

Student:

John Doll

Email:

dolljm@miamioh.edu

Time on Task:

0 hours, 33 minutes

Progress:

42%

Report Generated: Monday, April 10, 2023 at 12:55 PM

## Hands-On Demonstration

### Part 1: Create a Neo4j Database

26. **Make a screen capture** of the **Database Information pane** that shows Node Labels, Relationship Types, and Property Keys.

Incomplete

30. **Make a screen capture** of the **result frame** that shows the query and the graph result.

Incomplete

### Part 2: Query Nodes Using Cypher

9. **Make a screen capture** of the **Database Information pane** to show the Node Labels, Relationship Types, and Property Keys.

Incomplete

12. **Make a screen capture** of the **Text view** of the query results.

Incomplete

17. **Make a screen capture** of the **Graph view** of the query results showing all the nodes.

Incomplete

21. **Make a screen capture** of the **Text** view of the list of majors.

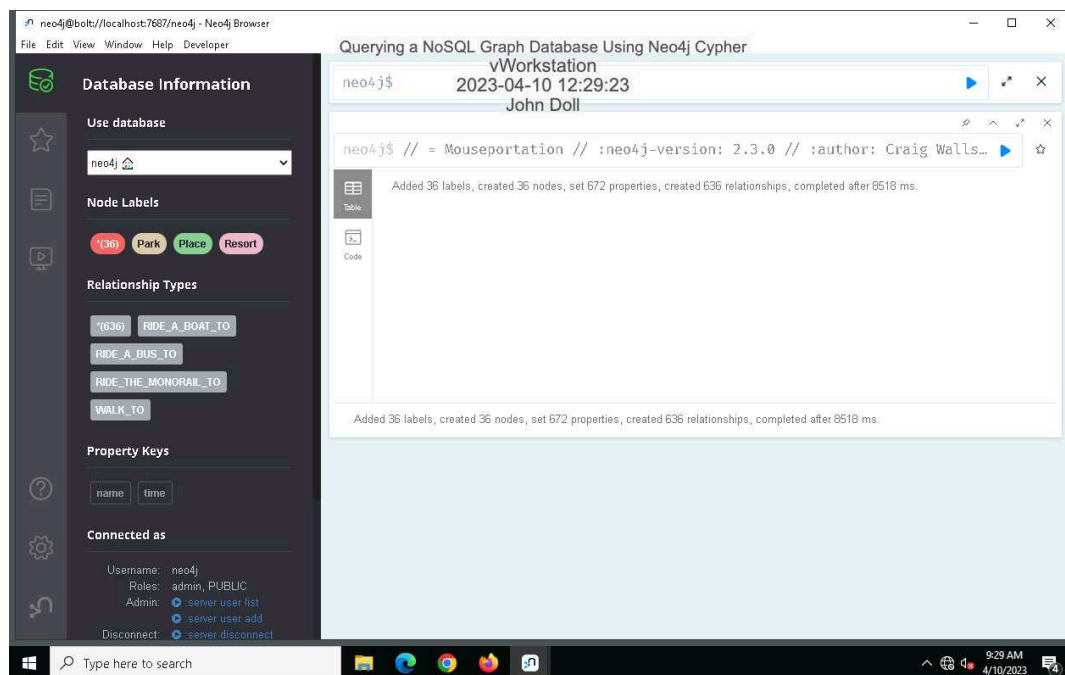
Incomplete

23. **Make a screen capture** of the **Text** view of the results of the previous query.

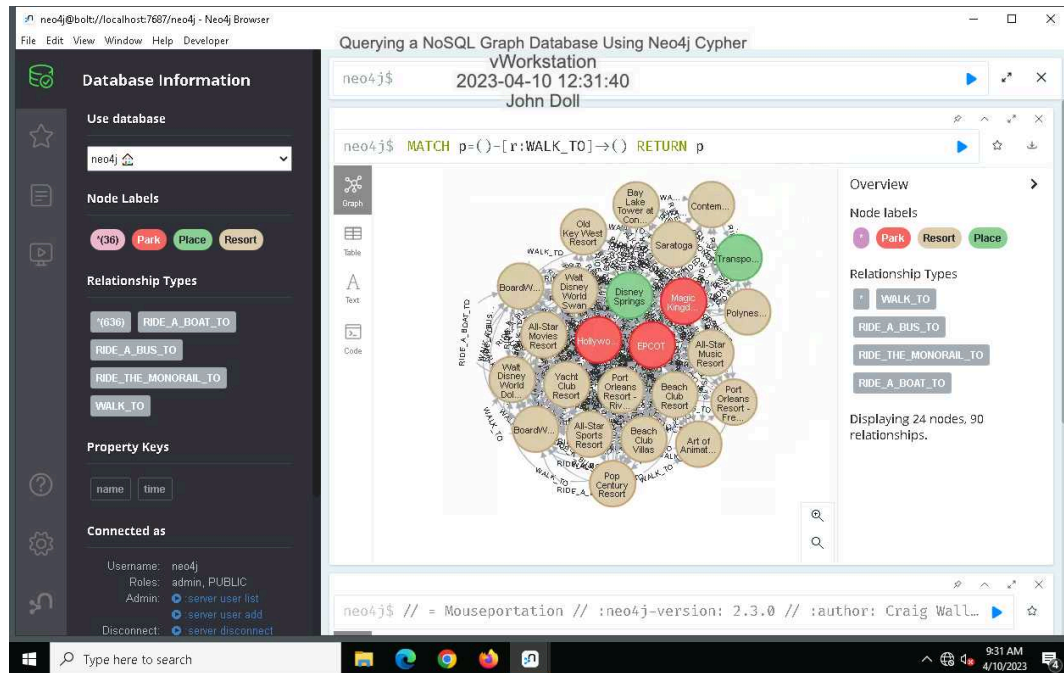
Incomplete

## Part 3: Write Queries over Relationships

14. **Make a screen capture** showing the **Node Labels**, **Relationship Labels**, and **Property Keys**.

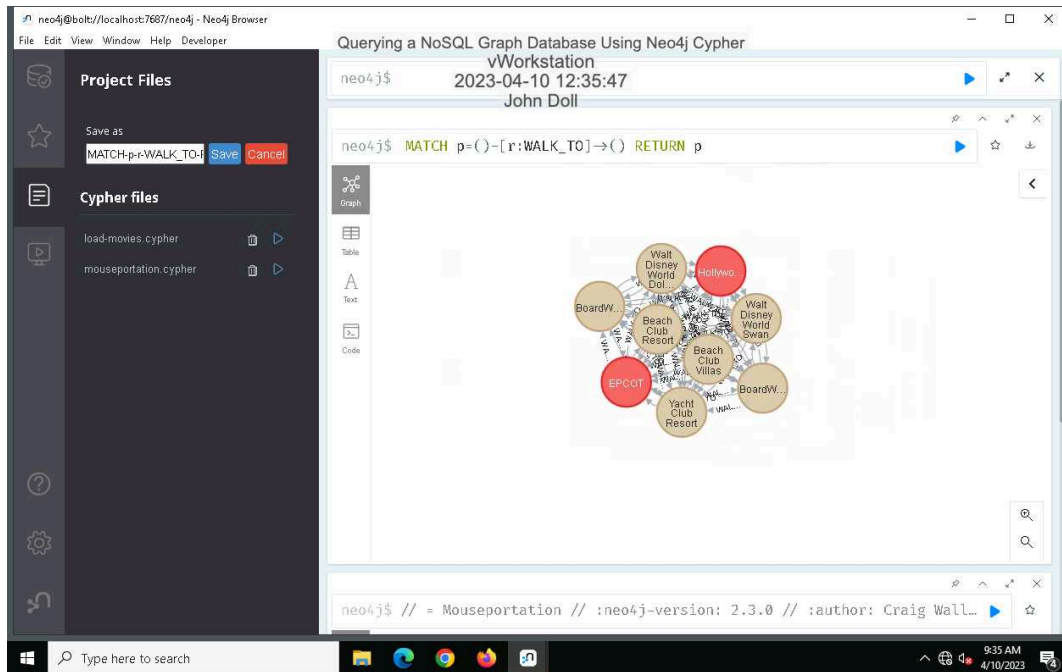


17. **Make a screen capture of the Overview panel** that shows how many **nodes** and **relationships** were returned.

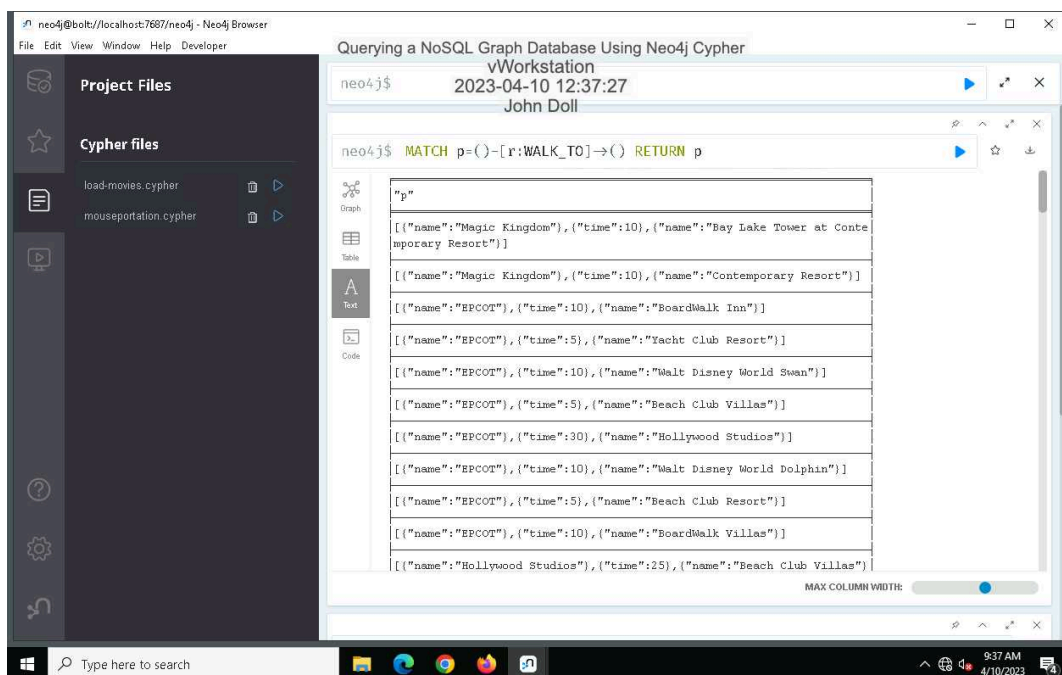


23. **Document the number of isolated (unconnected) subgraphs** that are returned by the query.

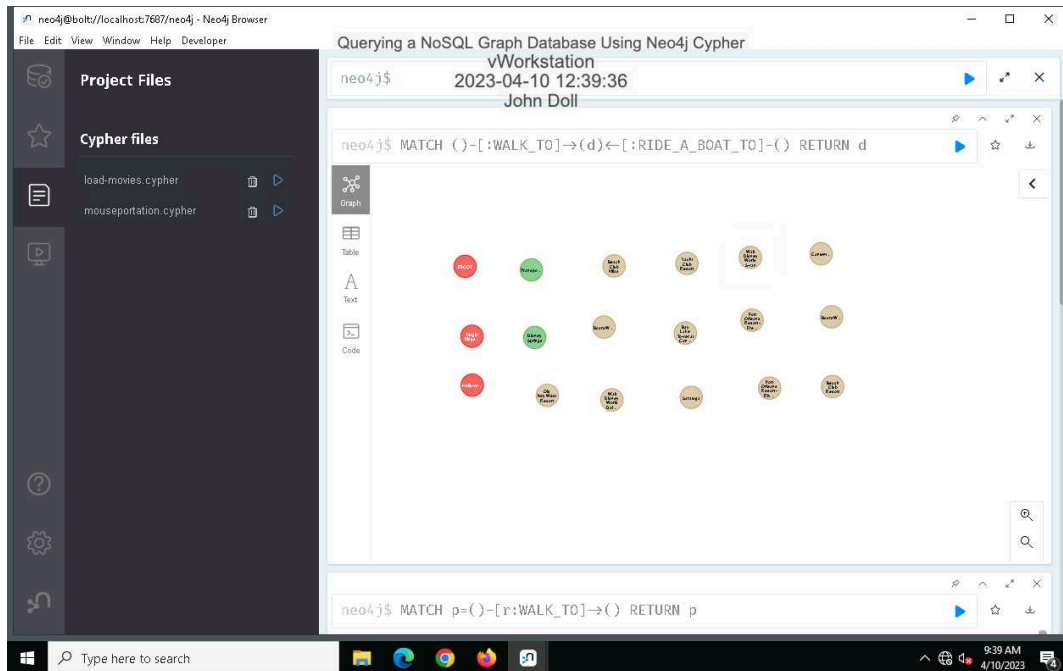
25. Make a screen capture showing the **Graph** view of the **largest subgraph** in the result.



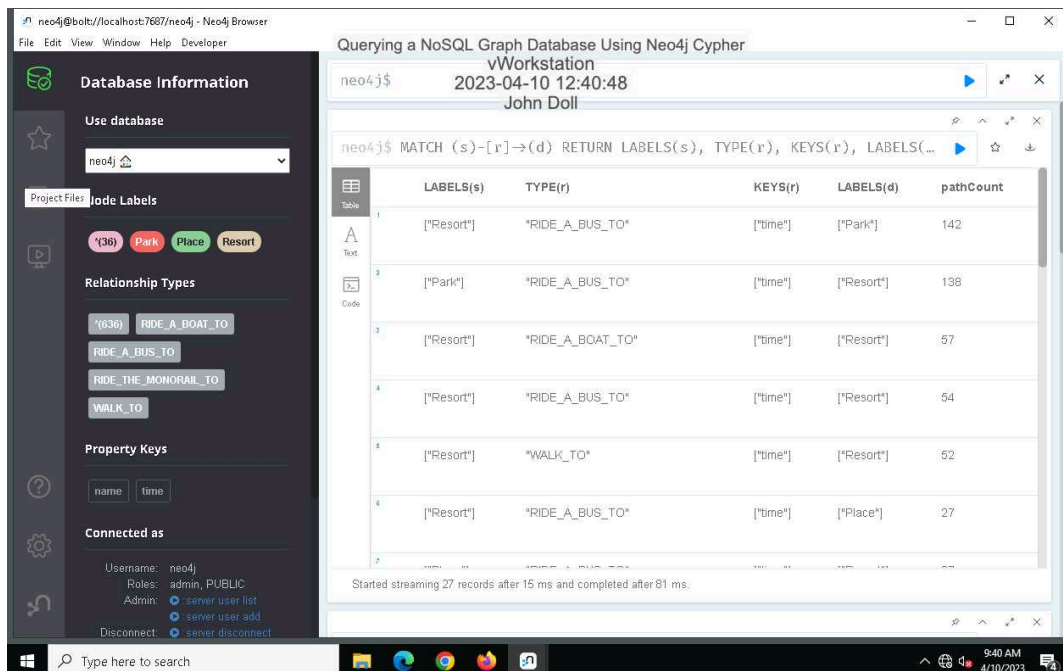
27. Make a screen capture showing the **Text** view of the **first 10 results**.



### 31. Make a screen capture showing the **Graph** view of the query results.



### 33. Make a screen capture showing the first nine rows in the **Text** view of the result.



34. Make a screen capture showing the last two rows in the **Text view** of the result.

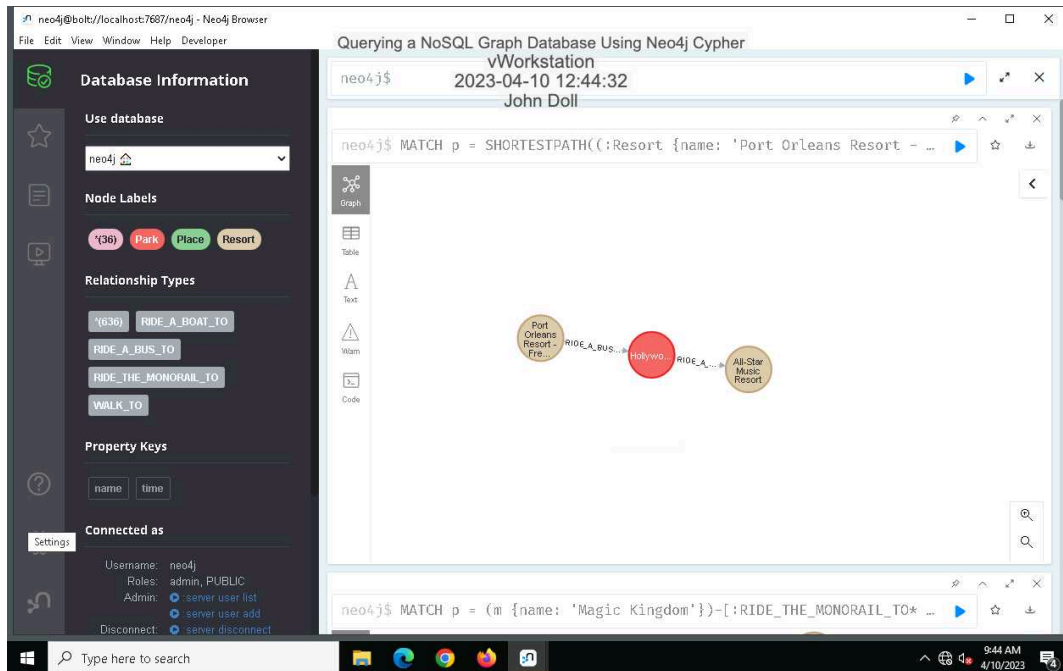
The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is visible, showing the database 'neo4j' and various node labels and relationship types. The main panel displays the Cypher query: `MATCH (s)-[r]-(d) RETURN LABELS(s), TYPE(r), KEYS(r), LABELS(d), pathCount`. The results are shown in a table with 5 columns: LABELS(s), TYPE(r), KEYS(r), LABELS(d), and pathCount. The last two rows of the result set are highlighted.

	LABELS(s)	TYPE(r)	KEYS(r)	LABELS(d)	pathCount
22	["Park"]	"RIDE_THE_MONORAIL_TO"	["time"]	["Place"]	2
23	["Park"]	"RIDE_A_BOAT_TO"	["time"]	["Park"]	2
24	["Park"]	"WALK_TO"	["time"]	["Park"]	2
25	["Place"]	"RIDE_THE_MONORAIL_TO"	["time"]	["Park"]	2
26	["Park"]	"RIDE_A_BOAT_TO"	["time"]	["Place"]	1
27	["Place"]	"RIDE_A_BOAT_TO"	["time"]	["Park"]	1

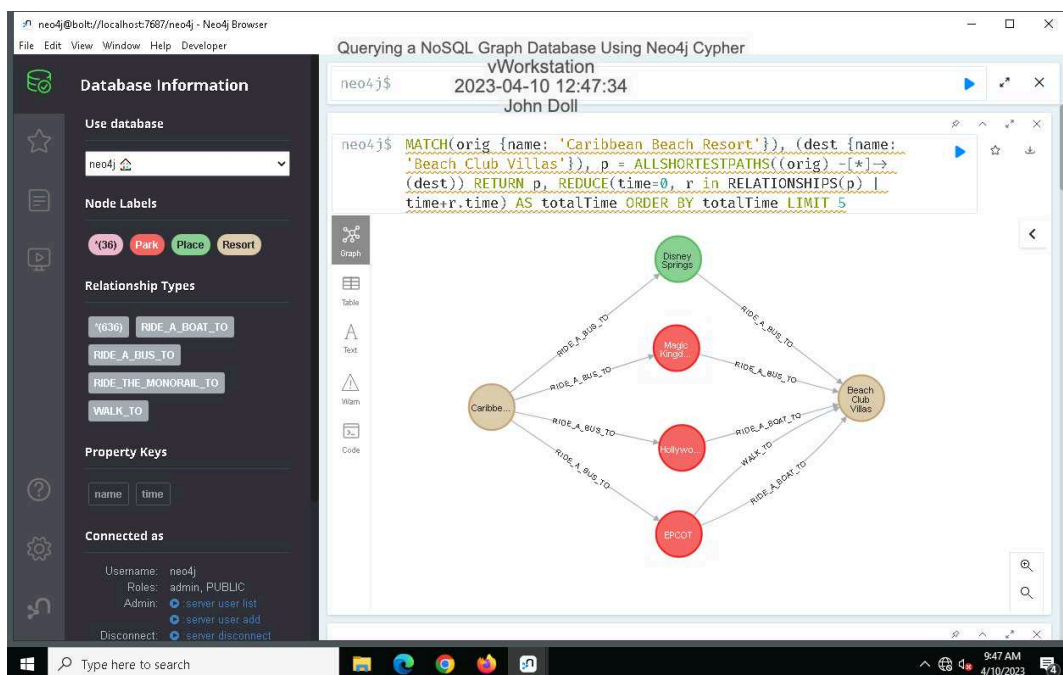
36. Make a screen capture showing the **Graph view** of the result where the relationship labels are readable.

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is visible. The main panel displays the Cypher query: `MATCH p = (m {name: 'Magic Kingdom'})-[:RIDE_THE_MONORAIL_TO]* ...`. The results are shown in a graph view, where nodes are represented by colored circles and relationships by lines. The graph shows a complex network of nodes and relationships, with labels like 'EPCOT', 'The Villages at Grand Floridian', 'Bay Lake Tower at Coronado Springs', 'Polynesian', 'Grand Floridian Resort', 'Contemporary Resort', 'Transcendental', and 'Magic Kingdom' visible. The relationship labels are also visible on the edges.

39. Make a screen capture showing the **Graph view** of the result.



43. Make a screen capture showing the **Graph view** of the result with the source on the left and the destination on the right.





### 44. Make a screen capture of the Text view of the result.

The screenshot shows the Neo4j Browser interface. On the left is a sidebar with 'Database Information', 'Node Labels', 'Relationship Types', 'Property Keys', and 'Connected as'. The main area displays a Cypher query and its results in text view.

**Query:**

```
neo4j$ MATCH (orig {name: 'Caribbean Beach Resort'}) -> (dest {name: 'Beach Club Villas'}) p = ALLSHORTESTPATHS((orig) -[*]-> (dest)) RETURN p, REDUCE(time=0, r in RELATIONSHIPS(p) | time+r.time) AS totalTime ORDER BY totalTime LIMIT 5
```

**Results (Text View):**

"p"	"totalTime"
[{"name": "Caribbean Beach Resort", "time": 15}, {"name": "EPCOT", "time": 20}, {"name": "Beach Club Villas"}]	35
[{"name": "Caribbean Beach Resort", "time": 10}, {"name": "Hollywood Studios", "time": 20}, {"name": "Beach Club Villas"}]	30
[{"name": "Caribbean Beach Resort", "time": 10}, {"name": "Disney Springs", "time": 20}, {"name": "Beach Club Villas"}]	30
[{"name": "Caribbean Beach Resort", "time": 15}, {"name": "Magic Kingdom", "time": 20}, {"name": "Beach Club Villas"}]	35
[{"name": "Caribbean Beach Resort", "time": 15}, {"name": "EPCOT", "time": 20}, {"name": "Beach Club Villas"}]	35

Below the results, another query is visible:

```
neo4j$ MATCH p = SHORTESTPATH((:Resort {name: 'Port Orleans Resort - ...
```

### 45. Document the answer to this question using the results of the query: What **type of transportation** should you use and **where should you go** to travel in the **shortest amount of time** from the Caribbean Beach Resort to the Beach Club Villas?

Ride a bus to Epcot, and ride a boat to Beach Club Villas from Epcot



### 47. Make a screen capture of the Text view of the result.

The screenshot shows the Neo4j Browser interface. On the left is a sidebar with 'Database Information' including 'Use database' (neo4j), 'Node Labels' (Park, Place, Resort), 'Relationship Types' (RIDE\_A\_BOAT\_TO, RIDE\_A\_BUS\_TO, RIDE\_THE\_MONORAIL\_TO, WALK\_TO), 'Property Keys' (name, time), and 'Connected as' (Username: neo4j, Roles: admin, PUBLIC). The main panel is titled 'Querying a NoSQL Graph Database Using Neo4j Cypher'. It shows a Cypher query: `neo4j$ MATCH (n {name: 'EPCOT'})-[r]-> () RETURN TYPE(r), count(*)`. The results are displayed in a table view with columns 'TYPE(r)' and 'count(\*)'. Below the table is a graph view showing a node 'Disney Springs' connected to 'EPCOT' via 'RIDE\_A\_BUS\_TO' and to 'Magic Kingdom' via 'RIDE\_A\_BOAT\_TO'. The 'Relationship Properties' panel on the right shows properties for 'RIDE\_A\_BOAT\_TO': <id> 53 and time 20.

"TYPE(r)"	"count(*)"
"RIDE_A_BUS_TO"	18
"RIDE_THE_MONORAIL_TO"	1
"RIDE_A_BOAT_TO"	8
"WALK_TO"	8

## Challenge and Analysis

### Part 1: To Walk or Not To Walk?

**Make a screen capture** showing the **Text view**.

Incomplete

**Make a screen capture** showing the **Text view**.

Incomplete

**Make a screen capture** showing the **default Graph view**.

Incomplete

**Make a screen capture** showing the **Graph View** arranged so that all six subgraphs are visible.

Incomplete

**Make a screen capture** showing the **Graph view** with all nodes and only the ride a boat to relationship. Arrange the nodes so that they are all visible.

Incomplete

6. **Document** your **answer** to this question:

Incomplete

### Part 2: Polynesian Village Resort Conundrum

**Make a screen capture** showing the **Graph view**.

Incomplete

**Make a screen capture** showing the **Graph view** arranged so that the Polynesian Village Resort is on the left, the Animal Kingdom Lodge is on the right, and all other nodes are in the middle.

Incomplete

**Make a screen capture** showing the **Graph view** arranged so that the Polynesian Village Resort is on the left, the Animal Kingdom Lodge is on the right, and all other nodes are in the middle.

Incomplete

**Make a screen capture** showing the **Text view**.

Incomplete

4. **Document** your **observations** about the differences between the graph views for the previous two results.

Incomplete

**Make a screen capture** showing the **Text view**.

Incomplete