AgentType]->(t:AgentType)
WHERE ID(a) = 12345
RETURN t.Name AS ptype, a.Name as pname
```

Participants(_, 12345, _, ptype, pname, _) :-
AgentType(12345, ptype),
AgentName(12345, pname)

```
ParticipantDetails(_, _, org, country, “Some Name”, _)

MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[:Affiliation]->(o:Organization)
WHERE a.AliasList CONTAINS “Some Name”
RETURN o.Name AS org, c.Name AS country
```

ParticipantDetails(_, _, org, country, “Some Name”, _) :-
From(sid, country),
Affiliation(sid, org, _, _),
Actor(sid, _, “Some Name”)

```
ParticipantDetails(_, _, org, “Afghanistan”, name, _)

MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[:Affiliation]->(o.Organization)
WHERE c.Name = “Afghanistan”
RETURN o.Name AS org, a.Name AS name
```

ParticipantDetails(_, _, org, “Afghanistan”, name, _) :-
From(sid, “Afghanistan”),
Affiliation(sid, org, _, _),
Actor(sid, _, name)

```
ParticipantDetails(_, _, “Taliban”, country, name, _)

MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[:Affiliation]->(o.Organization)
WHERE o.Name = “Taliban”
RETURN c.Name AS country, a.Name AS name
```

ParticipantDetails(_, _, “Taliban”, country, name, _) :-
From(sid, country),
Affiliation(sid, “Taliban”, _, _),
Actor(sid, _, name)

```
ParticipantDetails(_, _, "Taliban", "Afghanistan", name, _)

      MATCH (a:Actor)-[:From]->(c:Country),
            (a:Actor)-[:Affiliation]->(o.Organization)
      WHERE c.Name = "Afghanistan" AND
            o.Name = "Taliban"
      RETURN a.Name AS name
```

ParticipantDetails(_, _, "Taliban", "Afghanistan", name, _) :-
From(sid, "Afghanistan"),
Affilliation(sid, "Taliban", _, _),
Actor(sid, _, name)

```
ParticipantDetails(_, _, "Taliban", country, "Some Name", _)

      MATCH (a:Actor)-[:From]->(c:Country),
            (a:Actor)-[:Affiliation]->(o.Organization)
      WHERE o.Name = "Taliban" AND
            a.AliasList CONTAINS "Some Name"
      RETURN c.Name AS country
```

ParticipantDetails(_, _, "Taliban", country, "Some Name", _) :-
From(sid, country),
Affiliation(sid, "Taliban", _, _)
Actor(sid, _, "Some Name")

```
ParticipantDetails(_, _, org, "Afghanistan", "Some Name", _)

      MATCH (a:Actor)-[:From]->(c:Country),
            (a:Actor)-[:Affiliation]->(o.Organization)
      WHERE c.Name = "Afghanistan" AND
            a.AliasList CONTAINS "Some Name"
      RETURN o.Name AS org
```

ParticipantDetails(_, _, org, "Afghanistan", "Some Name", _) :-
From(sid, "Afghanistan"),
Affiliation(sid, org, _, _),
Actor(sid, "Some Name")

```
ParticipantDetails(_, category, _, _, "Some Name", _)

      MATCH (a:AgentName)-[:AgentAlias]->(n:Aliases),
            (a:AgentName)-[:Sector]->(s1:Sector),
            (s1:Sector)-[:is-a]->(s2:Sector)
      WHERE n.AliasList CONTAINS "Some Name"
      RETURN s1.Name + s2.Name AS desc
```

ParticipantDetails(_, _, sector_name1+sector_name2, _, "Some Name", _) :-
Aliases(sid, "Some Name"),
AgentSector(sid, sector_id1),
SectorName(sector_id1, sector_name1),

SectorIs-A(sector_id1, sector_id2),
SectorName(sector_id2, sector_name2)

```
ParticipantDetails(_, _, org, country, name, _),
Events( …, “2015-01-02”, … )
```

```
MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[aff:Affiliation]->(o:Organization)
WHERE “2015-01-02”>= aff.beginDate AND
      “2015-01-02”<= aff.endDate
RETURN o.Name AS org, c.Name AS country, a.Name as name
```

ParticipantDetails(_, _, org, country, name, _), Events(…,“2015-01-02”, …) :-
From(sid, country),
Affiliation(sid, org, start, end),
Actor(sid, _, name),
“2015-01-02” >= start,
“2015-01-02”<= end

```
ParticipantDetails(_, _, org, country, “Some Name”, _),
Events( …, “2015-01-02”, … )
```

```
MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[aff:Affiliation]->(o:Organization)
WHERE a.AliasList CONTAINS “Some Name” AND
      “2015-01-02”>= aff.beginDate AND
      “2015-01-02”<= aff.endDate
RETURN o.Name AS org, c.Name AS country
```

ParticipantDetails(_, _, org, country, “Some Name”, _), Events(…,“2015-01-02”, …) :-
From(sid, country),
Affiliation(sid, org, start, end),
AliasList(sid, “Some Name”),
“2015-01-02” >= start,
“2015-01-02”<= end

```
ParticipantDetails(_, _, org, “Afghanistan”, name, _)
Events( …, “2015-01-02”, … )
```

```
MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[aff:Affiliation]->(o.Organization)
WHERE c.Name = “Afghanistan” AND
      “2015-01-02”>= aff.beginDate AND
      “2015-01-02”<= aff.endDate
RETURN o.Name AS org, a.Name AS name
```

ParticipantDetails(_, _, org, “Afghanistan”, name, _), Events(…,“2015-01-02”, …) :-
From(sid, “Afghanistan”),
Affiliation(sid, org, start, end),
AliasList(sid, name),
“2015-01-02”>= start,

"2015-01-02"<= end

```
ParticipantDetails(_, _, "Taliban", country, name, _)
Events( …, "2015-01-02", … )
```

```
MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[aff:Affiliation]->(o.Organization)
WHERE o.Name = "Taliban" AND
      "2015-01-02">= aff.beginDate AND
      "2015-01-02"<= aff.endDate
RETURN c.Name AS country, a.Name AS name
```

ParticipantDetails(_, _, "Taliban", country, name, _), Events(…,"2015-01-02", …) :-
From(sid, country),
Affiliation(sid, "Taliban", start, end),
AliasList(sid, name),
"2015-01-02">= start,
"2015-01-02"<= end

```
ParticipantDetails(_, _, "Taliban", "Afghanistan", name, _)
Events( …, "2015-01-02", … )
```

```
MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[aff:Affiliation]->(o.Organization)
WHERE c.Name = "Afghanistan" AND
      o.Name = "Taliban" AND
      "2015-01-02">= aff.beginDate AND
      "2015-01-02"<= aff.endDate
RETURN a.Name AS name
```

ParticipantDetails(_, _, "Taliban", "Afghanistan", name, _), Events(…,"2015-01-02", …) :-
From(sid, "Afghanistan"),
Affiliation(sid, "Taliban", start, end),
AliasList(sid, name),
"2015-01-02">= start,
"2015-01-02"<= end

```
ParticipantDetails(_, _, "Taliban", country, "Some Name", _)
Events( …, "2015-01-02", … )
```

```
MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[aff:Affiliation]->(o.Organization)
WHERE o.Name = "Taliban" AND
      a.AliasList CONTAINS "Some Name" AND
      "2015-01-02">= aff.beginDate AND
      "2015-01-02"<= aff.endDate
RETURN c.Name AS country
```

ParticipantDetails(_, _, "Taliban", country, "Some Name", _), Events(…,"2015-01-02", …) :-
From(sid, country),
Affiliation(sid, "Taliban", start, end),
AliasList(sid, "Some Name"),
"2015-01-02">= start,
"2015-01-02"<= end

```
ParticipantDetails(_, _, org, "Afghanistan", "Some Name", _)
Events( …, "2015-01-02", … )
```

```
MATCH (a:Actor)-[:From]->(c:Country),
      (a:Actor)-[aff:Affiliation]->(o.Organization)
WHERE c.Name = "Afghanistan" AND
      a.AliasList CONTAINS "Some Name" AND
      "2015-01-02">= aff.beginDate AND
      "2015-01-02"<= aff.endDate
RETURN o.Name AS org
```

ParticipantDetails(_, _, org, "Afghanistan", "Some Name", _), Events(…,"2015-01-02", …) :-
From(sid, "Afghanistan"),
Affiliation(sid, org, start, end),
AliasList(sid, "Some Name"),
"2015-01-02">= start,
"2015-01-02"<= end