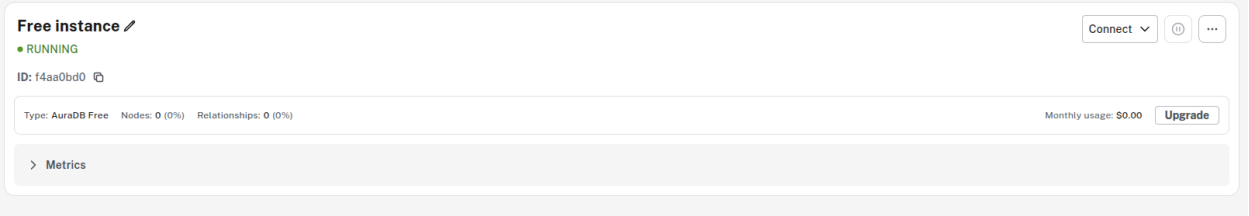
Fall 2025: CS 224P, UCI **Instructor: Prof Chen Li** **TA's: Sarah Asad, Ali Risheh**

Homework Assignment #4

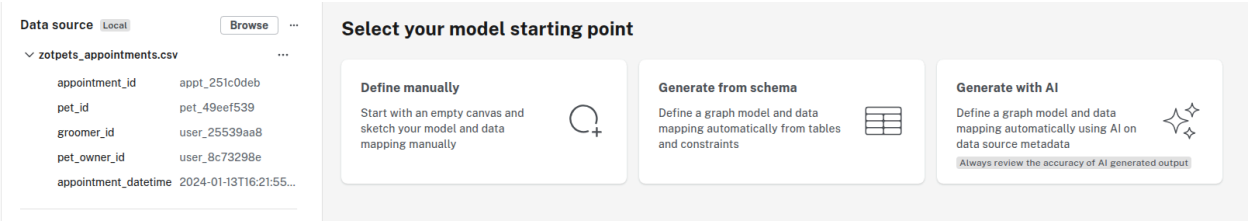
Deadline: 11/06/2026 (Thursday) 11:45PM

Setup Neo4j

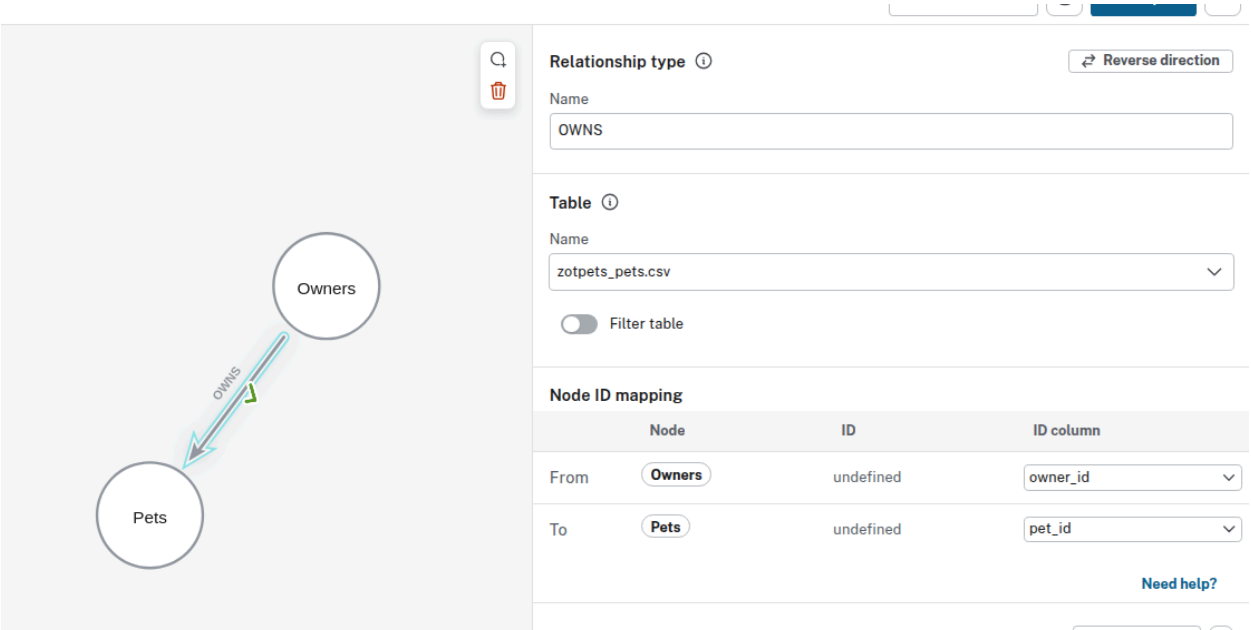
1. Go to neo4j.com and start a new account, it will create a free instance for you:



2. Next go to the "Import" tab and upload CSV files of this assignment.
3. To learn the fundamentals of a graph database, select "Define manually" after uploading files:



4. Next you can define your graph model of data visually, the following is an example of two nodes:



The above screenshot is equal to the following command:

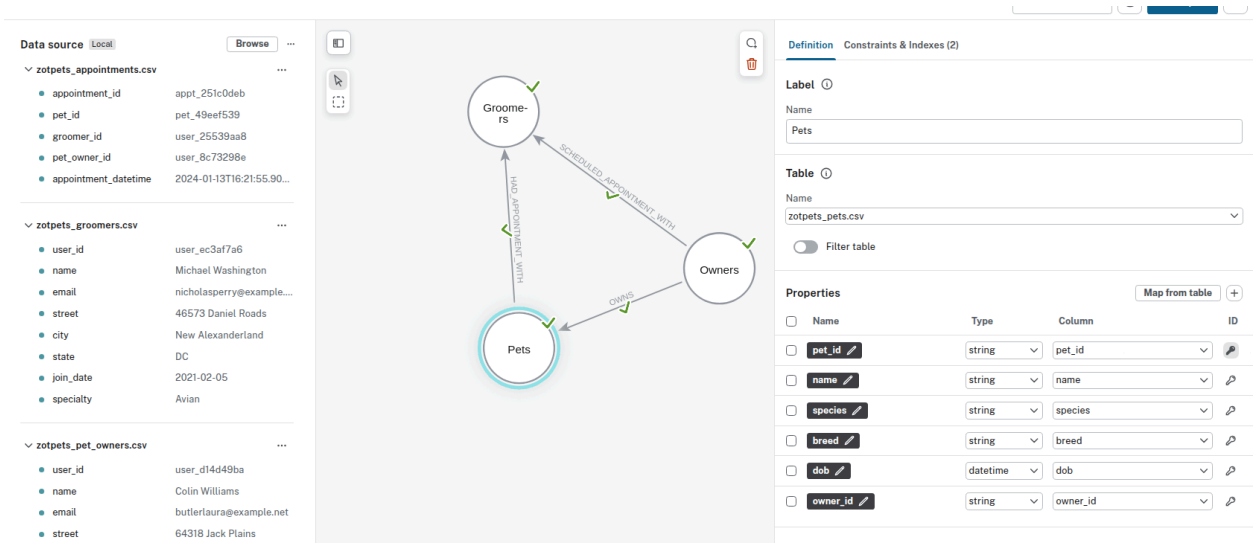
```
SQL

MATCH (o:PetOwner {user_id: row.owner_id})
MATCH (p:Pet {pet_id: row.pet_id})
CREATE (o)-[:OWNS]->(p);
```

Create the full graph corresponding to the following Cyphers:

```
SQL

MATCH (p:Pet {pet_id: row.pet_id})
MATCH (g:Groomer {user_id: row.groomer_id})
MATCH (o:PetOwner {user_id: row.pet_owner_id})

// Create relationship between Pet and Groomer
CREATE (p)-[:HAD_APPOINTMENT_WITH {
    datetime: datetime(row.appointment_datetime),
    appointment_id: row.appointment_id
}]->(g)

// Create relationship between Owner and Groomer
CREATE (o)-[:SCHEDULED_APPOINTMENT_WITH {
    datetime: datetime(row.appointment_datetime),
```

```
    appointment_id: row.appointment_id
}]->(g);
```

Your final graph should look like this:



5. After creating the graph, click on "Import Data". It will give an error about missing an id for a node, you should understand the structure of the graph and solve the issue. In your final report, explain how you solved the problem.

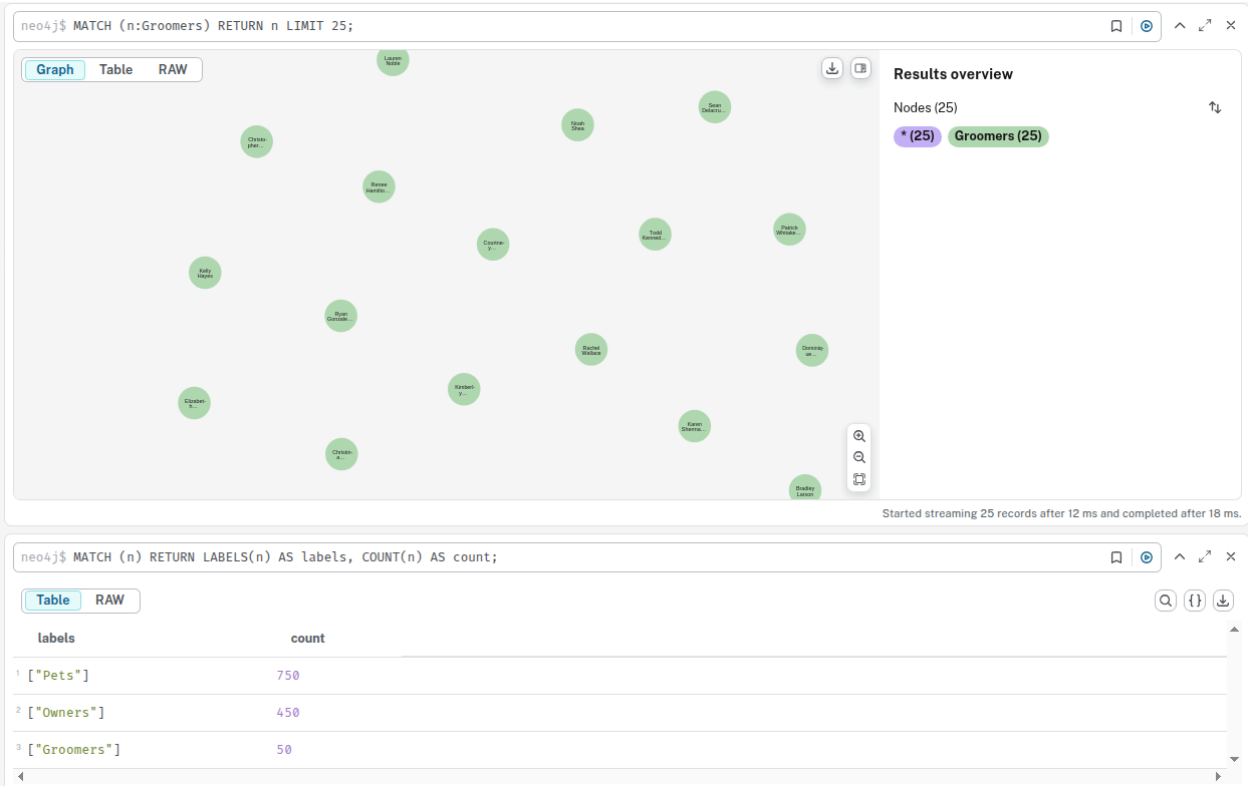
The ZotPets Network

Your boss from the ZotPets relational database project was once again very satisfied with your work, but went to a party last Friday where there was lots of buzz about graph databases. All of their friends seem to be using them now! You have thus been asked to continue working at the startup, but to use Neo4j for the next set of tasks.

Specifically, your boss wants to understand the *social network* of Pet Owners to see how clients are connected (e.g., by using the same groomers) to create a new referral program. They want to find "influential" clients and pets within this new network.

Explore Neo4j cloud

Move to the "Query" tab and start running basic queries to explore the database.



Question 1

To start, your boss wants to send a "thank you" gift to one of the clinic's earliest clients. Write a Cypher query to find the `name`, `email`, and `join_date` of the `PetOwner` who has the single earliest (oldest) `join_date`.

Question 2

To identify key "connectors" in the referral network, your boss wants to find the most popular groomer. Write a Cypher query to find the `name` of the `Groomer` who has had appointments with the highest number of *distinct* pets. Also return this count.

(Hint: You'll need to match `Groomers` nodes, match the `HAD_APPOINTMENT_WITH` relationship coming from `Pets` nodes, `COUNT` the `DISTINCT` pets, and then `ORDER BY` the count in `DESC` (descending) order, taking the `LIMIT 1`.)

Question 3

Your boss now wants to identify the most "influential" pet owners. A good proxy for this is finding the owner who has the most pets registered with ZotPets. Write a Cypher query to find the `name` of the `Owners` (PetOwner) who owns the *most* `Pets`. Also, return the count of their pets.

Question 4

Now for the ultimate "connectivity" question! Your boss wants to see the "Six Degrees of ZotPets" in action. Write a query to find the **shortest path** between the earliest client (from Q1) and the most "influential" client (from Q3). You can use 10 as the maximum number of hops in shortestPath function.

Your query should return the entire path, which the Neo4j console will visualize as a graph. Add the screenshot of the graph to your submission.

Question 5

Your boss is impressed with the 'most popular' groomer (from Q2). Now, they want to identify other groomers who share clients with this top groomer, as this could reveal important community connections (or competitors).

Write a single Cypher query that first finds the most popular `Groomer` (the one with the most distinct `Pets`, from Q2), and then finds all *other* `Groomers` who have had an appointment with at least one `Pet` that the most popular groomer has also seen.

Return the `name` of these "secondary" groomers and the `count` of the distinct pets they *share* with the most popular groomer. Order the results by the shared pet count in descending order.

Question 6

For the final task, your boss wants to visualize the entire "referral network" of the *most influential client* (the owner from Q3).

Write a single Cypher query that will return a graph showing:

1. The most influential `Owners` node (from Q3).
2. All the `Pets` they `OWNS`.
3. All the `Groomers` those pets have `HAD_APPOINTMENT_WITH`.
4. All the *other* `Pets` that have also visited those same groomers.
5. The `Owners` of those *other* pets.

This will create a large, rich graph visualization of your top client's community.

What to turn in?

For this homework, you should submit two files:

- A PDF file that includes all the cypher queries, commands, and execution results
- A TXT file that includes all the cypher queries

Notes:

- Write queries and results in the order of questions in the homework. For example, your first query and its results should belong to question 1.
- We can run your queries during grading. So make sure the sequence of commands and queries for each question is correct.