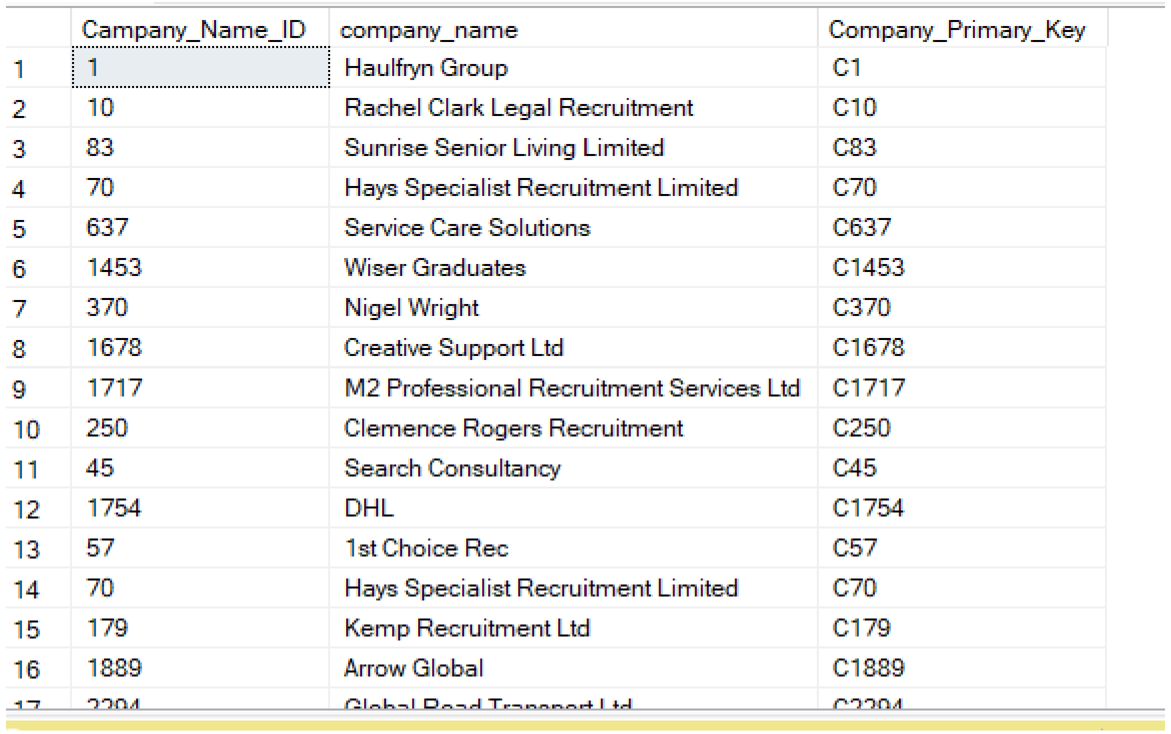
DATA WAREHOUSING-PROJECT 2

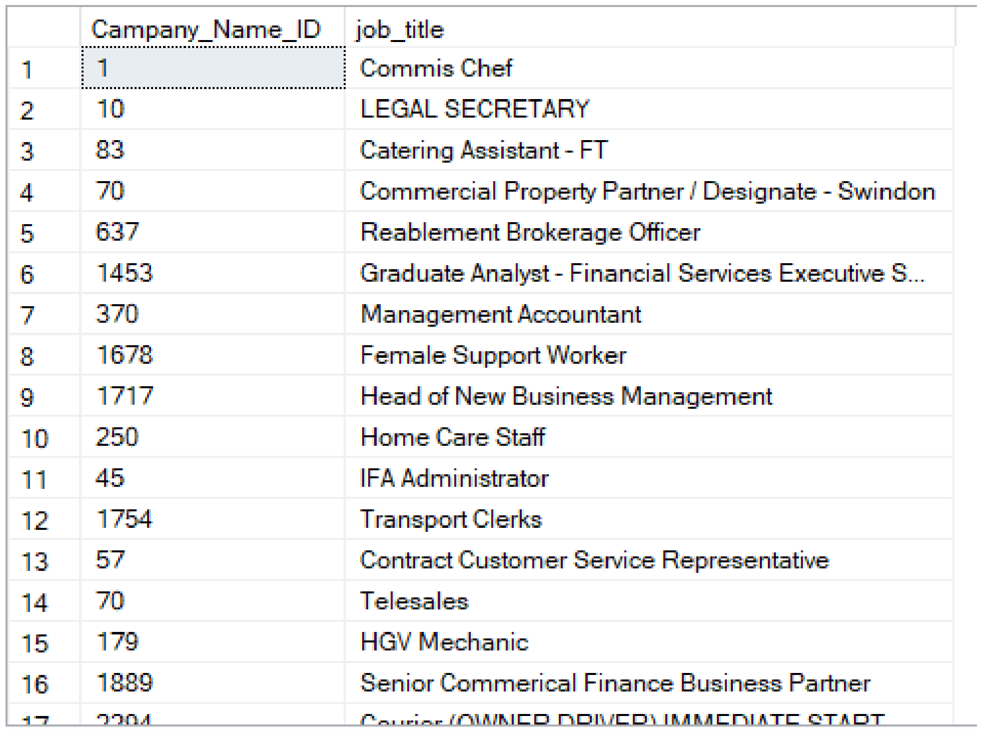
Ankit Kamboj
23372936

DATA MINING CASE AND NESTED TABLES



	Campany_Name_ID	company_name	Company_Primary_Key
1	1	Haulfryn Group	C1
2	10	Rachel Clark Legal Recruitment	C10
3	83	Sunrise Senior Living Limited	C83
4	70	Hays Specialist Recruitment Limited	C70
5	637	Service Care Solutions	C637
6	1453	Wiser Graduates	C1453
7	370	Nigel Wright	C370
8	1678	Creative Support Ltd	C1678
9	1717	M2 Professional Recruitment Services Ltd	C1717
10	250	Clemence Rogers Recruitment	C250
11	45	Search Consultancy	C45
12	1754	DHL	C1754
13	57	1st Choice Rec	C57
14	70	Hays Specialist Recruitment Limited	C70
15	179	Kemp Recruitment Ltd	C179
16	1889	Arrow Global	C1889
17	2294	Global Road Transport Ltd	C2294

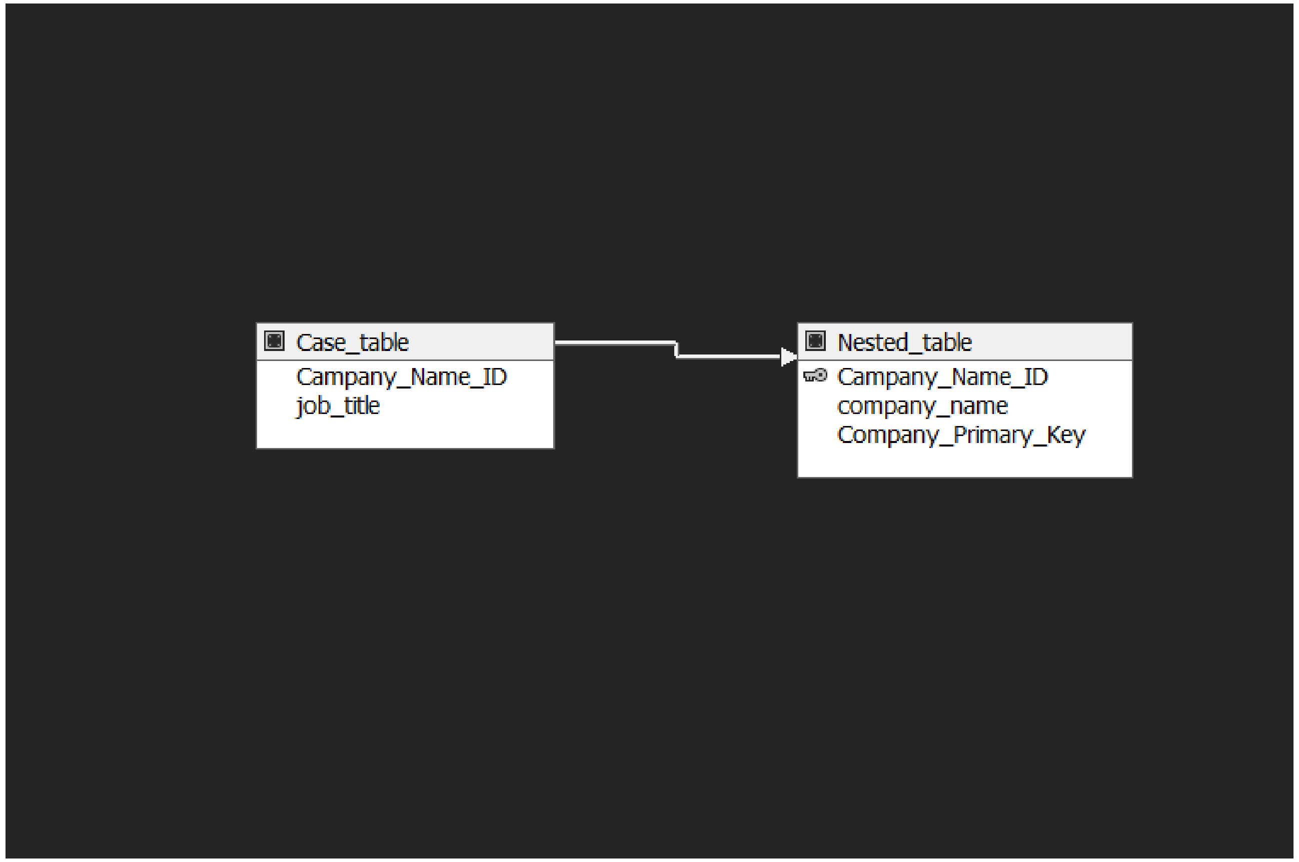
Nested Table



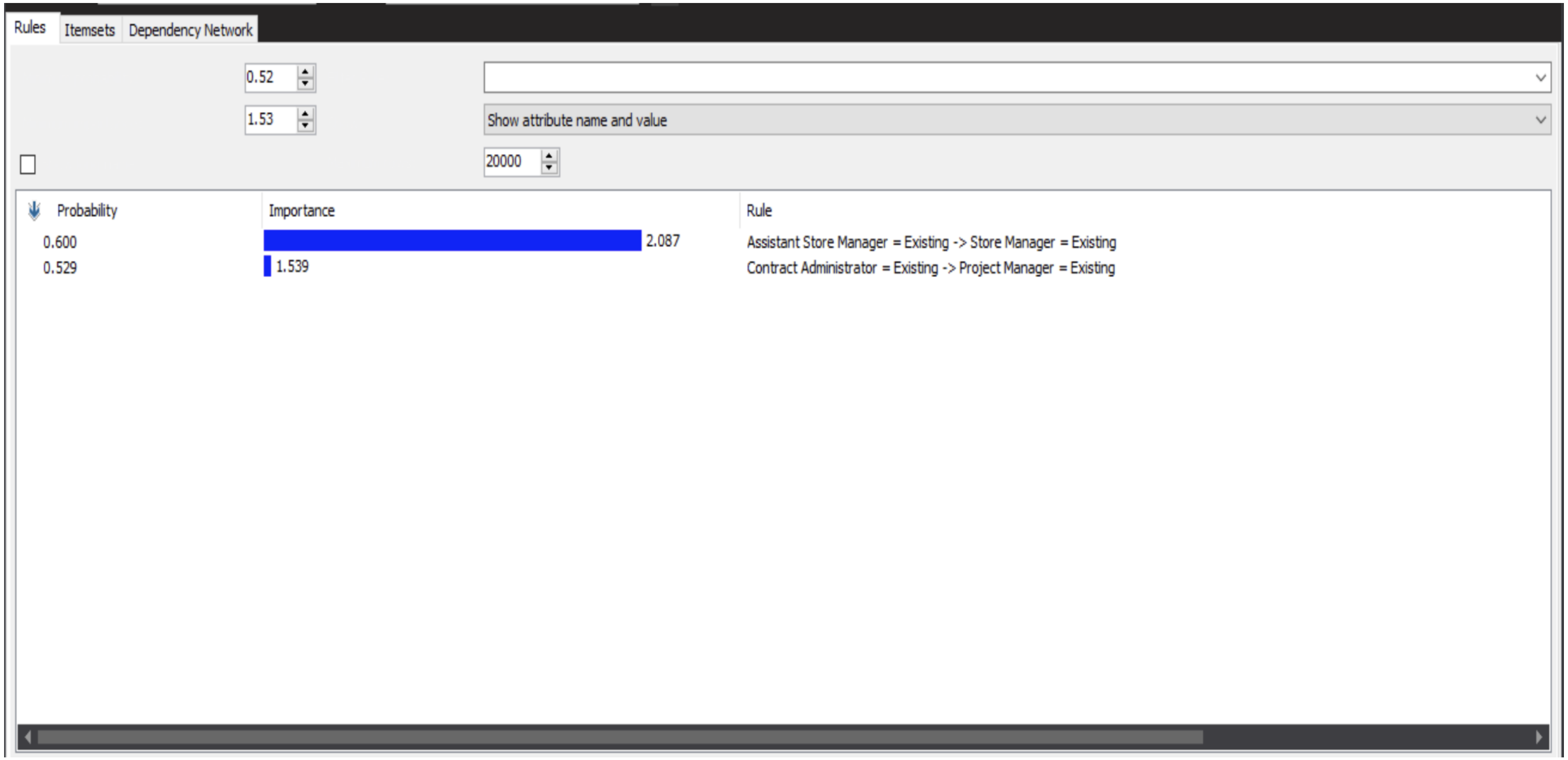
	Campany_Name_ID	job_title
1	1	Commis Chef
2	10	LEGAL SECRETARY
3	83	Catering Assistant - FT
4	70	Commercial Property Partner / Designate - Swindon
5	637	Reablement Brokerage Officer
6	1453	Graduate Analyst - Financial Services Executive S...
7	370	Management Accountant
8	1678	Female Support Worker
9	1717	Head of New Business Management
10	250	Home Care Staff
11	45	IFA Administrator
12	1754	Transport Clerks
13	57	Contract Customer Service Representative
14	70	Telesales
15	179	HGV Mechanic
16	1889	Senior Commerical Finance Business Partner
17	2294	Courier (OWNER DRIVER) IMMEDIATE START

Case Table

DATA SOURCE VIEWS



MINING MODEL VIEWER

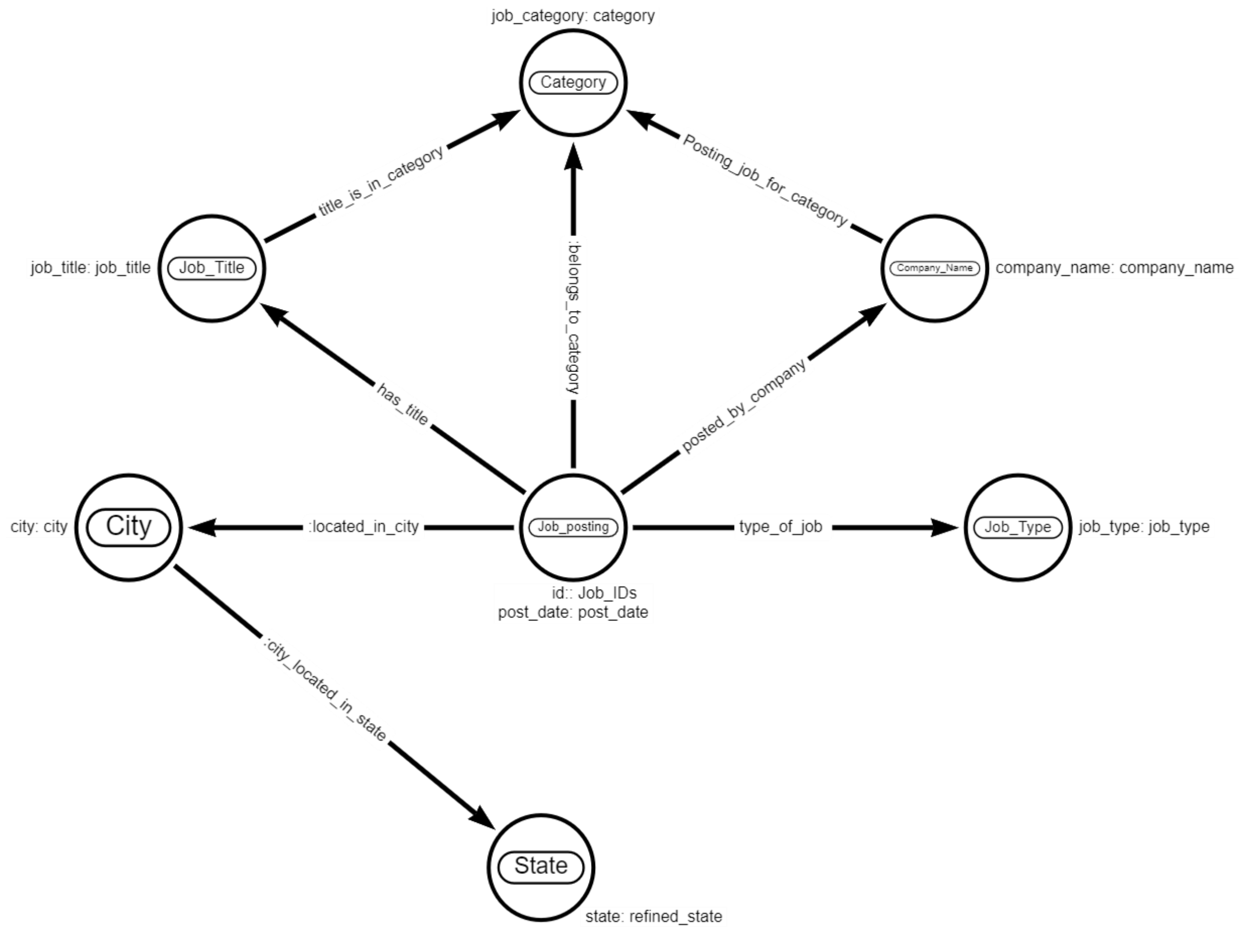


Probability	Importance	Rule
0.600	2.087	Assistant Store Manager = Existing -> Store Manager = Existing
0.529	1.539	Contract Administrator = Existing -> Project Manager = Existing

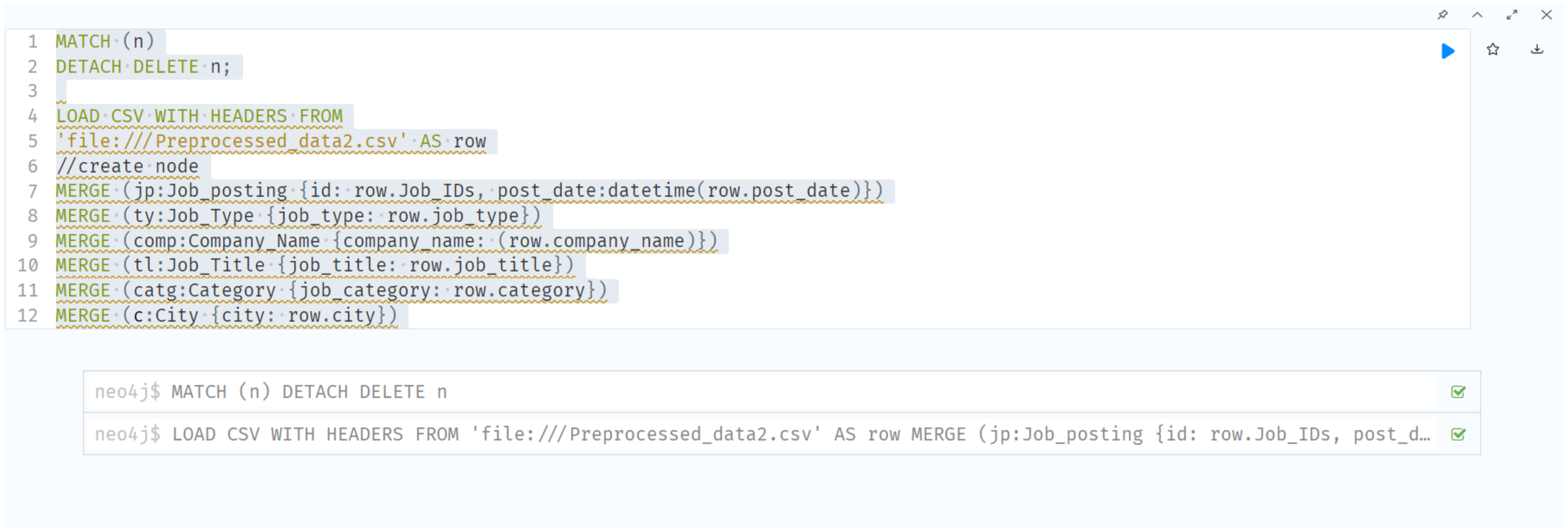
GRAPHICAL DATABASE – CYPHER FILE USED

```
MATCH (n)
DETACH DELETE n;

LOAD CSV WITH HEADERS FROM
'file:///Preprocessed_data2.csv' AS row
//create node
MERGE (jp:Job_posting {id: row.Job_IDs, post_date:datetime(row.post_date)})
MERGE (ty:Job_Type {job_type: row.job_type})
MERGE (comp:Company_Name {company_name: (row.company_name)})
MERGE (tl:Job_Title {job_title: row.job_title})
MERGE (catg:Category {job_category: row.category})
MERGE (c:City {city: row.city})
MERGE (s:State {state: row.refined_state})

// relations
MERGE (jp)-[:posted_by_company]->(comp)
MERGE (jp)-[:has_title]->(tl)
MERGE (jp)-[:belongs_to_category]->(catg)
MERGE (jp)-[:located_in_city]->(c)
MERGE (c)-[:city_located_in_state]->(s)
MERGE (comp)-[:Posting_job_for_category]->(catg)
MERGE (tl)-[:title_is_in_category]->(catg)
MERGE (jp)-[:type_of_job]->(ty);


// Q1 How many jobs are advertised for a given job category in a specified city?
WITH 'Information & Communication Technology' AS category, 'Melbourne' AS city
MATCH (catg:Category)<-[ :belongs_to_category ]-(jp:Job_posting)-[:located_in_city ]->(c:City)
WHERE  c.city = city AND catg.job_category = category
RETURN city, category, COUNT(jp.id) AS job_count;

// Q2 Find job_ids that share the same job_title.

MATCH (tl:Job_Title)<-[:has_title]-(jp:Job_posting)
WITH tl, collect(jp.id) AS job_IDs, COUNT(jp.id) AS Numer_of_jobs_under_title
WHERE size(job_IDs) > 1
RETURN tl.job_title AS job_title,Numer_of_jobs_under_title, job_IDs ;

// Q3 Find all companies that offer jobs in different categories.
```

DATABASE DESIGN



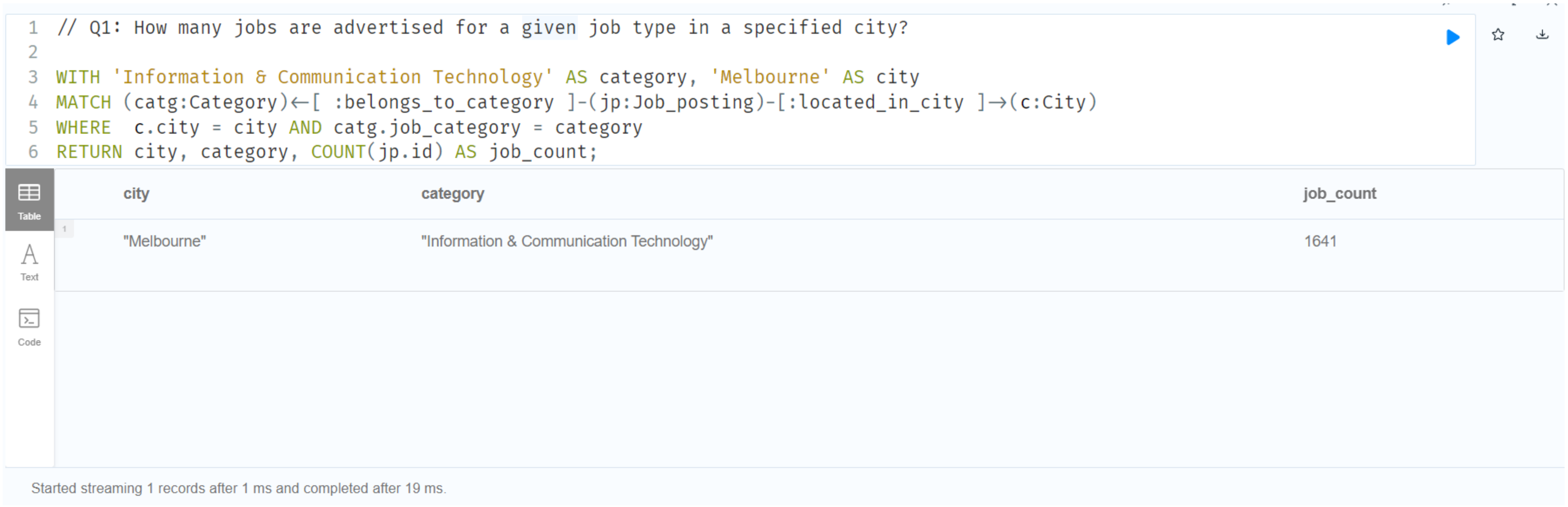
UPLOADING CSV AND CREATING NODES/RELATIONSHIPS



```
MATCH (n)
DETACH DELETE n;

LOAD CSV WITH HEADERS FROM
'file:///Preprocessed_data2.csv' AS row
//create node
MERGE (jp:Job_posting {id: row.Job_IDs, post_date:datetime(row.post_date)})
MERGE (ty:Job_Type {job_type: row.job_type})
MERGE (comp:Company_Name {company_name: (row.company_name)})
MERGE (tl:Job_Title {job_title: row.job_title})
MERGE (catg:Category {job_category: row.category})
MERGE (c:City {city: row.city})
```

```
neo4j$ MATCH (n) DETACH DELETE n
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///Preprocessed_data2.csv' AS row MERGE (jp:Job_posting {id: row.Job_IDs, post_d…
```

BUSINESS QUERIES SCRIPT AND OUTPUT

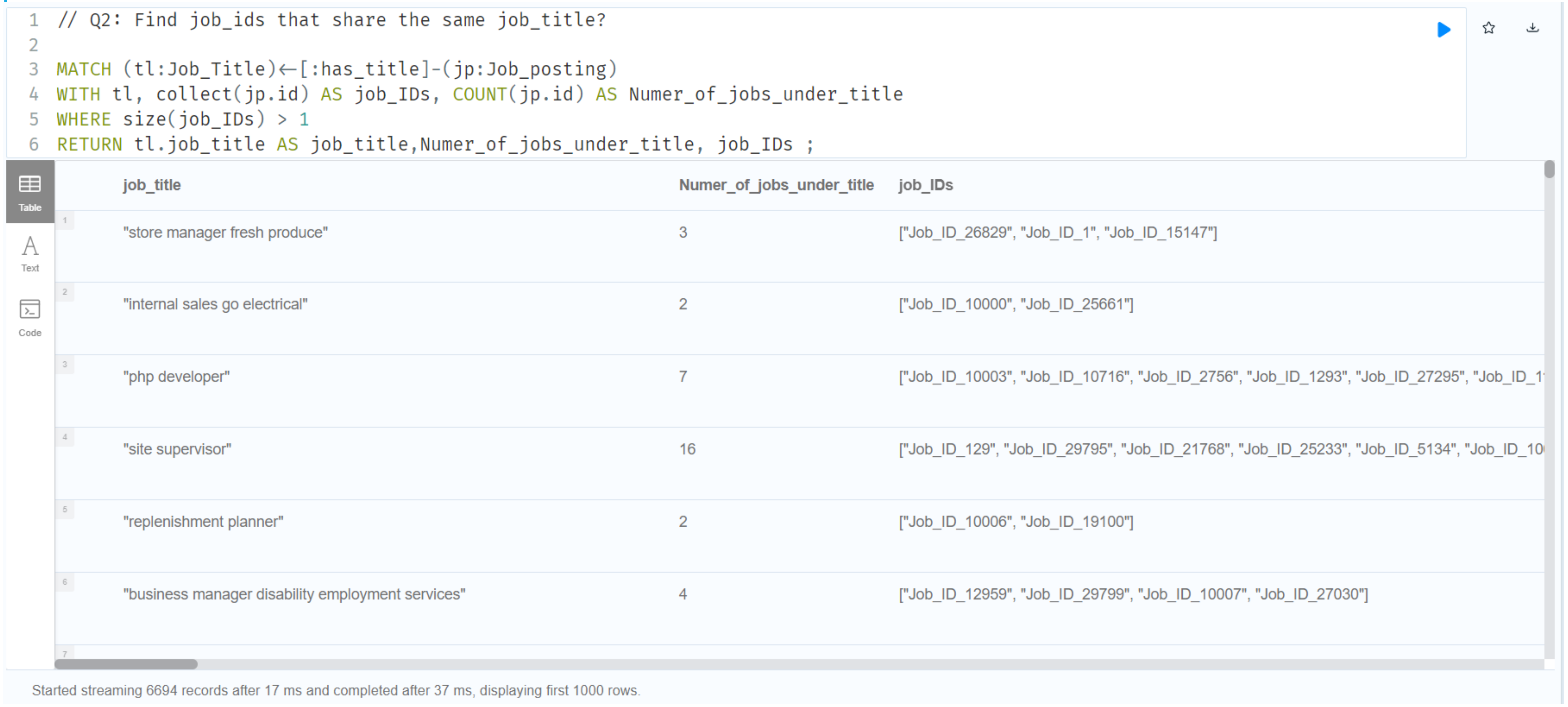


```
// Q1: How many jobs are advertised for a given job type in a specified city?

WITH 'Information & Communication Technology' AS category, 'Melbourne' AS city
MATCH (catg:Category)←[ :belongs_to_category ]-(jp:Job_posting)-[:located_in_city ]→(c:City)
WHERE  c.city = city AND catg.job_category = category
RETURN city, category, COUNT(jp.id) AS job_count;
```

	city	category	job_count
1	"Melbourne"	"Information & Communication Technology"	1641

Started streaming 1 records after 1 ms and completed after 19 ms.

BUSINESS QUERIES SCRIPT AND OUTPUT



```
// Q2: Find job_ids that share the same job_title?

MATCH (tl:Job_Title)←[:has_title]-(jp:Job_posting)
WITH tl, collect(jp.id) AS job_IDs, COUNT(jp.id) AS Numer_of_jobs_under_title
WHERE size(job_IDs) > 1
RETURN tl.job_title AS job_title,Numer_of_jobs_under_title, job_IDs ;
```

Table

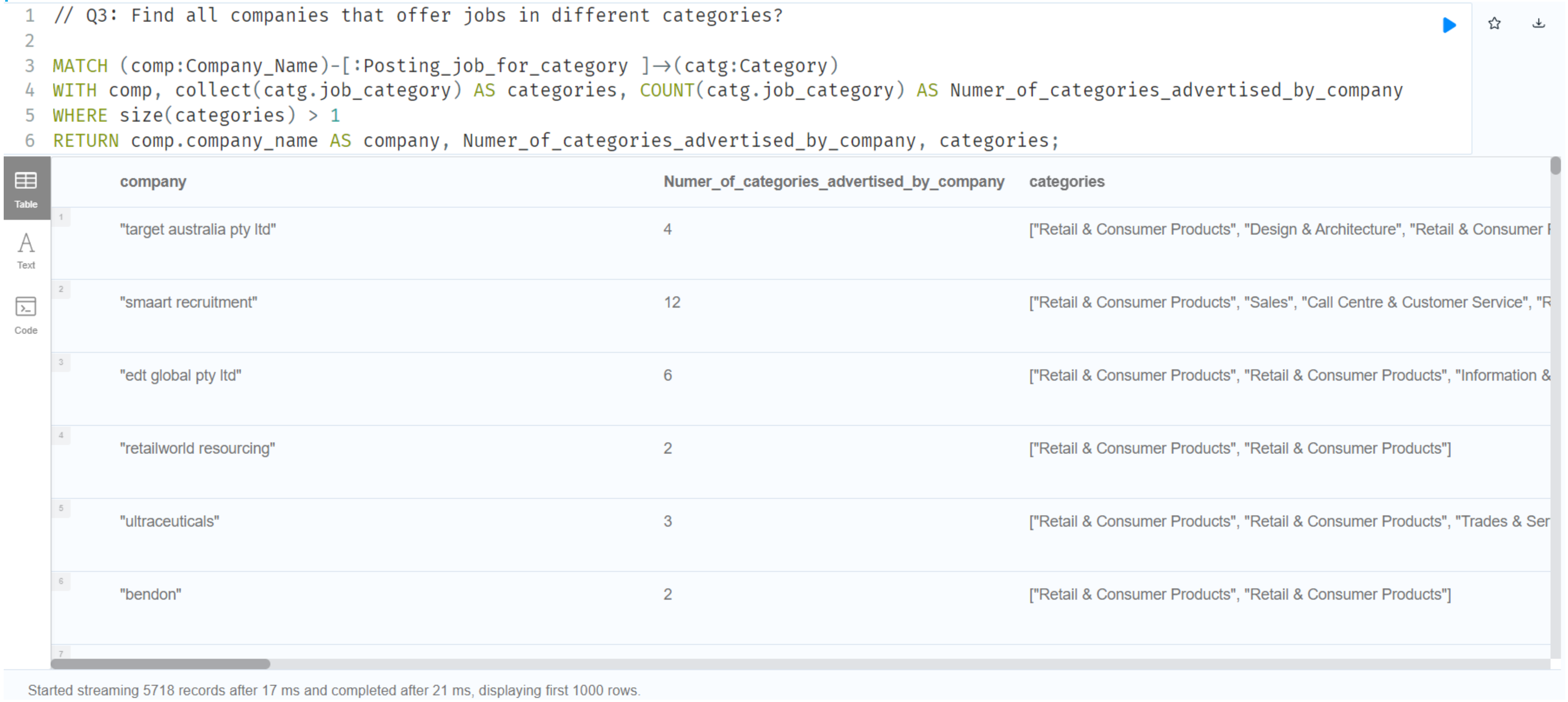
Text

Code

	job_title	Numer_of_jobs_under_title	job_IDs
1	"store manager fresh produce"	3	["Job_ID_26829", "Job_ID_1", "Job_ID_15147"]
2	"internal sales go electrical"	2	["Job_ID_10000", "Job_ID_25661"]
3	"php developer"	7	["Job_ID_10003", "Job_ID_10716", "Job_ID_2756", "Job_ID_1293", "Job_ID_27295", "Job_ID_1
4	"site supervisor"	16	["Job_ID_129", "Job_ID_29795", "Job_ID_21768", "Job_ID_25233", "Job_ID_5134", "Job_ID_10
5	"replenishment planner"	2	["Job_ID_10006", "Job_ID_19100"]
6	"business manager disability employment services"	4	["Job_ID_12959", "Job_ID_29799", "Job_ID_10007", "Job_ID_27030"]
7			

Started streaming 6694 records after 17 ms and completed after 37 ms, displaying first 1000 rows.

BUSINESS QUERIES SCRIPT AND OUTPUT

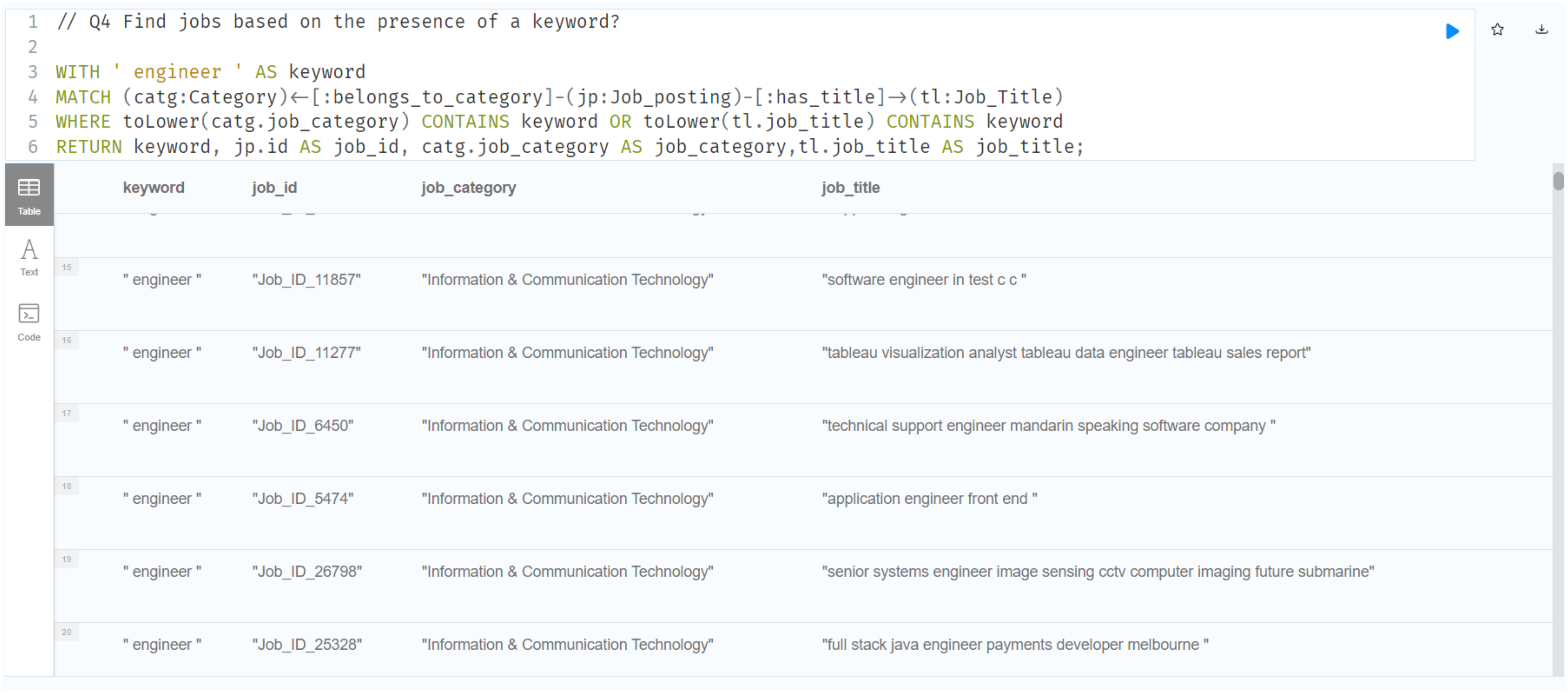


```
// Q3: Find all companies that offer jobs in different categories?

MATCH (comp:Company_Name)-[:Posting_job_for_category ]→(catg:Category)
WITH comp, collect(catg.job_category) AS categories, COUNT(catg.job_category) AS Numer_of_categories_advertised_by_company
WHERE size(categories) > 1
RETURN comp.company_name AS company, Numer_of_categories_advertised_by_company, categories;
```

	company	Numer_of_categories_advertised_by_company	categories
1	"target australia pty ltd"	4	["Retail & Consumer Products", "Design & Architecture", "Retail & Consumer
2	"smaart recruitment"	12	["Retail & Consumer Products", "Sales", "Call Centre & Customer Service", "R
3	"edt global pty ltd"	6	["Retail & Consumer Products", "Retail & Consumer Products", "Information &
4	"retailworld resourcing"	2	["Retail & Consumer Products", "Retail & Consumer Products"]
5	"ultraceuticals"	3	["Retail & Consumer Products", "Retail & Consumer Products", "Trades & Ser
6	"bendon"	2	["Retail & Consumer Products", "Retail & Consumer Products"]
7			

Started streaming 5718 records after 17 ms and completed after 21 ms, displaying first 1000 rows.

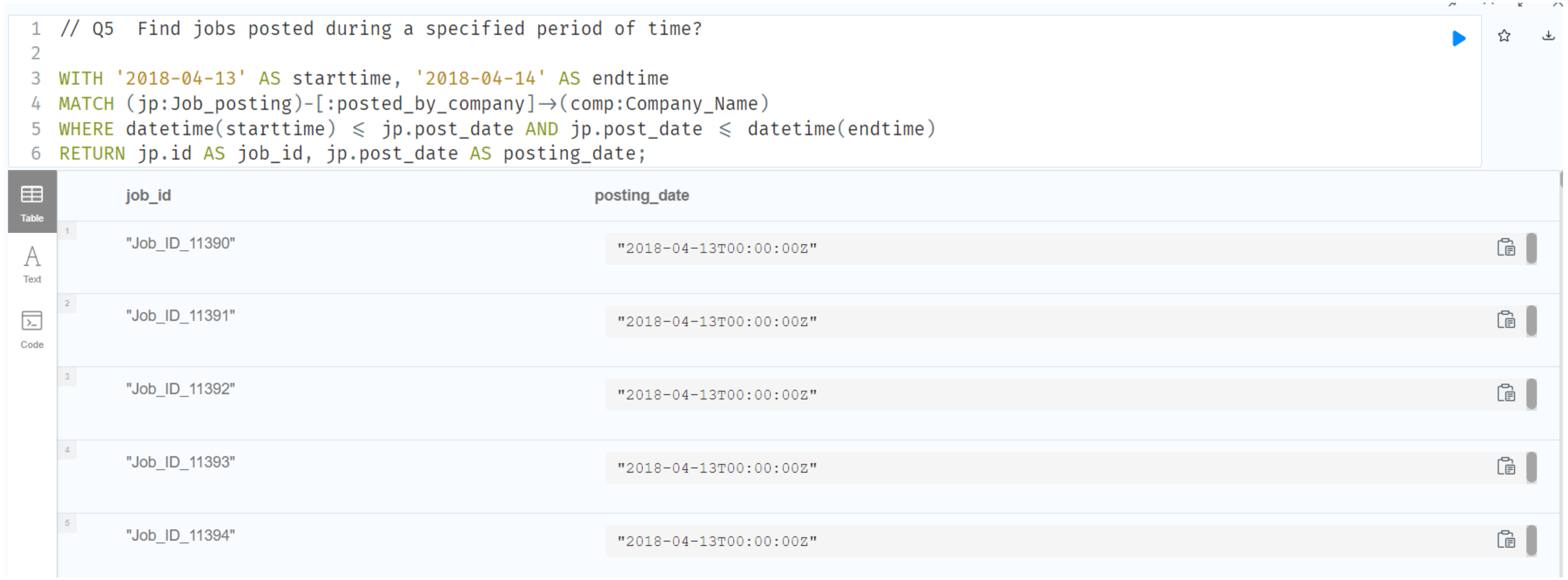
BUSINESS QUERIES SCRIPT AND OUTPUT



```
// Q4 Find jobs based on the presence of a keyword?

WITH ' engineer ' AS keyword
MATCH (catg:Category)←[:belongs_to_category]-(jp:Job_posting)-[:has_title]→(tl:Job_Title)
WHERE toLower(catg.job_category) CONTAINS keyword OR toLower(tl.job_title) CONTAINS keyword
RETURN keyword, jp.id AS job_id, catg.job_category AS job_category,tl.job_title AS job_title;
```

	keyword	job_id	job_category	job_title
15	" engineer "	"Job_ID_11857"	"Information & Communication Technology"	"software engineer in test c c "
16	" engineer "	"Job_ID_11277"	"Information & Communication Technology"	"tableau visualization analyst tableau data engineer tableau sales report"
17	" engineer "	"Job_ID_6450"	"Information & Communication Technology"	"technical support engineer mandarin speaking software company "
18	" engineer "	"Job_ID_5474"	"Information & Communication Technology"	"application engineer front end "
19	" engineer "	"Job_ID_26798"	"Information & Communication Technology"	"senior systems engineer image sensing cctv computer imaging future submarine"
20	" engineer "	"Job_ID_25328"	"Information & Communication Technology"	"full stack java engineer payments developer melbourne "

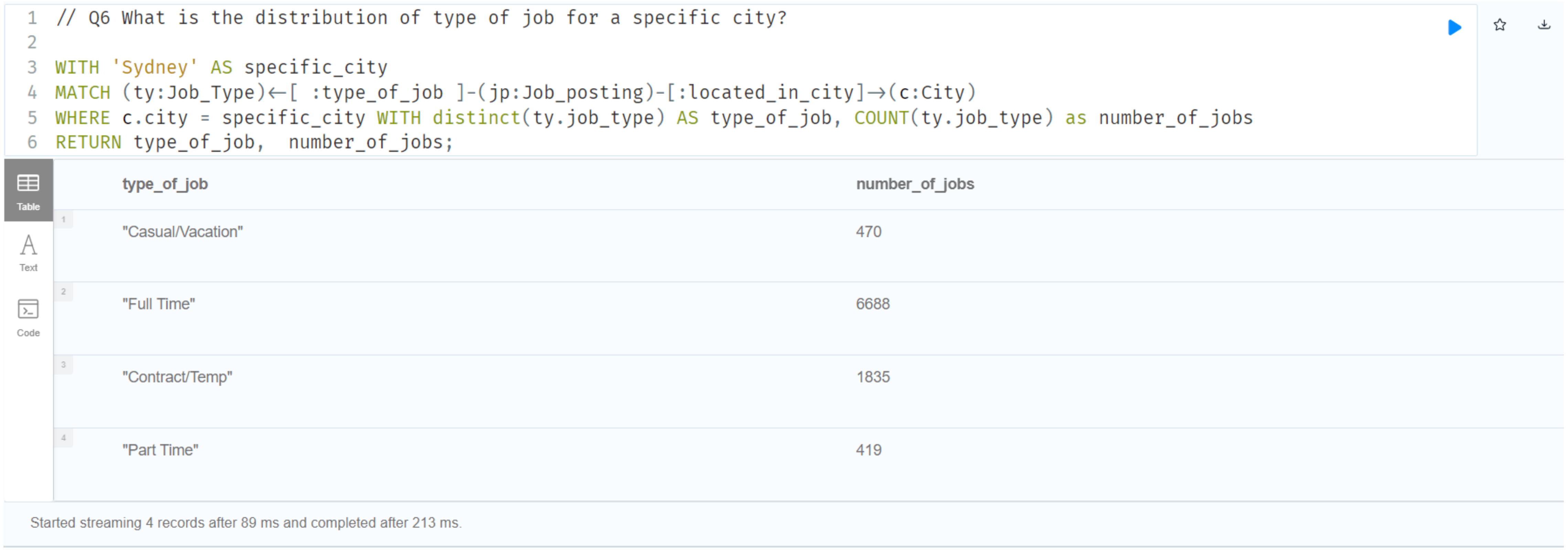
BUSINESS QUERIES SCRIPT AND OUTPUT



```
// Q5  Find jobs posted during a specified period of time?

WITH '2018-04-13' AS starttime, '2018-04-14' AS endtime
MATCH (jp:Job_posting)-[:posted_by_company]→(comp:Company_Name)
WHERE datetime(starttime) ≤ jp.post_date AND jp.post_date ≤ datetime(endtime)
RETURN jp.id AS job_id, jp.post_date AS posting_date;
```

	job_id	posting_date
1	"Job_ID_11390"	"2018-04-13T00:00:00Z"
2	"Job_ID_11391"	"2018-04-13T00:00:00Z"
3	"Job_ID_11392"	"2018-04-13T00:00:00Z"
4	"Job_ID_11393"	"2018-04-13T00:00:00Z"
5	"Job_ID_11394"	"2018-04-13T00:00:00Z"

BUSINESS QUERIES SCRIPT AND OUTPUT

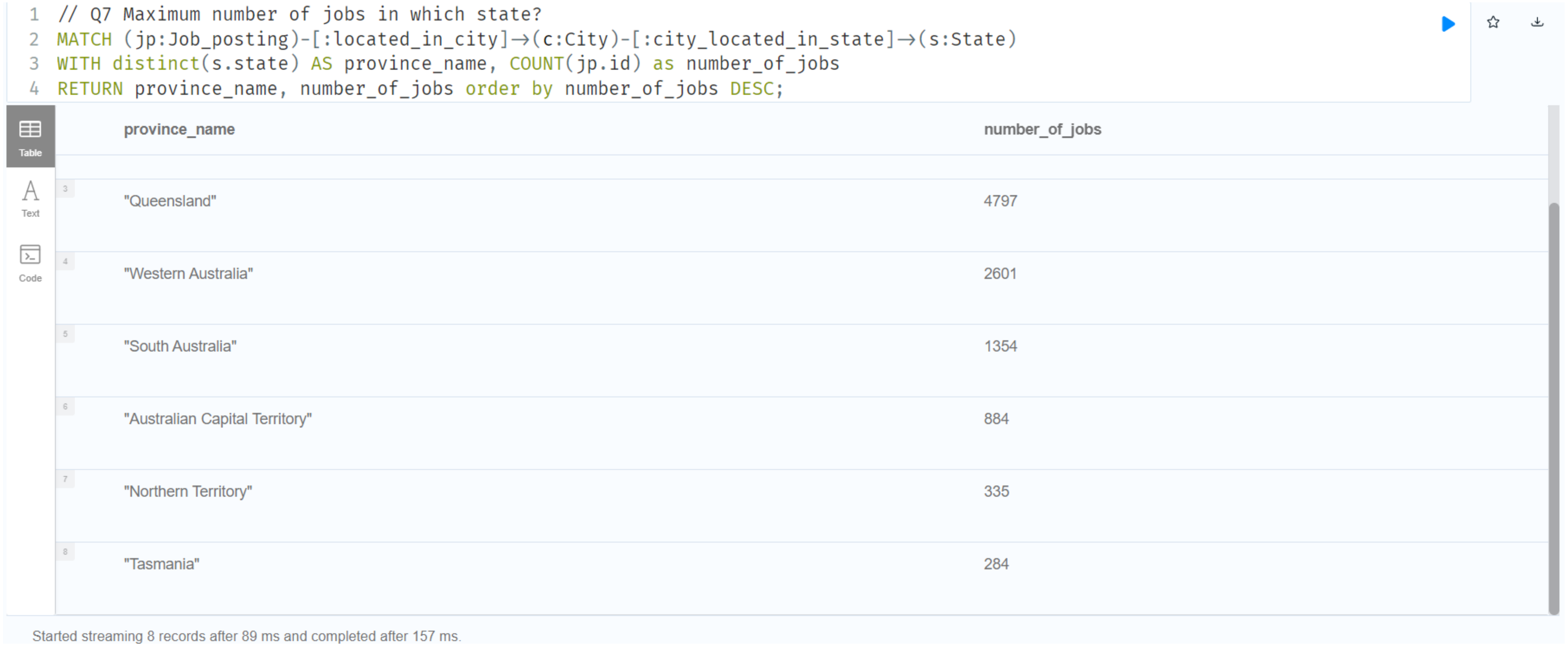


```
// Q6 What is the distribution of type of job for a specific city?

WITH 'Sydney' AS specific_city
MATCH (ty:Job_Type)←[ :type_of_job ]-(jp:Job_posting)-[:located_in_city]→(c:City)
WHERE c.city = specific_city WITH distinct(ty.job_type) AS type_of_job, COUNT(ty.job_type) as number_of_jobs
RETURN type_of_job,  number_of_jobs;
```

	type_of_job	number_of_jobs
1	"Casual/Vacation"	470
2	"Full Time"	6688
3	"Contract/Temp"	1835
4	"Part Time"	419

Started streaming 4 records after 89 ms and completed after 213 ms.

BUSINESS QUERIES SCRIPT AND OUTPUT



```
// Q7 Maximum number of jobs in which state?
MATCH (jp:Job_posting)-[:located_in_city]→(c:City)-[:city_located_in_state]→(s:State)
WITH distinct(s.state) AS province_name, COUNT(jp.id) as number_of_jobs
RETURN province_name, number_of_jobs order by number_of_jobs DESC;
```

Table

Text

Code

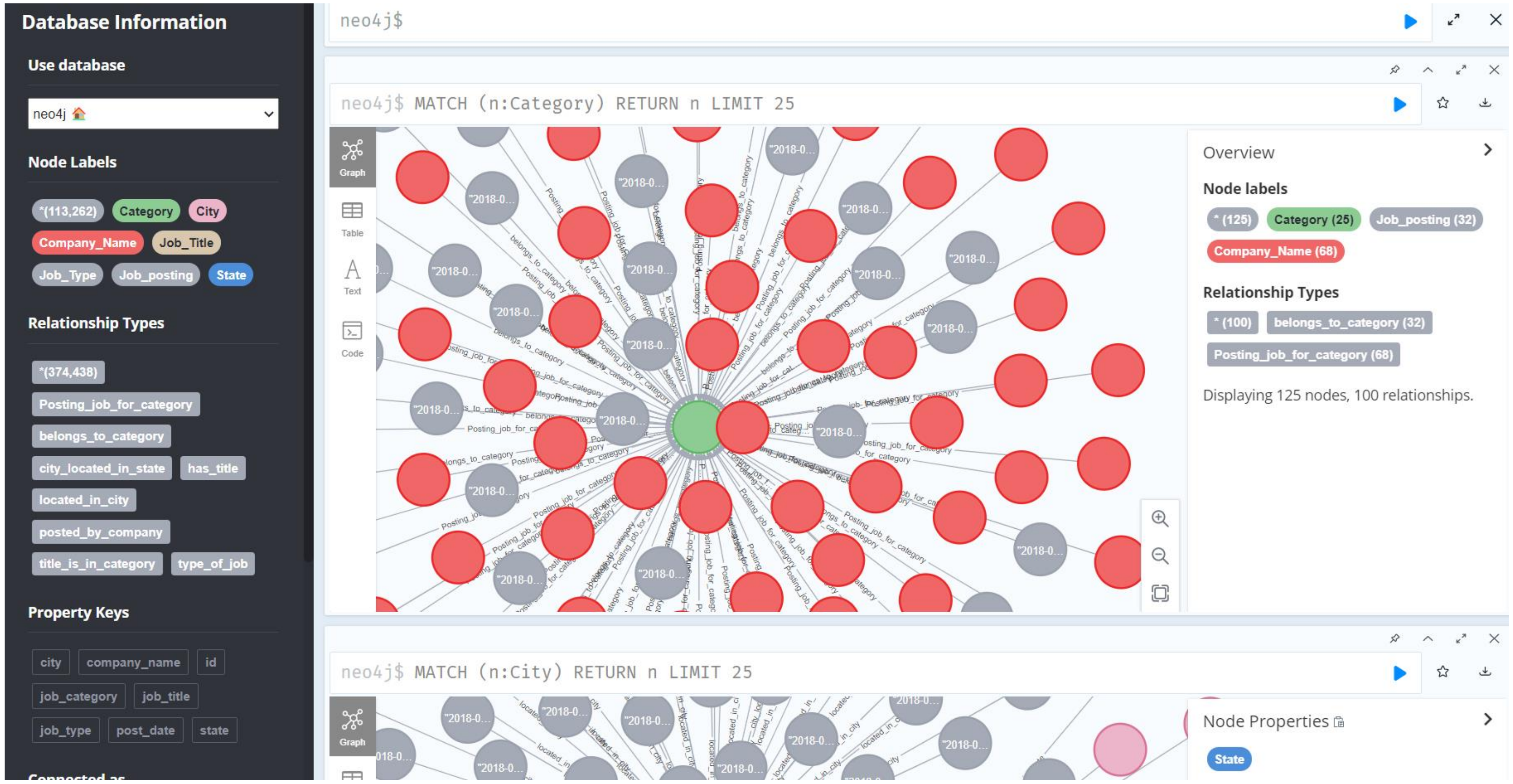
	province_name	number_of_jobs
3	"Queensland"	4797
4	"Western Australia"	2601
5	"South Australia"	1354
6	"Australian Capital Territory"	884
7	"Northern Territory"	335
8	"Tasmania"	284

Started streaming 8 records after 89 ms and completed after 157 ms.

DESIGN CHOICES WITH PROS AND CONS IDENTIFIED

- The Nodes have been created to address the specific business queries that need to be addressed.
- For example, we want to find the relationship between the category and job titles, therefore the relationship has been created between the two.
- Similarly, for category and company name.
- Cons for it is that it takes additional computational time and space to create the graphs at initial stage but it is very easy to solve the business inquiries that uses these nodes as input along with the relationship between them

MEANINGFUL GRAPH DATABASE NAVIGATION DISCUSSED



Database Information

Use database

neo4j

Node Labels

*(113,262) Category City Company_Name Job_Title Job_Type Job_posting State

Relationship Types

*(374,438) Posting_job_for_category belongs_to_category city_located_in_state has_title located_in_city posted_by_company title_is_in_category type_of_job

Property Keys

city company_name id job_category job_title job_type post_date state

```
neo4j$
```

```
neo4j$ MATCH (n:Category) RETURN n LIMIT 25
```

Overview

Node labels

* (125) Category (25) Job_posting (32) Company_Name (68)

Relationship Types

* (100) belongs_to_category (32) Posting_job_for_category (68)

Displaying 125 nodes, 100 relationships.

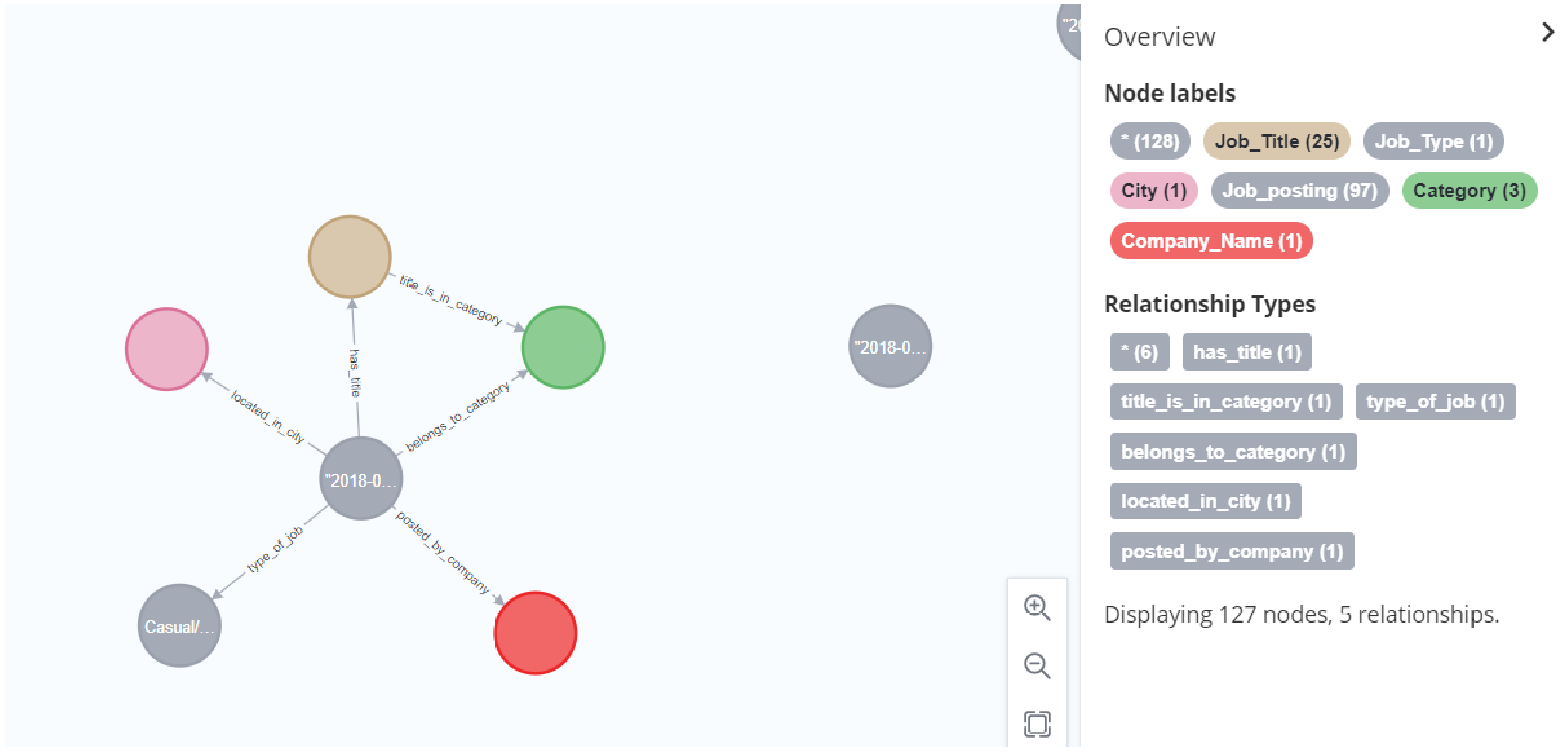
```
neo4j$ MATCH (n:City) RETURN n LIMIT 25
```

Node Properties

State

Its easy to navigate between the nodes after running Inquiry and using graph interface

MEANINGFUL GRAPH DATABASE NAVIGATION DISCUSSED



Displaying 127 nodes, 5 relationships.

Finding characteristics of certain job advertisement and relationships