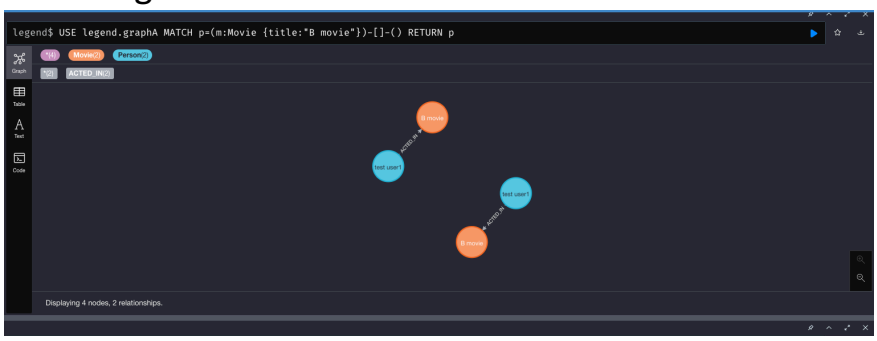
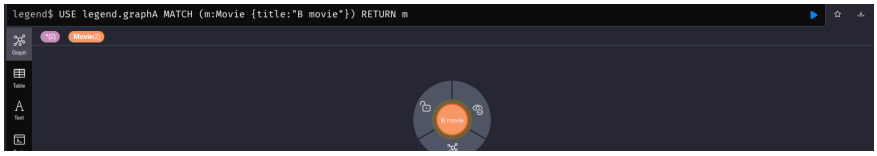
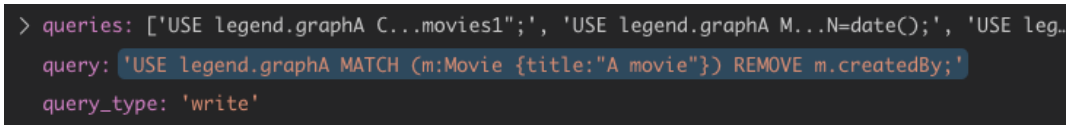
Test cases

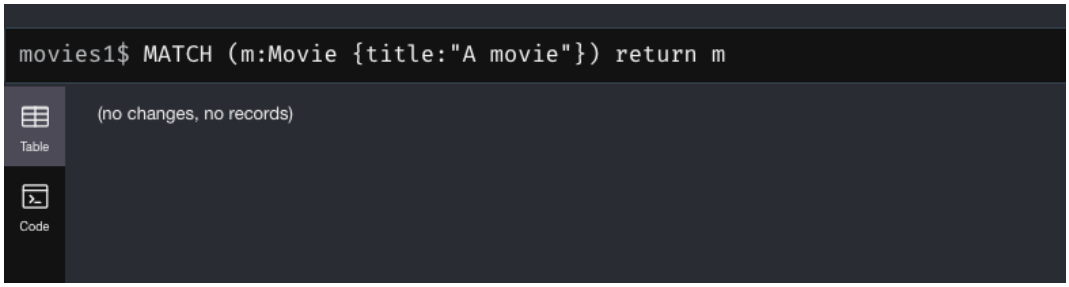
Monday, August 9, 2021 10:44 AM

1. Loading the data to a single data base
    a. using load csv using file.
        i. Question comes if we have two routing fabric servers which import folder we have to use.
            1) Need to load data to both.
            2) If the call is going to the other fabric server where files are not there, will get error "Couldn't load the external resource at: file:/home/ec2-user/neo4j-enterprise-4.3.2/import/orders.csv"
        ii. Need to use cypher subquery as csv files are on fabric and query need to be executed on different server.
    b. using https
        i. It is straight forward.
    c. using batch:
        i. Batch can be done in two ways using
2. Transaction with multiple writes to the database:
    a. Scenario 1: USE statement in each query
        i. In the transaction when each query has mentioned with USE database, query run successfully and created all node.
        ii. Browser behavior:
            1) When using with the fabric browser:
                a) Created Relations are not automatically open when we double click on the node. It made me to think relation not existing.
                b) When we go to the primary data base and see, relations are existing.
                c) On primary database if we run the return path, it is returning the relations too.
                d) 



b. Scenario 2 : USE statement once
   i. If USE db is not mentioned, it will try to write to the local database.
   ii. Error : 2021-08-09 15:14:50,178 [ERROR] {code: Neo.ClientError.Statement.AccessMode} {message: Writing to more than one database per transaction is not allowed. Attempted write to Local.
   iii. As the error is at transaction level, even the first query did not created the node. Might be it got rollbacked.

3. Transaction boundary testing:
   a. Running 3 write queries and pause the application. Observe any commit happened or not?
   b. Result: ran as expected. As commit was not happened at that instant nodes were not showing up.
      i. 

      ii. 

      iii. After running all steps: Node appered. So transaction boundary is respected.

      1) 

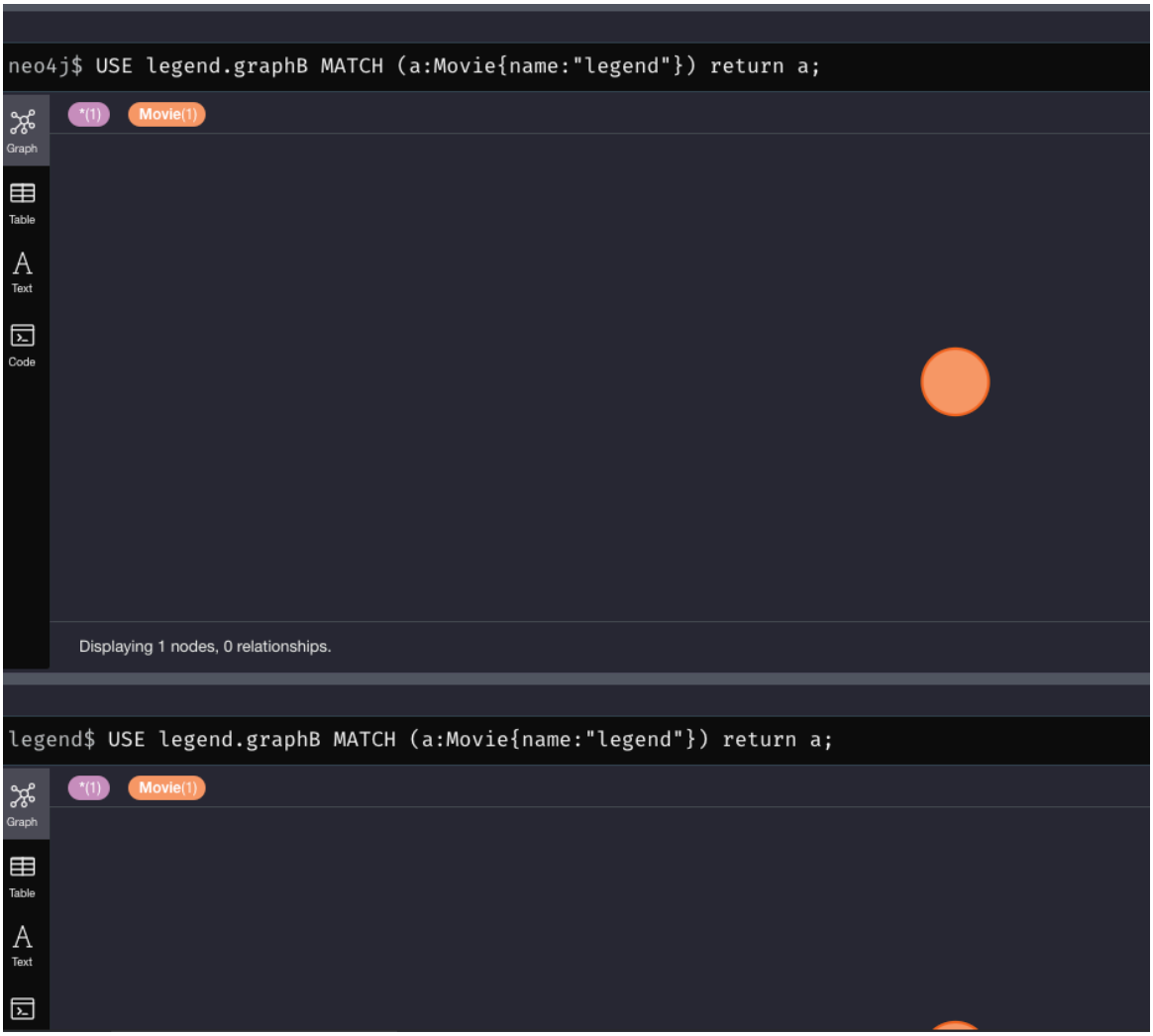
4. Running queries without mentioning the database parameter in the session.
   a. {message: Multiple graphs in the same query not allowed here. This

feature is only available in a Fabric database.
Attempted to access graph legend.graphB
"USE legend.graphB MATCH (n:Movie) RETURN n.title,n.db,"movies2" as db,"qry#123r" as type LIMIT 5"
^}

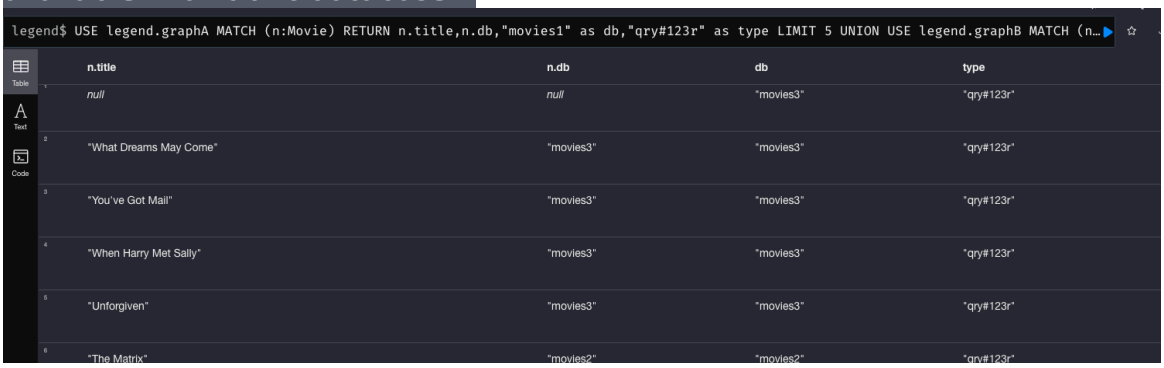
b. Will discuss query by query here.

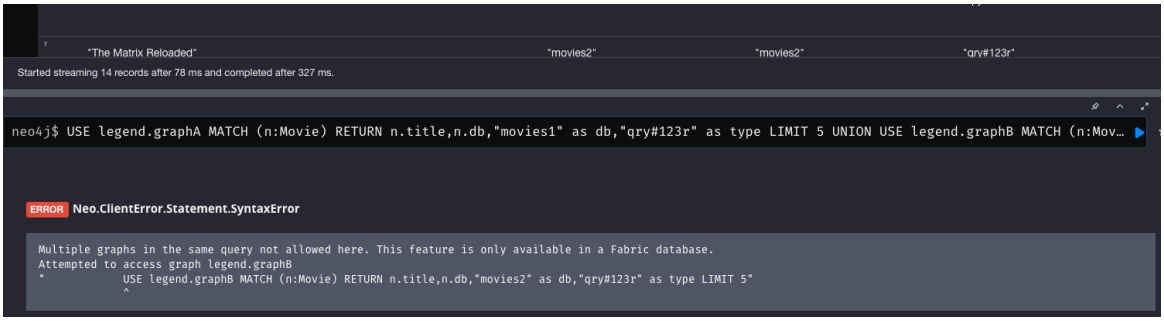
c. On fabric server browser or using ingest: Single data base queries will run by Default "neo4j" or "legend".

d.



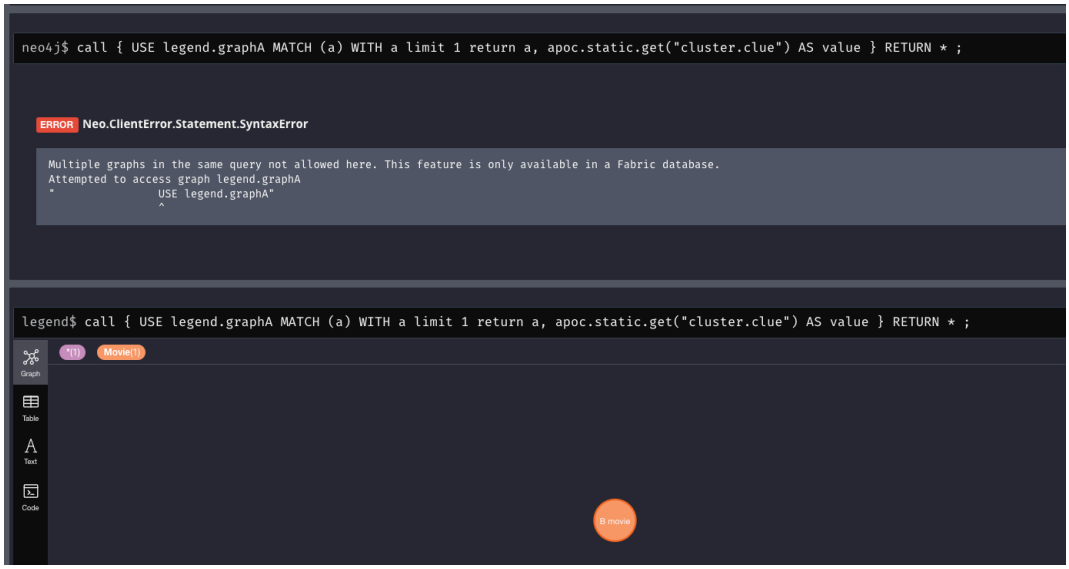
e. On fabric server browser or using ingest: Multi data base queries will run by only fabric databse "legend". If run against "neo4j" will get the error "Multiple graphs in the same query not allowed here. This feature is only available in a Fabric database."





f. About queries using CALL: Queries using CALL should only run against database "legend" which is defined fabric database.

i.



5. Plugins:
   a. If we are using the fabric.routing.servers concept, then we need to install all related plugins in each of the fabric servers.
6. Sharding Implementation challenges:
   a. Sharding criteria could be stored as graph or a function/procedure on the Fabric server. If we are using fabric.routing.servers, these need to be in sync on all servers.
      i. Example queries:
         1) Mapping function defined on both the Fabric routing servers:

```
CALL apoc.custom.asFunction(
 'get_shard_db',
 'WITH $movie_name AS movie_name
WITH left(movie_name,1) as title_starting
RETURN CASE WHEN toLower(title_starting)<="n"
THEN 0
 WHEN "o"<=toLower(title_starting)<="t" THEN 1
 ELSE 3 END AS db_name;
',
 'string',
```

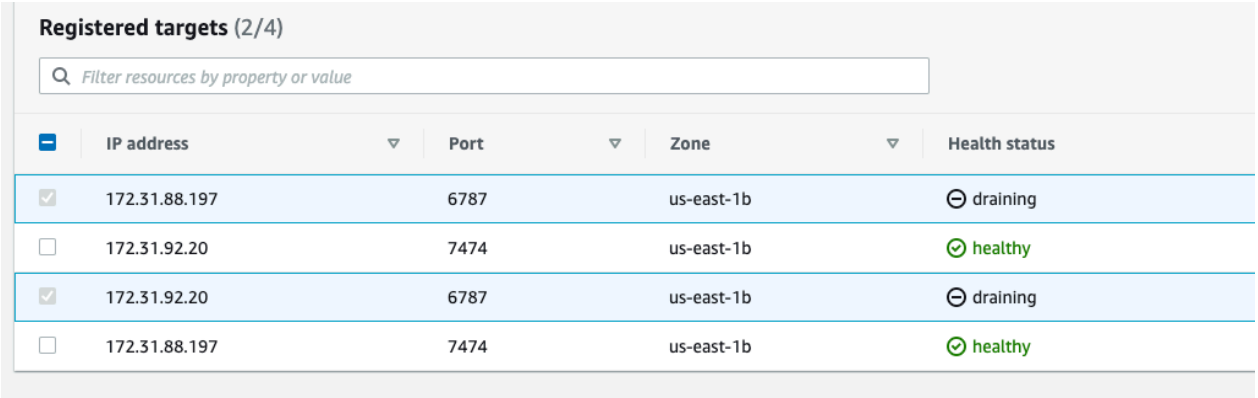
```
        [['movie_name','string']]
      );
```

2) The shading is on the first alphabet of the title. So included that logic to identify the which db it has to go.
3) Retrieval query from corresponding db.

```
WITH "The Da Vinci Code" as title
WITH title,custom.get_shard_db(title) as db_code
CALL {
WITH title,db_code
USE legend.graph(db_code)
MATCH (m:Movie) WHERE m.title = title
RETURN m
}
RETURN *
```

b. Routing context: Each routing context has to be mentioned as a separate database.
   i. Ref: "cluster routing context" in https://neo4j.com/docs/operations-manual/current/fabric/configuration/

Issues:

1. Load balancer:
   a. Load balancer has two listening address on port 7474 and 7687. Target groups are defined as
2. 

   a. Public addresses not accepted in the target groups. When private ips are used it was howing unhealth. so unregestered.