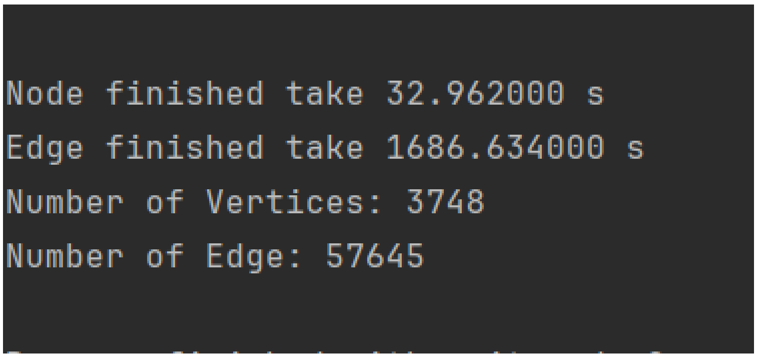
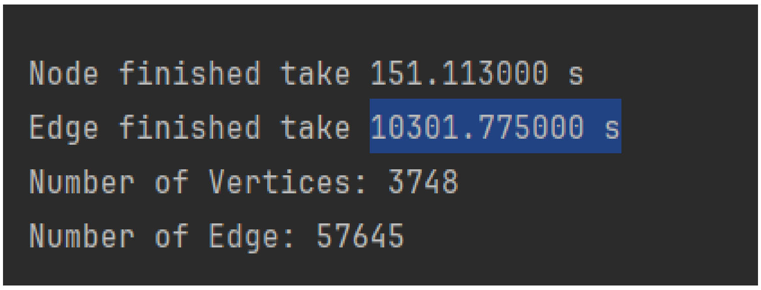
1. Inmemoery database
    a. In this part I was able to get a basic understanding of Janusgraph.I was able to create the scheme based on the provided json file, and loaded all the edges to the in-memory database. The overall time took to load is follow:



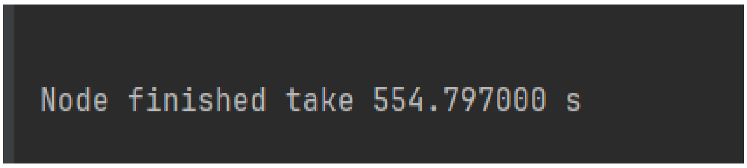
```
Node finished take 32.962000 s
Edge finished take 1686.634000 s
Number of Vertices: 3748
Number of Edge: 57645
```

2. BerkeleyDB:
    a. In this part, I was able to use the provided dependency from maven repo to intriguing a backend storage for Janus graph. The result look follow:

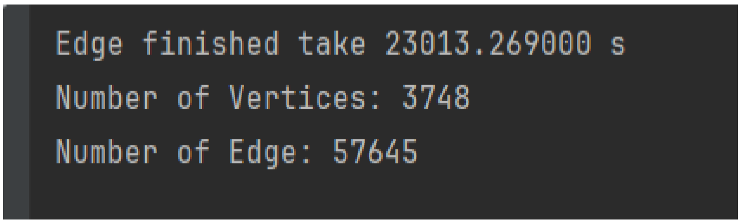


```
Node finished take 151.113000 s
Edge finished take 10301.775000 s
Number of Vertices: 3748
Number of Edge: 57645
```

    b. We can see the time taken was significantly increasing because the data has to be persisted in the disk.
3. FoundationDB:
    a. In this part I was able to set up foundationDb with the given jar file in the zip and Ebay’s internal dependence. The main challenge is to set up FDB and connect with it. I was able to set up the code on an X86 machine by copying the DLL file into the directory.
    b. The result is follow:



```
Node finished take 554.797000 s
```



```
Edge finished take 23013.269000 s
Number of Vertices: 3748
Number of Edge: 57645
```

    c. We can see the result was slower than BerkeleyDB. The result may be that FoundationDB supports ACID, which requires locking and replication in each transaction. Thus slow down the overall time.