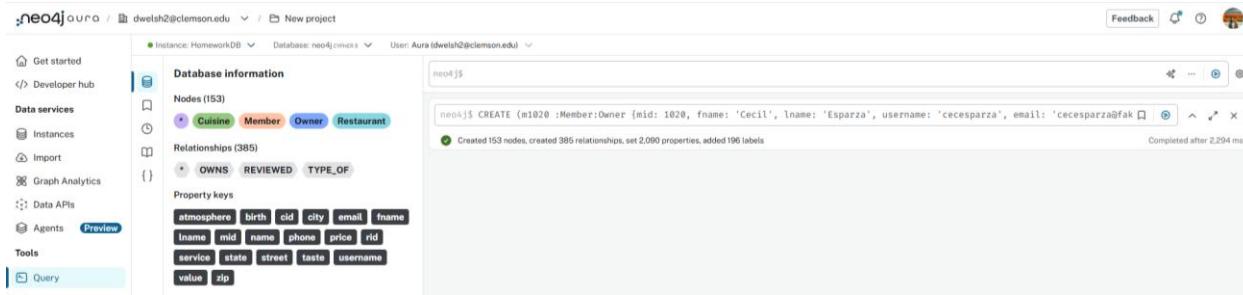


## 0. Database Generation

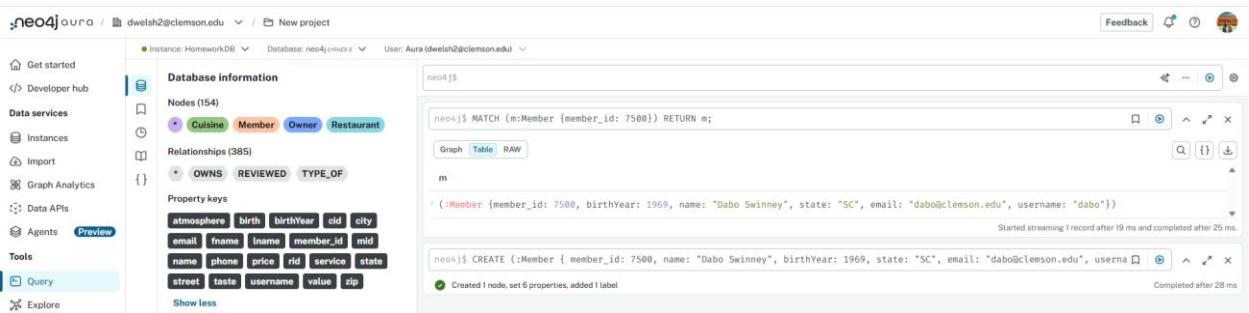


The screenshot shows the Neo4j Aura interface. On the left, the sidebar includes 'Get started', 'Developer hub', 'Data services' (selected), 'Instances', 'Import', 'Graph Analytics', 'Data APIs', 'Agents' (selected), 'Tools', 'Query' (selected), and 'Explore'. The main area displays 'Database information' with 'Nodes (153)' and 'Relationships (385)'. A query window shows the creation of a node: 

```
neo4j$ CREATE (:Member {mid: 1020, fname: 'Cecil', lname: 'Esparza', username: 'cecesparza', email: 'cecesparza@fak...'})
```

 with a note 'Created 153 nodes, created 385 relationships, set 2,000 properties, added 196 labels' and 'Completed after 2,294 ms'.

## 1. Adding Dabo Sweeny to the DB



The screenshot shows the Neo4j Aura interface. The sidebar is identical to the previous one. The main area shows 'Database information' with 'Nodes (154)' and 'Relationships (385)'. A query window shows the creation of a member node: 

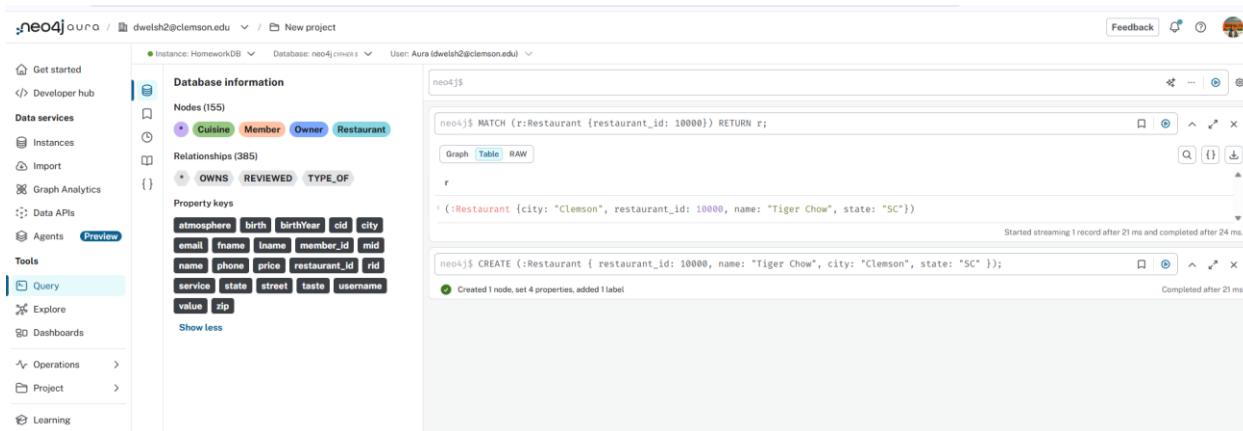
```
neo4j$ MATCH (:Member {member_id: 7500}) RETURN m;
```

 followed by 

```
neo4j$ CREATE (:Member {member_id: 7500, birthYear: 1969, name: "Dabo Swinney", state: "SC", email: "dabo@clemson.edu", username: "dabo"})
```

 with a note 'Started streaming 1 record after 19 ms and completed after 25 ms.' and 'Completed after 28 ms'.

## 2. Setting the Tiger Chow Restaurant



The screenshot shows the Neo4j Aura interface. The sidebar includes 'Get started', 'Developer hub', 'Data services' (selected), 'Instances', 'Import', 'Graph Analytics', 'Data APIs', 'Agents' (selected), 'Tools', 'Query' (selected), 'Explore', 'Dashboards', 'Operations' (selected), 'Project' (selected), and 'Learning'. The main area shows 'Database information' with 'Nodes (155)' and 'Relationships (385)'. A query window shows the creation of a restaurant node: 

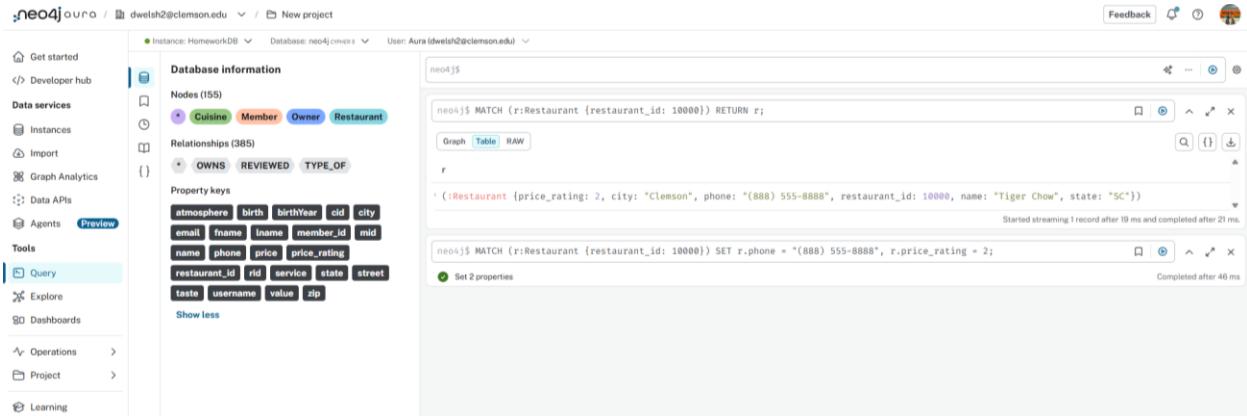
```
neo4j$ MATCH (r:Restaurant {restaurant_id: 10000}) RETURN r;
```

 followed by 

```
neo4j$ CREATE (:Restaurant {city: "Clemson", restaurant_id: 10000, name: "Tiger Chow", state: "SC"})
```

 with a note 'Started streaming 1 record after 21 ms and completed after 24 ms.' and 'Completed after 21 ms'.

### 3. Updating Tiger Chow



The screenshot shows the Neo4j Aura interface. On the left, the sidebar includes 'Get started', 'Developer hub', 'Data services', 'Instances', 'Import', 'Graph Analytics', 'Data APIs', 'Agents' (selected), 'Query' (selected), 'Explore', 'Dashboards', 'Operations', 'Project', and 'Learning'. The 'Query' section is active, showing a database information panel with nodes (155) and relationships (385). Nodes include Cuisine, Member, Owner, and Restaurant. Relationships include OWNS, REVIEWED, and TYPE\_OF. A property keys panel lists atmosphere, birth, birthYear, cid, city, email, fname, lname, member\_id, mid, name, phone, price, price\_rating, restaurant\_id, rid, service, state, street, taste, username, value, and zip. The main area contains a query editor with the following code:

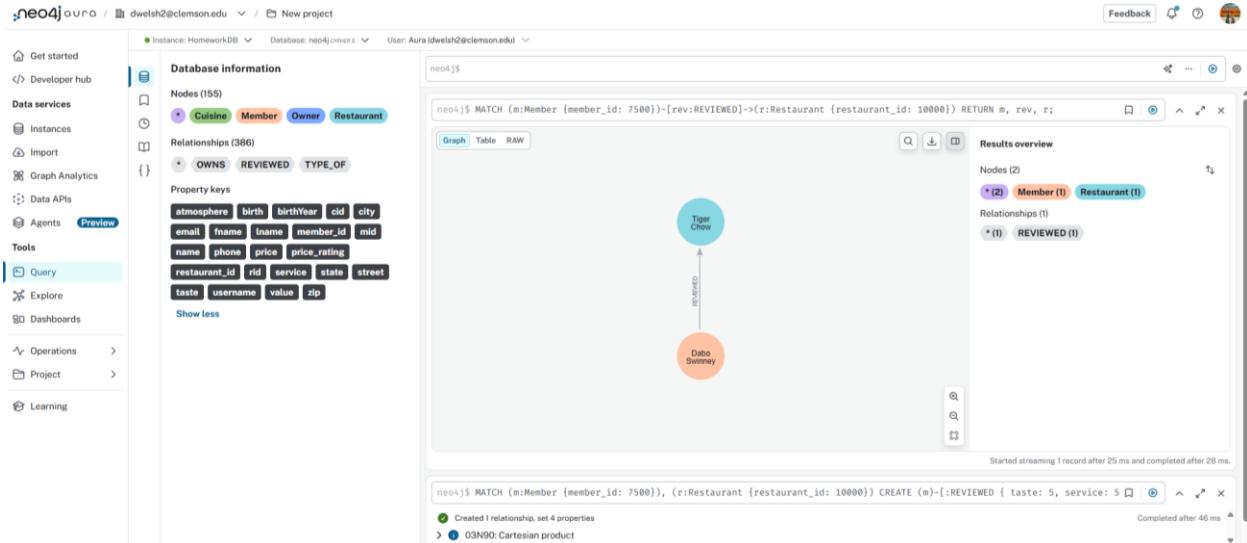
```
neo4j$ MATCH (r:Restaurant {restaurant_id: 10000}) RETURN r;
```

Graph, Table, RAW tabs are shown. The results table shows one record: {Restaurant {price\_rating: 2, city: "Clemson", phone: "(888) 555-8888", restaurant\_id: 10000, name: "Tiger Chow", state: "SC"}}. Below the query editor, another query is shown:

```
neo4j$ MATCH (r:Restaurant {restaurant_id: 10000}) SET r.phone = "(888) 555-8888", r.price_rating + 2;
```

Graph, Table, RAW tabs are shown. The results table shows one record: Set 2 properties. The status bar indicates 'Completed after 46 ms'.

### 4. Creating the Reviewed Relationship



The screenshot shows the Neo4j Aura interface. The sidebar and database information panel are identical to the previous screenshot. The main area shows a query editor with the following code:

```
neo4j$ MATCH (m:Member {member_id: 7500})-[rev:REVIEWED]-(r:Restaurant {restaurant_id: 10000}) RETURN m, rev, r;
```

Graph, Table, RAW tabs are shown. The results overview panel shows 2 nodes (1 Member, 1 Restaurant) and 1 relationship (1 REVIEWED). The graph visualization shows a blue circle labeled 'Tiger Chow' connected by a vertical line to an orange circle labeled 'Daboo Swinney'. The status bar indicates 'Started streaming 1 record after 25 ms and completed after 28 ms'.

Below the query editor, another query is shown:

```
neo4j$ MATCH (m:Member {member_id: 7500}), (r:Restaurant {restaurant_id: 10000}) CREATE (m)-[:REVIEWED { taste: 5, service: 5 }]->(r);
```

Graph, Table, RAW tabs are shown. The results table shows one record: Created 1 relationship, set 4 properties. The status bar indicates 'Completed after 46 ms'.

## 5. Created Fran's reviewed relationship

I should've asked how you'd wanted these formatted, but I'm pushing both images I figured would be relevant

The screenshot shows the Neo4j Aura interface. On the left, the sidebar includes 'Get started', 'Developer hub', 'Data services' (with 'Instances', 'Import', 'Graph Analytics', 'Data APIs'), 'Agents' (selected), 'Tools' (with 'Query' selected), 'Explore', 'Dashboards', 'Operations', 'Project', and 'Learning'. The main area has tabs for 'Database information' (Nodes: 155, Relationships: 387) and 'Relationships' (OWNS, REVIEWED, TYPE\_OF). Below these are 'Property keys' and a 'Show less' link. The central workspace contains a query editor with the following content:

```
neo4j$ MATCH (m:Member {mid: 1089}), (r:Restaurant {restaurant_id: 10000}) CREATE (m)-[:REVIEWED { taste: 4, service: 4, atmo: 2 }]->(r);
```

Below the query is a note: "Completed after 27 ms". The bottom status bar says "Started streaming 1 record after 21 ms and completed after 23 ms."

The screenshot shows the Neo4j Aura interface, identical to the one above in layout and sidebar. The central workspace displays a query and its results.

Query content:

```
neo4j$ MATCH (m:Member {mid: 1089})-[rev:REVIEWED]->(r:Restaurant {restaurant_id: 10000}) RETURN m, rev, r;
```

The results section shows a graph visualization with two nodes: 'Fran' (orange circle) and 'Tiger Chow' (blue circle). A directed edge labeled 'REVIEWED' points from 'Fran' to 'Tiger Chow'. To the right, there is a 'Results overview' panel showing 'Nodes (2)' (Fran, Tiger Chow), 'Relationships (1)' (Fran -> Tiger Chow), and a 'Relationships (1)' section for 'REVIEWED'.

The bottom status bar says "Started streaming 1 record after 21 ms and completed after 23 ms."

## 6. Queried the restaurants that Fran could stand

**neo4jaura** / dwelsh2@clemson.edu / New project

Instance: HomeworkDB Database: neo4j:cineplex User: Aura (dwelsh2@clemson.edu)

**Database information**

Nodes (155) **Cuisine** **Member** **Owner** **Restaurant**

Relationships (388) **OWNS** **REVIEWED** **TYPE\_OF**

Property keys **atmosphere** **birth** **birthYear** **cid** **city** **email** **fname** **lname** **member\_id** **mid** **name** **phone** **price** **price\_rating** **restaurant\_id** **rid** **service** **state** **street** **taste** **username** **value** **zip**

Show less

```
neo4j$ MATCH (m:Member {mid: 1089})->(r:Restaurant) WHERE rev.value >= 1 RETURN m.fname + ' ' + m.lname AS member, r.name AS restaurant, r.value_rating AS value_rating
Table RAW
member          restaurant      value_rating
"Fran Dagwood" "Butcher Paper Steaks" 2
"Fran Dagwood" "Marlins"           1
"Fran Dagwood" "Tiger Chow"        4
"Fran Dagwood" "Tiger Chow"        4
```

Started streaming 4 records after 57 ms and completed after 62 ms.

## 7. Queried the Italian and Seafood Restaurants

**neo4jaura** / dwelsh2@clemson.edu / New project

Instance: HomeworkDB Database: neo4j:cineplex User: Aura (dwelsh2@clemson.edu)

**Database information**

Nodes (155) **Cuisine** **Member** **Owner** **Restaurant**

Relationships (388) **OWNS** **REVIEWED** **TYPE\_OF**

Property keys **atmosphere** **birth** **birthYear** **cid** **city** **email** **fname** **lname** **member\_id** **mid** **name** **phone** **price** **price\_rating** **restaurant\_id** **rid** **service** **state** **street** **taste** **username** **value** **zip**

Show less

```
neo4j$ MATCH (o:Owner)-[:OWNS]->(r:Restaurant)-[:TYPE_OF]->(c:Cuisine) WHERE c.name IN ["Italian", "Seafood"] RETURN c.name AS cuisine, r.name AS restaurant, o.fname + ' ' + o.lname AS owner
Table RAW
cuisine          restaurant      owner
"Italian"       "Authentica Rustica" "Dolly Michael"
"Italian"       "The Sicilian"       "Lea Sterling"
"Italian"       "Maestro Grill"     "Ashton Johnston"
"Italian"       "The Pasta Bowl"     "Rachel Avery"
"Italian"       "Guidos Cucina"       "Harland Roberts"
"Italian"       "Taste of Rome"        "Ruben Fish"
"Seafood"        "Captain Seafood"    "Carlene Gonzalez"
"Seafood"        "Hook, Line, and Stinker" "Ezekial Pineda"
"Seafood"        "Marlins"           "Precious Weir"
"Seafood"        "Boathouse Floaters" "Leigh Fowler"
```

Started streaming 13 records after 29 ms and completed after 32 ms.

## 8. Find the shortest path between Dabo and Herb

**neo4jaura** / dwelsh2@clemson.edu / New project

Instance: HomeworkDB Database: neo4j:cineplex User: Aura (dwelsh2@clemson.edu)

**Database information**

Nodes (155) **Cuisine** **Member** **Owner** **Restaurant**

Relationships (388) **OWNS** **REVIEWED** **TYPE\_OF**

Property keys **atmosphere** **birth** **birthYear** **cid** **city** **email** **fname** **lname** **member\_id** **mid** **name** **phone** **price** **price\_rating** **restaurant\_id** **rid** **service** **state** **street** **taste** **username** **value** **zip**

Show less

```
1 MATCH
2   (d:Member {mid: 7500}),      // Dabo Swinney
3   (h:Member {mid: 1091})       // Herb Christopher
4   MATCH p = shortestPath(
5     (d)-[":REVIEWED*"]-(h)    // only REVIEWED edges allowed
6   )
7   RETURN p;
```

No changes, no records  
2 info messages

```
neo4j$ MATCH (d:Member {fname: "Dabo", lname: "Swinney"}), (h:Member {fname: "Herb", lname: "Christopher"}) MATCH p = shortestP
```

Completed after 34 ms

No changes, no records  
2 info messages

Completed after 66 ms