

Name: Prem Kumar Yadav
Jaguar ID: J01280187

To run the program:

```
# Create virtual environment  
python -m venv venv  
# Activate virtual environment (on mac)  
source venv/bin/activate
```

```
#Install the requirements  
pip install -r requirements.txt
```

```
# Start Neo4j using Docker Compose  
docker-compose up -d
```

```
# After 30s, Check if it's running  
docker-compose ps
```

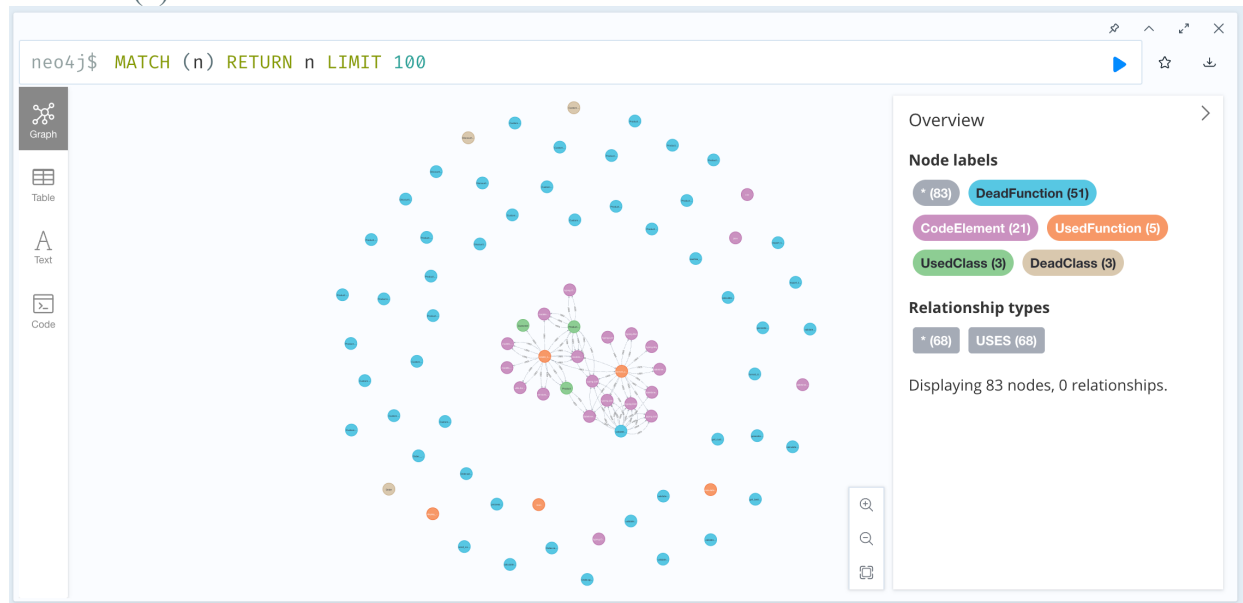
Then, Open browser and
<http://localhost:7474>
Login ID: neo4j
Password: password

Then run the command to get the results:
python3 deadcode.py ./sample_code --output sample_code_results.txt

Then Use these Cypher Queries to view the results:

1. To view graph database of the sample code

```
MATCH (n) RETURN n LIMIT 100
```



2. Get overall statistics of the codebase

```
MATCH (e:CodeElement)
```

```
RETURN
```

```
COUNT(e) as total_elements,
```

```
SUM(CASE WHEN e.is_used THEN 1 ELSE 0 END) as used_elements,
```

```
SUM(CASE WHEN e.is_used THEN 0 ELSE 1 END) as unused_elements,
```

```
ROUND(100.0 * SUM(CASE WHEN e.is_used THEN 0 ELSE 1 END) / COUNT(e), 2) as
```

```
dead_code_percentage;
```

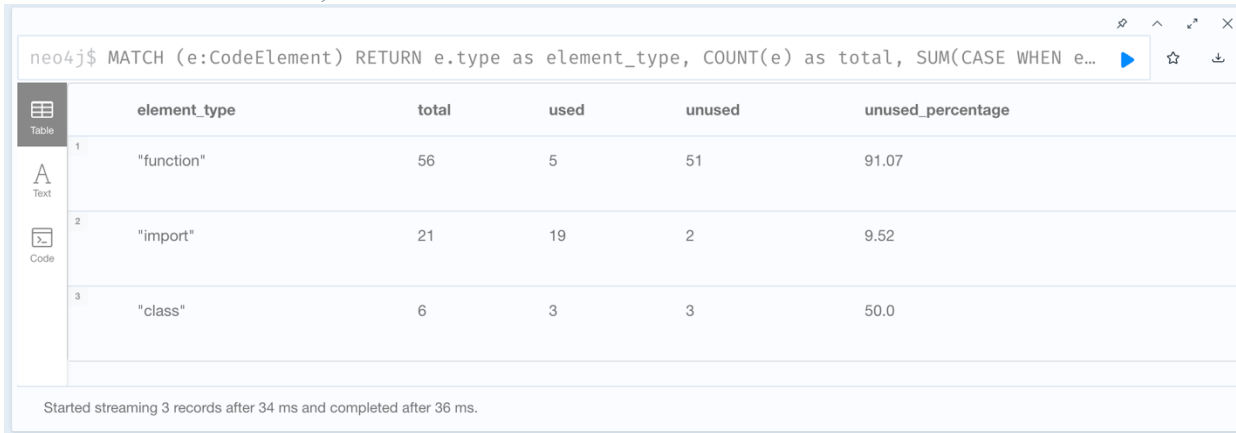
The image shows the Neo4j interface with the Cypher query `neo4j$ MATCH (e:CodeElement) RETURN COUNT(e) as total_elements, SUM(CASE WHEN e.is_used THEN 1... RETURN` executed. The main view displays a table with the following data:

	total_elements	used_elements	unused_elements	dead_code_percentage
1	83	27	56	67.47

Started streaming 1 records after 13 ms and completed after 15 ms.

3. Break down statistics by functions, classes, imports

```
MATCH (e:CodeElement)
RETURN
e.type as element_type,
COUNT(e) as total,
SUM(CASE WHEN e.is_used THEN 1 ELSE 0 END) as used,
SUM(CASE WHEN e.is_used THEN 0 ELSE 1 END) as unused,
ROUND(100.0 * SUM(CASE WHEN e.is_used THEN 0 ELSE 1 END) / COUNT(e), 2) as unused_percentage
ORDER BY total DESC;
```



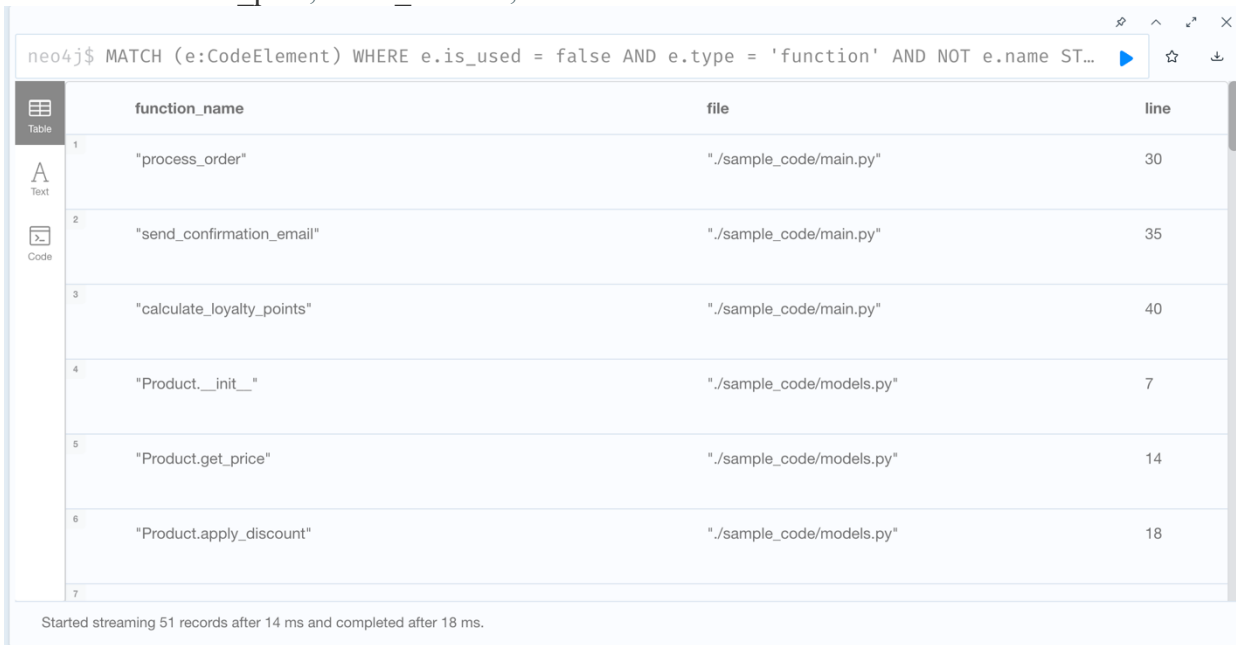
The image shows a Neo4j Cypher query interface. The query is: `neo4j$ MATCH (e:CodeElement) RETURN e.type as element_type, COUNT(e) as total, SUM(CASE WHEN e...` . The results are displayed in a table with 5 columns: `element_type`, `total`, `used`, `unused`, and `unused_percentage`. The table contains 3 records. The first record is for `"function"` with a total of 56, 5 used, 51 unused, and 91.07% unused percentage. The second record is for `"import"` with a total of 21, 19 used, 2 unused, and 9.52% unused percentage. The third record is for `"class"` with a total of 6, 3 used, 3 unused, and 50.0% unused percentage. The interface also shows a status bar at the bottom: "Started streaming 3 records after 34 ms and completed after 36 ms."

	element_type	total	used	unused	unused_percentage
1	"function"	56	5	51	91.07
2	"import"	21	19	2	9.52
3	"class"	6	3	3	50.0

Started streaming 3 records after 34 ms and completed after 36 ms.

4. List all unused functions with their locations

```
MATCH (e:CodeElement)
WHERE e.is_used = false
AND e.type = 'function'
AND NOT e.name STARTS WITH '_'
AND NOT e.name IN ['main', '__init__']
RETURN
e.name as function_name,
e.file_path as file,
e.line_number as line
ORDER BY e.file_path, e.line_number;
```



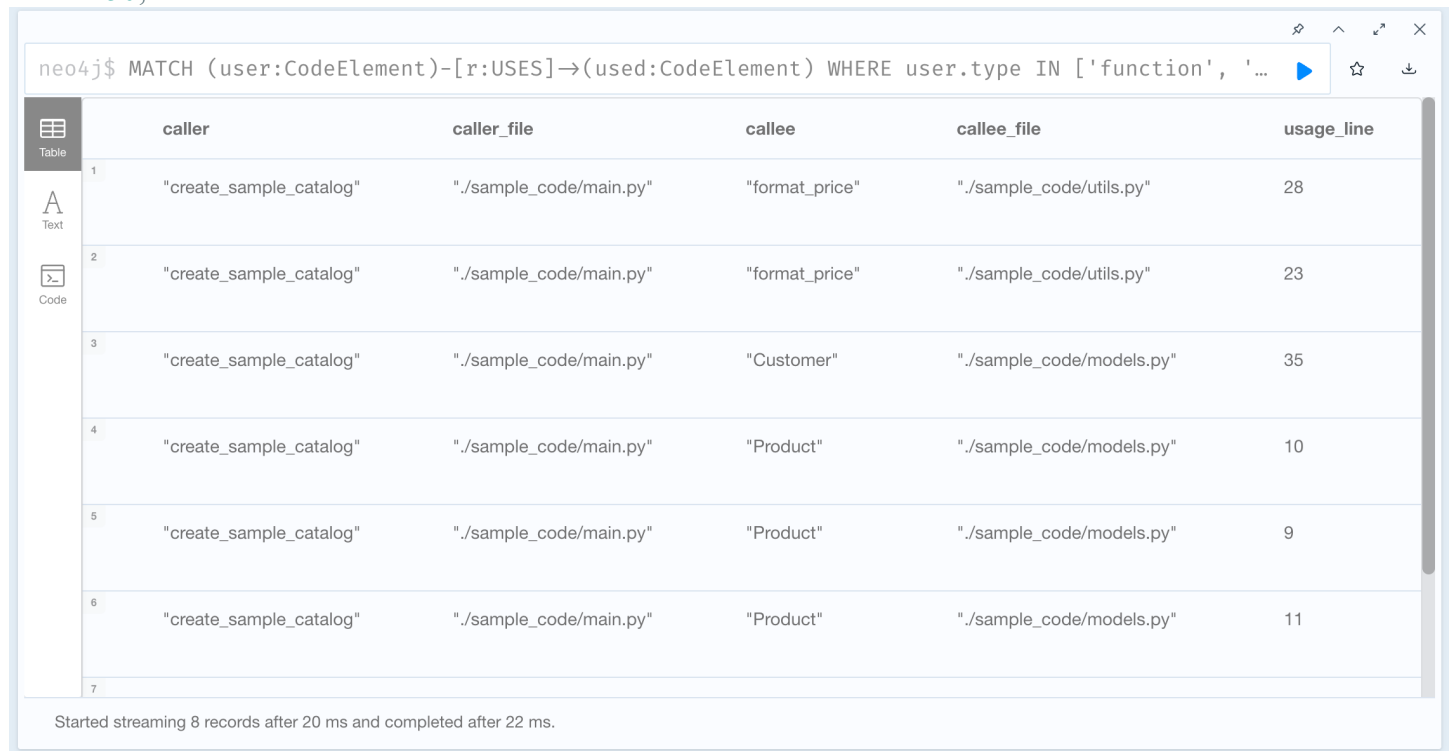
The image shows a Neo4j Cypher query interface. The query is: `neo4j$ MATCH (e:CodeElement) WHERE e.is_used = false AND e.type = 'function' AND NOT e.name ST...`. The results are displayed in a table with 3 columns: `function_name`, `file`, and `line`. The table contains 6 records. The first record is for `"process_order"` in `"/sample_code/main.py"` at line 30. The second record is for `"send_confirmation_email"` in `"/sample_code/main.py"` at line 35. The third record is for `"calculate_loyalty_points"` in `"/sample_code/main.py"` at line 40. The fourth record is for `"Product.__init__"` in `"/sample_code/models.py"` at line 7. The fifth record is for `"Product.get_price"` in `"/sample_code/models.py"` at line 14. The sixth record is for `"Product.apply_discount"` in `"/sample_code/models.py"` at line 18. The interface also shows a status bar at the bottom: "Started streaming 51 records after 14 ms and completed after 18 ms."

	function_name	file	line
1	"process_order"	"/sample_code/main.py"	30
2	"send_confirmation_email"	"/sample_code/main.py"	35
3	"calculate_loyalty_points"	"/sample_code/main.py"	40
4	"Product.__init__"	"/sample_code/models.py"	7
5	"Product.get_price"	"/sample_code/models.py"	14
6	"Product.apply_discount"	"/sample_code/models.py"	18

Started streaming 51 records after 14 ms and completed after 18 ms.

5. Show who uses what (for graph visualization)

```
MATCH (user:CodeElement)-[r:USES]->(used:CodeElement)
WHERE user.type IN ['function', 'class']
AND used.type IN ['function', 'class']
RETURN
user.name as caller,
user.file_path as caller_file,
used.name as callee,
used.file_path as callee_file, r.line_number as usage_line
LIMIT 50;
```



The screenshot shows a Neo4j query interface. At the top, a text input field contains the following Cypher query:

```
neo4j$ MATCH (user:CodeElement)-[r:USES]->(used:CodeElement) WHERE user.type IN ['function', '...'
```

Below the query input, there is a table view showing the results of the query. The table has six columns: **caller**, **caller_file**, **callee**, **callee_file**, and **usage_line**. The results are as follows:

	caller	caller_file	callee	callee_file	usage_line
1	"create_sample_catalog"	"/sample_code/main.py"	"format_price"	"/sample_code/utils.py"	28
2	"create_sample_catalog"	"/sample_code/main.py"	"format_price"	"/sample_code/utils.py"	23
3	"create_sample_catalog"	"/sample_code/main.py"	"Customer"	"/sample_code/models.py"	35
4	"create_sample_catalog"	"/sample_code/main.py"	"Product"	"/sample_code/models.py"	10
5	"create_sample_catalog"	"/sample_code/main.py"	"Product"	"/sample_code/models.py"	9
6	"create_sample_catalog"	"/sample_code/main.py"	"Product"	"/sample_code/models.py"	11
7					

At the bottom of the interface, a status bar indicates: "Started streaming 8 records after 20 ms and completed after 22 ms."

6. Find code with no relationships (neither using nor being used)

```
MATCH (e:CodeElement)
WHERE NOT (e)-[:USES]-()
AND NOT ()-[:USES]->(e)
AND e.type IN ['function', 'class']
AND e.is_used = false
RETURN
e.type as type,
e.name as name,
e.file_path as file,
e.line_number as line
```

ORDER BY e.file_path;

neo4j\$ MATCH (e:CodeElement) WHERE NOT (e)-[:USES]-() AND NOT ()-[:USES]→(e) AND e.type IN ['...'

	type	name	file	line
1	"function"	"process_order"	"/sample_code/main.py"	30
2	"function"	"send_confirmation_email"	"/sample_code/main.py"	35
3	"function"	"calculate_loyalty_points"	"/sample_code/main.py"	40
4	"function"	"Product.__init__"	"/sample_code/models.py"	7
5	"function"	"Product.get_price"	"/sample_code/models.py"	14
6	"function"	"Product.apply_discount"	"/sample_code/models.py"	18
7				

Started streaming 53 records after 24 ms and completed after 28 ms.

7. Find all unused functions and classes

MATCH (e:CodeElement)
WHERE e.is_used = false
AND e.type IN ['function', 'class']
AND NOT e.name STARTS WITH ' '
AND NOT e.name IN ['main', '__init__']
RETURN e.name, e.type, e.file_path, e.line_number
ORDER BY e.file_path, e.line_number;

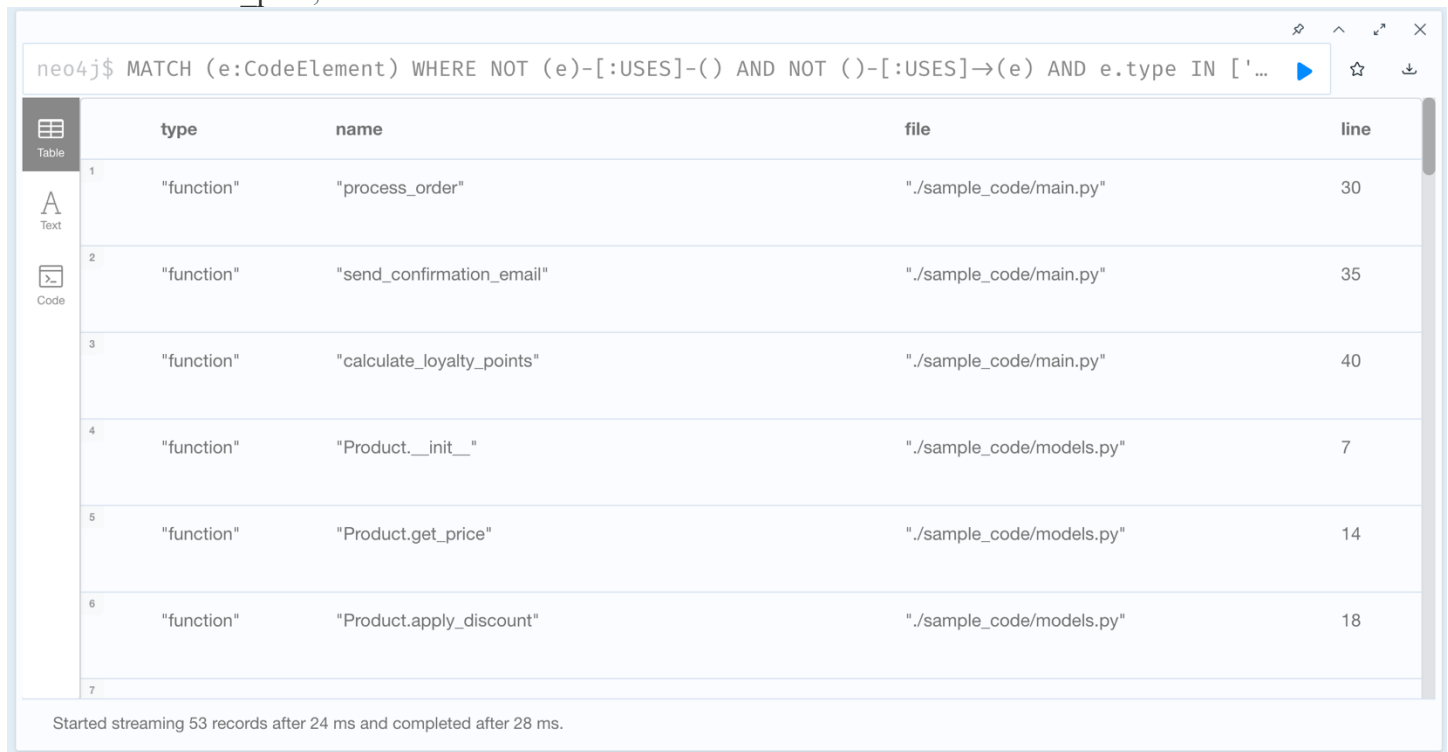
neo4j\$ MATCH (e:CodeElement) WHERE e.is_used = false AND e.type IN ['function', 'class'] AND N...

	e.name	e.type	e.file_path	e.line_number
1	"process_order"	"function"	"/sample_code/main.py"	30
2	"send_confirmation_email"	"function"	"/sample_code/main.py"	35
3	"calculate_loyalty_points"	"function"	"/sample_code/main.py"	40
4	"Product.__init__"	"function"	"/sample_code/models.py"	7
5	"Product.get_price"	"function"	"/sample_code/models.py"	14
6	"Product.apply_discount"	"function"	"/sample_code/models.py"	18
7				

Started streaming 54 records after 13 ms and completed after 14 ms.

8. Find orphaned code elements (no incoming or outgoing relationships)

```
MATCH (e:CodeElement)
WHERE NOT (e)-[:USES]-() AND NOT ()-[:USES]->(e)
AND e.type IN ['function', 'class']
RETURN e.name, e.type, e.file_path, e.line_number
ORDER BY e.file_path;
```



neo4j\$ MATCH (e:CodeElement) WHERE NOT (e)-[:USES]-() AND NOT ()-[:USES]->(e) AND e.type IN ['...]

	type	name	file	line
1	"function"	"process_order"	"/sample_code/main.py"	30
2	"function"	"send_confirmation_email"	"/sample_code/main.py"	35
3	"function"	"calculate_loyalty_points"	"/sample_code/main.py"	40
4	"function"	"Product.__init__"	"/sample_code/models.py"	7
5	"function"	"Product.get_price"	"/sample_code/models.py"	14
6	"function"	"Product.apply_discount"	"/sample_code/models.py"	18
7				

Started streaming 53 records after 24 ms and completed after 28 ms.

9. Find unused imports

```
MATCH (e:CodeElement {type: 'import'})
WHERE e.is_used = false
RETURN e.name, e.file_path, e.line_number
ORDER BY e.file_path, e.line_number;
```



neo4j\$ MATCH (e:CodeElement {type: 'import'}) WHERE e.is_used = false RETURN e.name, e.file_pa...

	e.name	e.file_path	e.line_number
1	"typing.Optional"	"/sample_code/models.py"	2
2	"datetime.timedelta"	"/sample_code/validators.py"	29

Started streaming 2 records after 14 ms and completed after 15 ms.

10. Get usage statistics by file

```
MATCH (e:CodeElement)
RETURN e.file_path,
e.type,
COUNT(e) as total,
SUM(CASE WHEN e.is_used THEN 1 ELSE 0 END) as used,
SUM(CASE WHEN e.is_used THEN 0 ELSE 1 END) as unused
```

ORDER BY e.file_path, e.type;

```
neo4j$ MATCH (e:CodeElement) RETURN e.file_path, e.type, COUNT(e) as total, SUM(CASE WHEN e.is_used THEN 1 ELSE 0 END) as used, SUM(C...
```

	e.file_path	e.type	total	used	unused
1	"/sample_code/main.py"	"function"	6	3	3
2	"/sample_code/main.py"	"import"	4	4	0
3	"/sample_code/models.py"	"class"	3	2	1
4	"/sample_code/models.py"	"function"	16	0	16
5	"/sample_code/models.py"	"import"	3	2	1
6	"/sample_code/services.py"	"class"	3	1	2
7	"/sample_code/services.py"	"function"	16	0	16
8	"/sample_code/services.py"	"import"	4	4	0
9	"/sample_code/utils.py"	"function"	8	1	7
10	"/sample_code/utils.py"	"import"	6	6	0

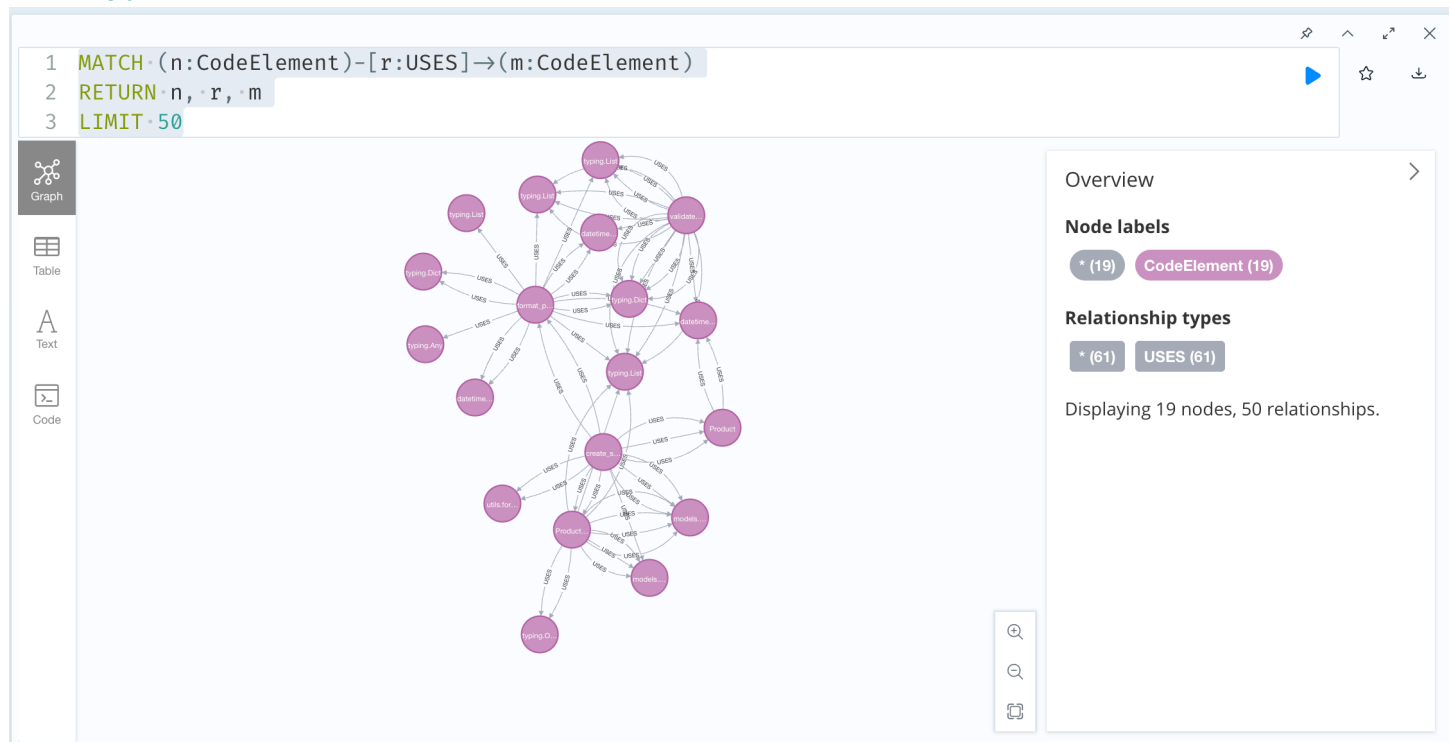
Started streaming 12 records after 21 ms and completed after 23 ms.

11. View Nodes WITH Relationships

MATCH (n:CodeElement)-[r:USES]->(m:CodeElement)

RETURN n, r, m

LIMIT 50



12. Find classes with no methods being used

MATCH (class:CodeElement {type: 'class'})

OPTIONAL MATCH (method:CodeElement {type: 'function'})




WHERE method.name **STARTS WITH** class.name + '.'

AND method.is_used = true

WITH class, **COUNT**(method) **as** used_methods

WHERE used_methods = 0

RETURN class.name, class.file_path, class.line_number;

neo4j\$ MATCH (class:CodeElement {type: 'class'}) OPTIONAL MATCH (method:CodeElement {type: 'fu...   

	class.name	class.file_path	class.line_number
1	"ProductService"	"/sample_code/services.py"	4
2	"CustomerService"	"/sample_code/services.py"	41
3	"DiscountService"	"/sample_code/services.py"	67
4	"Product"	"/sample_code/models.py"	4
5	"Customer"	"/sample_code/models.py"	40
6	"Order"	"/sample_code/models.py"	73

Started streaming 6 records after 22 ms and completed after 23 ms.