DSC 202 Final Project

DS Job Employment

Group members:
Shuying Li,
Jiaxian Xiang,
Shanglin Zeng

Project Overviews

This project's objective is to establish a Knowledge Graph tailored for individuals seeking employment, enabling them to locate jobs based on factors like sponsorship, salary ranges, company details, and more. It also provides a means for users to access pertinent information within specific companies, salary brackets, and other essential criteria.

Motivation:

To bridge the gap between job seekers and their ideal employment opportunities while facilitating a deeper understanding of the factors that matter most in their job search.

Datasets Introduction

- Data Science Job Posting on Glassdoor [link]
- Global AI, ML, Data Science Salary 2023 [Salaries]

- Sponsorship company lists (web scraping from https://h1binfo.org/)

Datasets Introduction #1

Data Science Job Posting on Glassdoor [link]

- **job_simp:** Job type
- **Salary Estimation:** Salary range for that particular job
- **Rating:** Rating of that post
- **Company:** Name of company
- **Location:** Location of the company
- **Headquarter:** Location of the headquarter
- **Size:** Total employee in that company
- **Type of ownership:** Describes the company type i.e non-profit/public/private farm etc
- **Industry, Sector:** Field applicant will work in
- **min_salary,max_salary,avg_salary:** Refers to the minimum, maximum and average salary for that post
- **job_state:** State where the applicant will work
- **same_state:** Same state as headquarter or not(Boolean)
- **python,excel,hadoop,spark,aws,tableau,big_data:** Some most appeared skills in boolean columns form

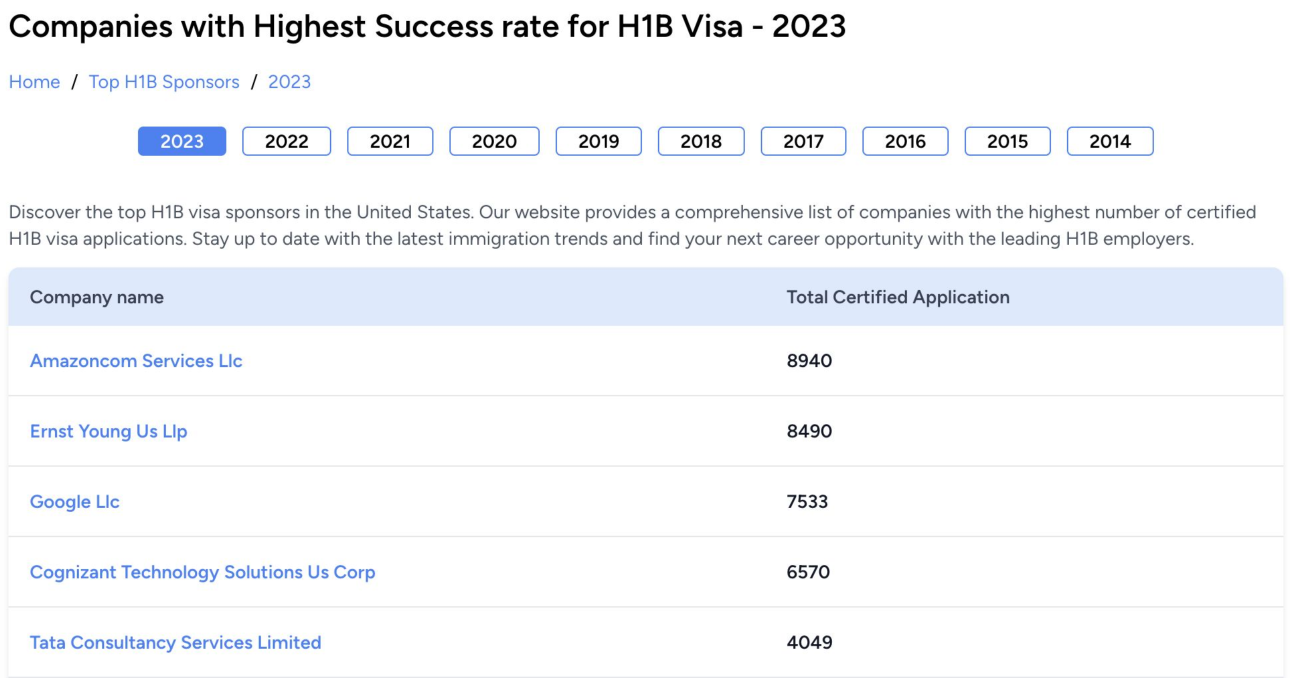
Datasets Introduction #2

Global AI, ML, Data Science Salary 2023 [Salaries]

- **Work_year:** The year the salary was paid
- **Experience_level:** The experience level in the job during the year with the following possible values
- **Employment_type:** The type of employment for the role
- **Job_title:** The role worked in during the year
- **Salary:** The total gross salary amount paid
- **Salary_currency:** The currency of the salary paid as an ISO 4217 currency code
- **Salary_in_usd:** The salary in USD (FX rate divided by avg. USD rate for the respective year via fxdata.foorilla.com)
- **Employee_residence:** Employee's primary country of residence in during the work year as an ISO 3166 country code
- **Remote_ratio:** The overall amount of work done remotely, possible values are as follows
- **Company_location:** The country of the employer's main office or contracting branch as an ISO 3166 country code
- **Company_size:** The average number of people that worked for the company during the year

Datasets Introduction #3

Sponsorship company lists (web scraping from https://h1binfo.org/)



Companies with Highest Success rate for H1B Visa - 2023

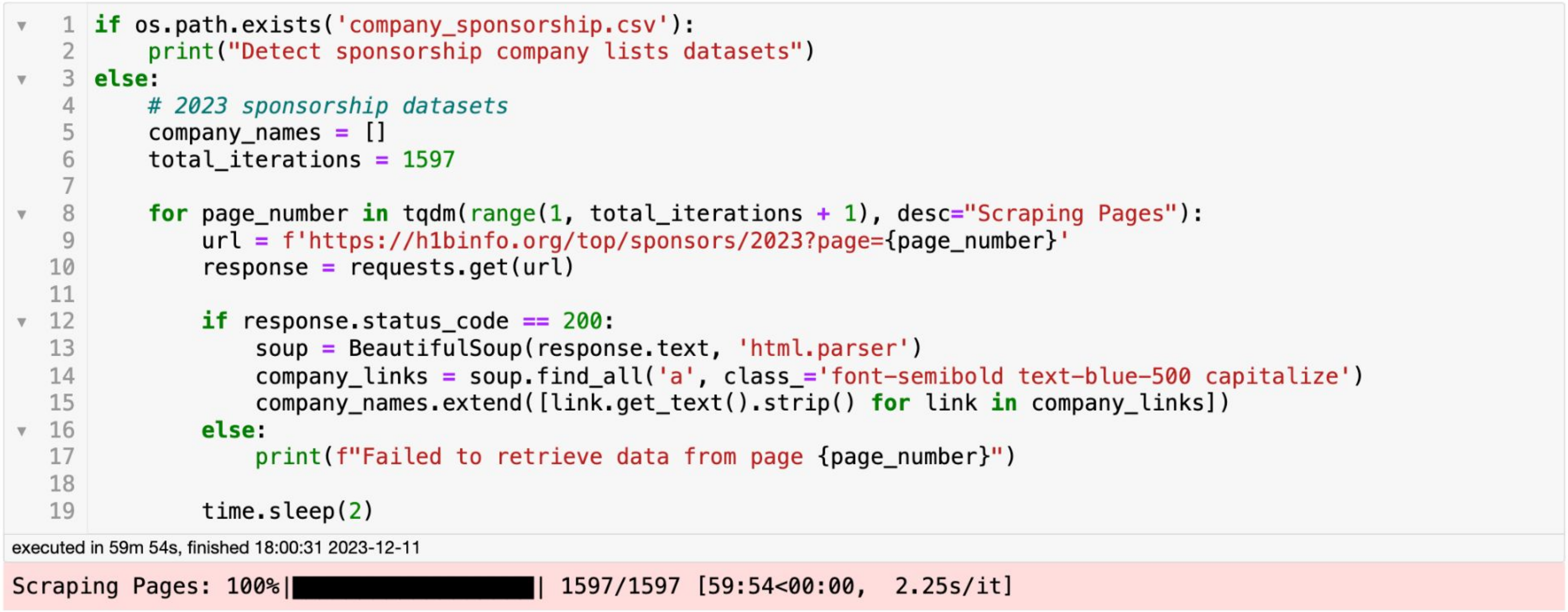
Home / Top H1B Sponsors / 2023

2023 2022 2021 2020 2019 2018 2017 2016 2015 2014

Discover the top H1B visa sponsors in the United States. Our website provides a comprehensive list of companies with the highest number of certified H1B visa applications. Stay up to date with the latest immigration trends and find your next career opportunity with the leading H1B employers.

Company name	Total Certified Application
Amazoncom Services Llc	8940
Ernst Young Us Llp	8490
Google Llc	7533
Cognizant Technology Solutions Us Corp	6570
Tata Consultancy Services Limited	4049

Web Scraping



```python
if os.path.exists('company_sponsorship.csv'):
    print("Detect sponsorship company lists datasets")
else:
    # 2023 sponsorship datasets
    company_names = []
    total_iterations = 1597

    for page_number in tqdm(range(1, total_iterations + 1), desc="Scraping Pages"):
        url = f'https://h1binfo.org/top/sponsors/2023?page={page_number}'
        response = requests.get(url)

        if response.status_code == 200:
            soup = BeautifulSoup(response.text, 'html.parser')
            company_links = soup.find_all('a', class_='font-semibold text-blue-500 capitalize')
            company_names.extend([link.get_text().strip() for link in company_links])
        else:
            print(f"Failed to retrieve data from page {page_number}")

        time.sleep(2)
```

executed in 59m 54s, finished 18:00:31 2023-12-11

```
Scraping Pages: 100%|██████████| 1597/1597 [59:54<00:00,  2.25s/it]
```

Question #1 and Query #1

Question:

If an employee with entry-level experience want to work in US now, what are the jobs and corresponding salaries he could choose from?

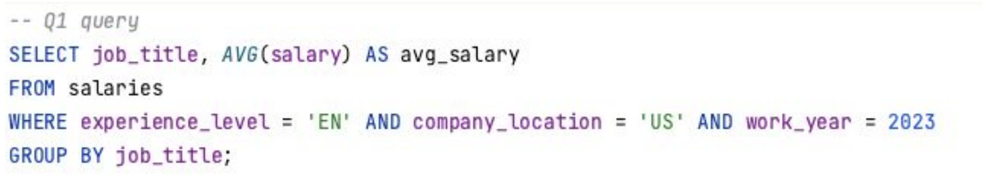
Methodology:

Using PostgreSQL to retrieve the average salary (AVG(salary)) for each unique job title from the "salaries" table. It specifically focuses on entries where the experience level is entry level and the company location is in 'US'. The results are grouped by the job title, providing a summary of average salaries for each distinct job title within the specified criteria.

Purpose - Why we choose PostgreSQL:

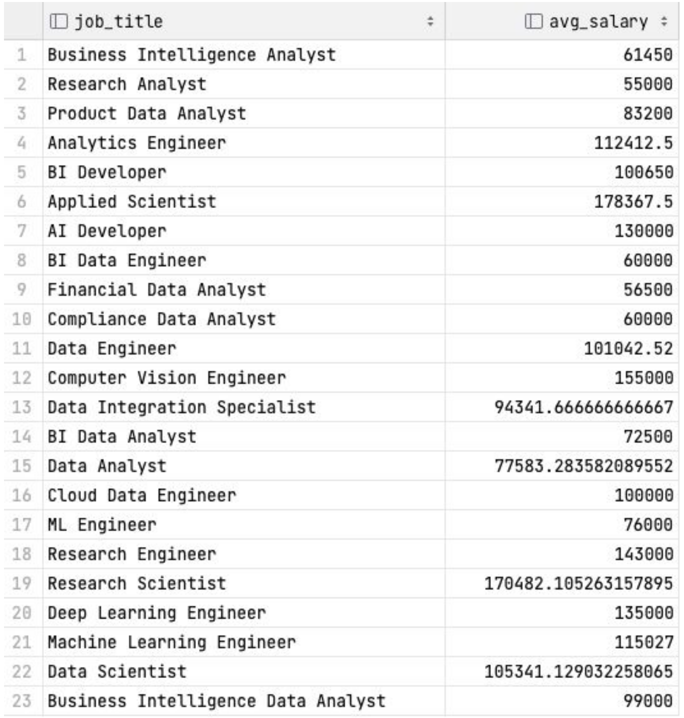
Easy to write query, do not need to design data structure in neo4j.

Question #1 Query AND Results



```sql
-- Q1 query
SELECT job_title, AVG(salary) AS avg_salary
FROM salaries
WHERE experience_level = 'EN' AND company_location = 'US' AND work_year = 2023
GROUP BY job_title;
```

Dataset#2



	job_title	avg_salary
1	Business Intelligence Analyst	61450
2	Research Analyst	55000
3	Product Data Analyst	83200
4	Analytics Engineer	112412.5
5	BI Developer	100650
6	Applied Scientist	178367.5
7	AI Developer	130000
8	BI Data Engineer	60000
9	Financial Data Analyst	56500
10	Compliance Data Analyst	60000
11	Data Engineer	101042.52
12	Computer Vision Engineer	155000
13	Data Integration Specialist	94341.66666666667
14	BI Data Analyst	72500
15	Data Analyst	77583.28358208952
16	Cloud Data Engineer	100000
17	ML Engineer	76000
18	Research Engineer	143000
19	Research Scientist	170482.10526315789
20	Deep Learning Engineer	135000
21	Machine Learning Engineer	115027
22	Data Scientist	105341.12903225806
23	Business Intelligence Data Analyst	99000

Question #2 and Query #2

Question:

For the companies who are actively hiring data scientist, list the top ten employers together with their company age who offer the highest salaries on average.

Methodology:

We use SQL to query the "Cleaned_DS_Job" table, selecting columns company_name, company_age, average_salary, and job_title. The query filters the results to include only entries where the lowercase version of the job_title is 'data scientist'. Subsequently, the results are ordered in descending order based on the average_salary column, and the output is limited to the top 10 records.

Purpose – Why we choose PostgreSQL:

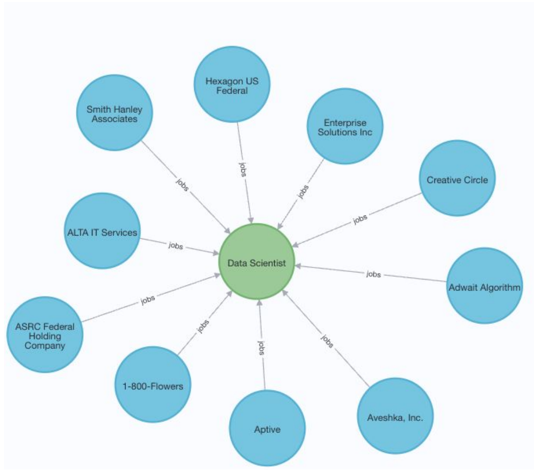
Easy to write query, do not need to design data structure in neo4j.

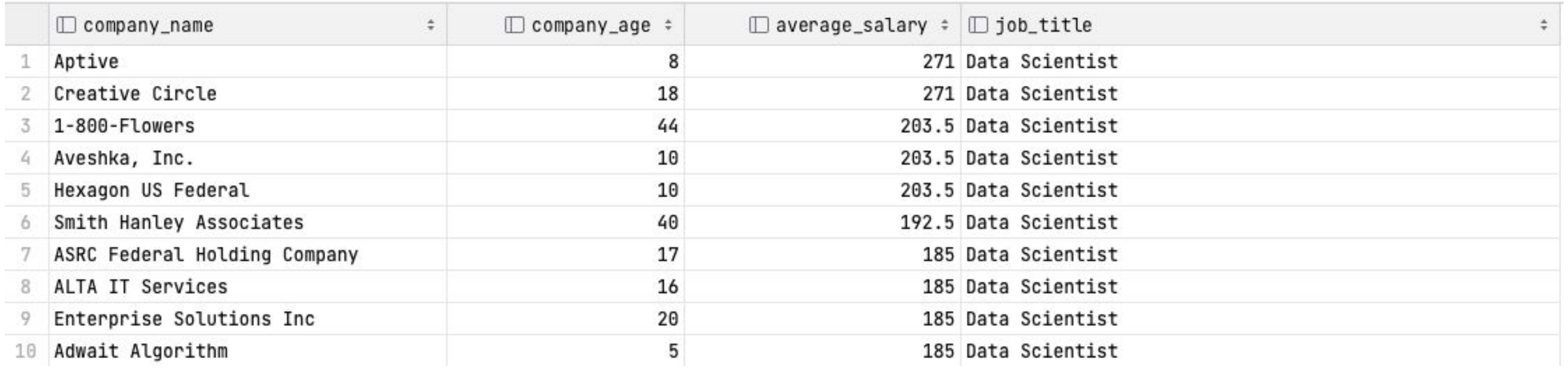
Question #2 and Query #2

Query:

```
SELECT company_name, company_age, AVG(avg_salary) as avg_salary, job_title
FROM "original_cleaned_ds_jobs"
WHERE job_title ILIKE '%Data Scientist%' AND company_age != -1
GROUP BY company_name, job_title, company_age
ORDER BY avg_salary DESC
LIMIT 10;
```

Question #2 Results AND Data Visualization





	company_name	company_age	average_salary	job_title
1	Aptive	8	271	Data Scientist
2	Creative Circle	18	271	Data Scientist
3	1-800-Flowers	44	203.5	Data Scientist
4	Aveshka, Inc.	10	203.5	Data Scientist
5	Hexagon US Federal	10	203.5	Data Scientist
6	Smith Hanley Associates	40	192.5	Data Scientist
7	ASRC Federal Holding Company	17	185	Data Scientist
8	ALTA IT Services	16	185	Data Scientist
9	Enterprise Solutions Inc	20	185	Data Scientist
10	Adwait Algorithm	5	185	Data Scientist

Question #3 and Query #3

Question:

What are the companies with a rating larger than 4 that have job openings located at their headquarters and provide sponsorship?

Methods:

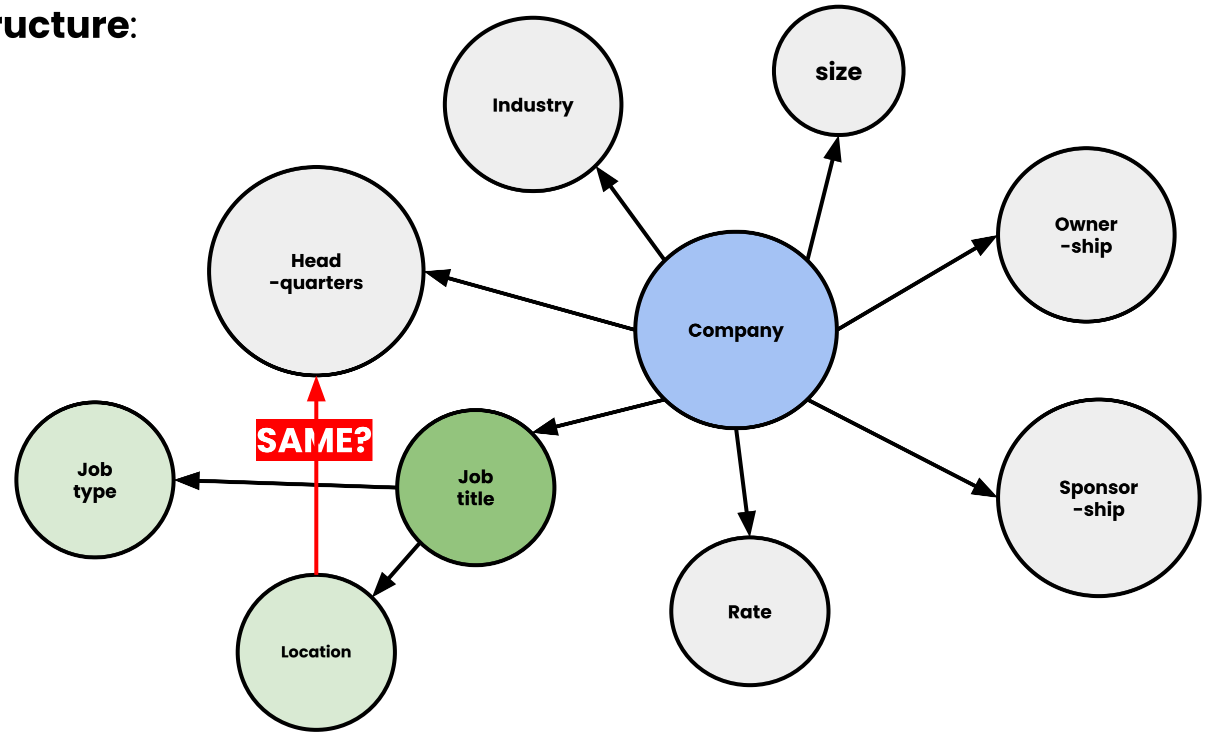
A data structure was developed to represent the relationships between companies, their job openings, headquarters locations, reviews, and sponsorship statuses.

Purpose – Why we choose neo4j:

Our objective is to create an insightful visualization of the connections between companies. By utilizing Neo4j, we can readily construct and navigate the complex network of company relationships.

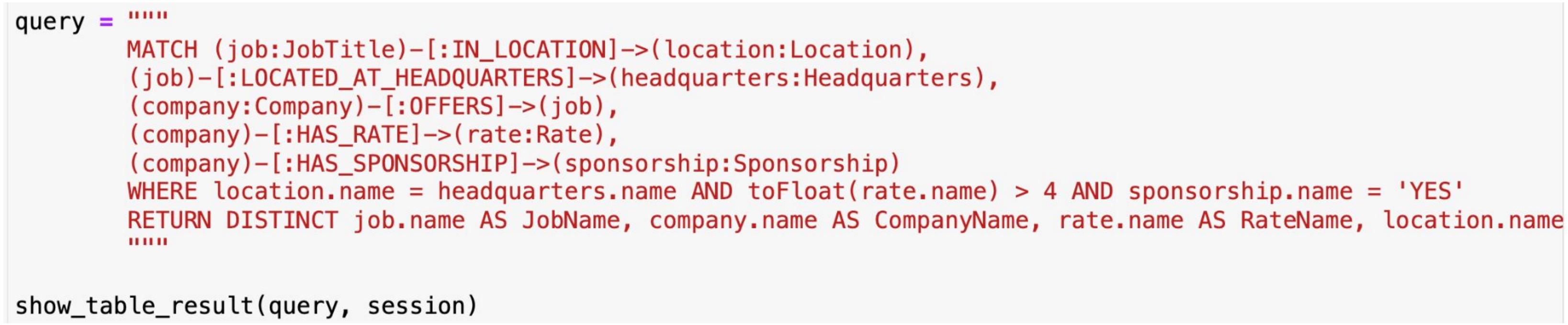
Question #3 and Query #3

Basic Structure:



Question #3 and Query #3

Query:



```
query = """
        MATCH (job:JobTitle)-[:IN_LOCATION]->(location:Location),
        (job)-[:LOCATED_AT_HEADQUARTERS]->(headquarters:Headquarters),
        (company:Company)-[:OFFERS]->(job),
        (company)-[:HAS_RATE]->(rate:Rate),
        (company)-[:HAS_SPONSORSHIP]->(sponsorship:Sponsorship)
        WHERE location.name = headquarters.name AND toFloat(rate.name) > 4 AND sponsorship.name = 'YES'
        RETURN DISTINCT job.name AS JobName, company.name AS CompanyName, rate.name AS RateName, location.name
        """

show_table_result(query, session)
```

Question #4 and Query #4

Question:

Retrieve the names of companies where the average salary is larger than 150 and the job requires AWS.

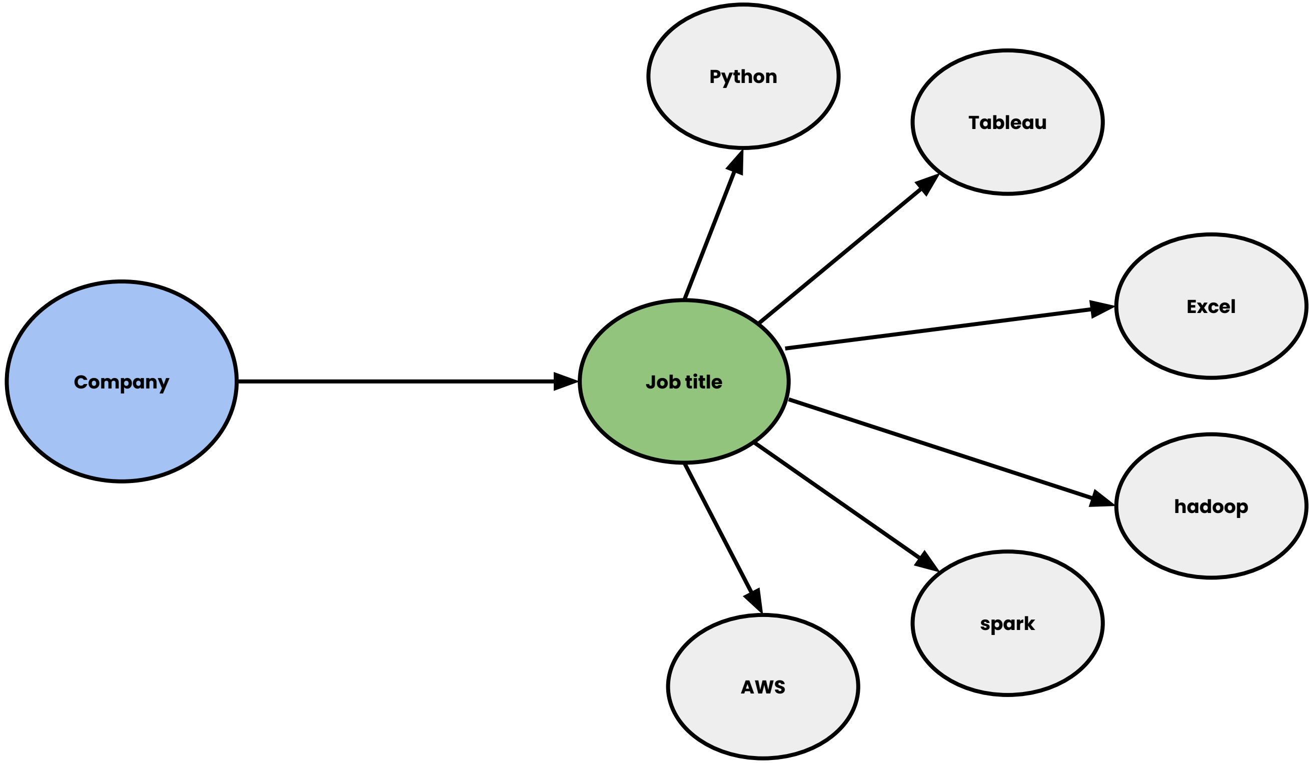
Methods:

Identifying a sequence where a Company node offers (OFFERS relationship) a Job node, and this job in turn requires (REQUIRES relationship) a specific Skill node with the name 'AWS'. The WHERE clause filters these jobs to only include those with an average_salary greater than 150.

Purpose - Why we choose Neo4j:

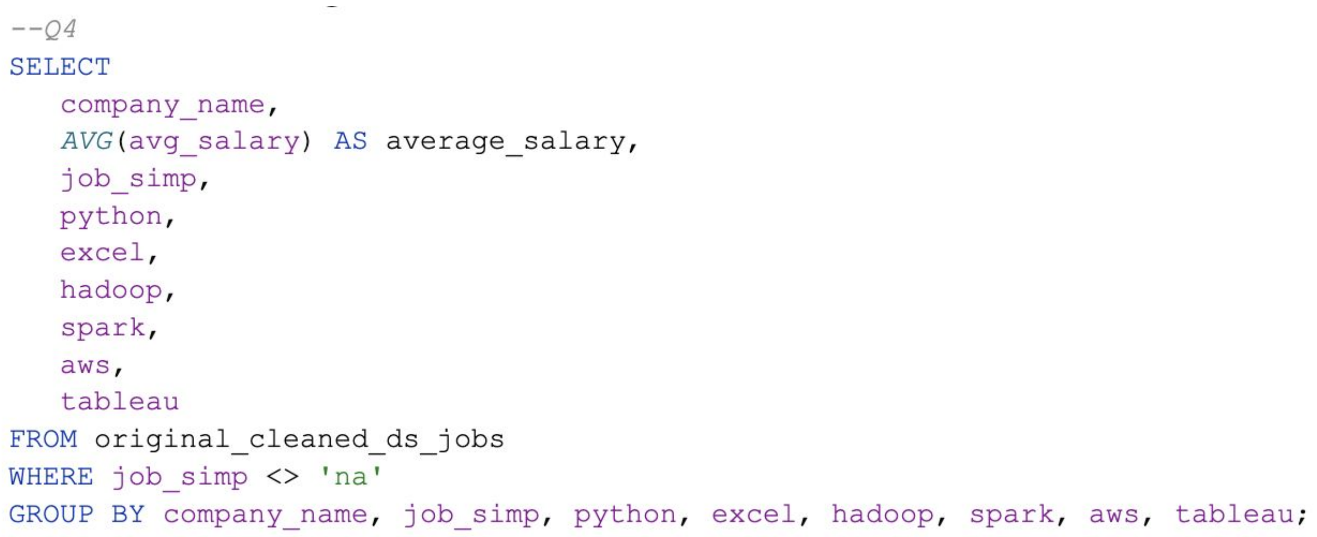
It ideal for analyzing interconnected data like job requirements and company structures in our dataset.

Question #4 and Query #4



Question #4 and Query #4

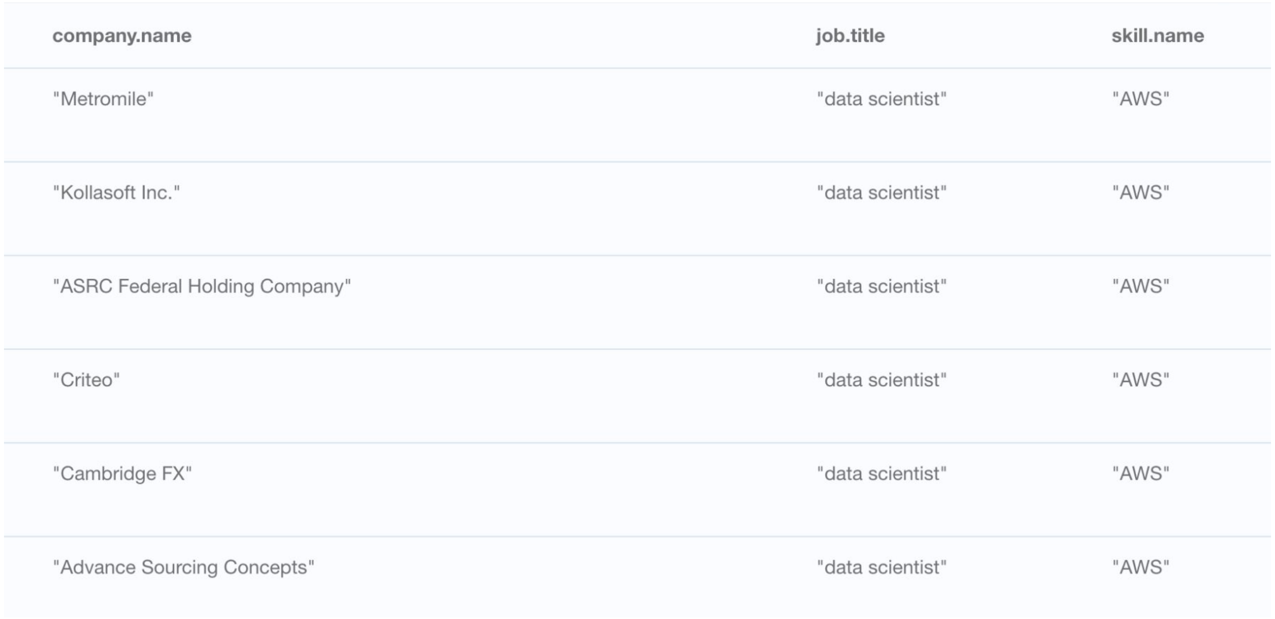
Query:



```sql
--Q4
SELECT
   company_name,
   AVG(avg_salary) AS average_salary,
   job_simp,
   python,
   excel,
   hadoop,
   spark,
   aws,
   tableau
FROM original_cleaned_ds_jobs
WHERE job_simp <> 'na'
GROUP BY company_name, job_simp, python, excel, hadoop, spark, aws, tableau;
```

Question #4 and Query #4

Result:



company.name	job.title	skill.name
"Metromile"	"data scientist"	"AWS"
"Kollasoft Inc."	"data scientist"	"AWS"
"ASRC Federal Holding Company"	"data scientist"	"AWS"
"Criteo"	"data scientist"	"AWS"
"Cambridge FX"	"data scientist"	"AWS"
"Advance Sourcing Concepts"	"data scientist"	"AWS"

Question #5 and Query #5

Question:

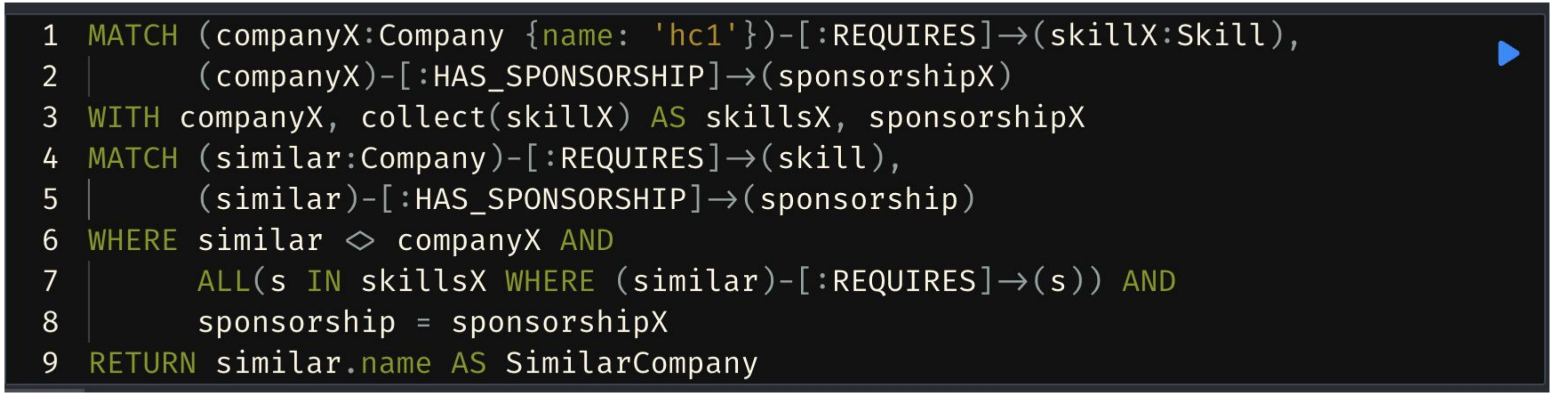
To recommend companies similar to "{Company X}" based on skills and sponsorship history for recommending companies similar to "Company X" based on sponsorship history and company information.

Methods:

In the question, we apply similarity recommendation methods to identify companies with comparable profiles. These methods leverage the interconnected nature of data, where entities such as companies, reviews, and job titles are nodes in a graph, and the relationships between them are the edges.

Question #5 and Query #5

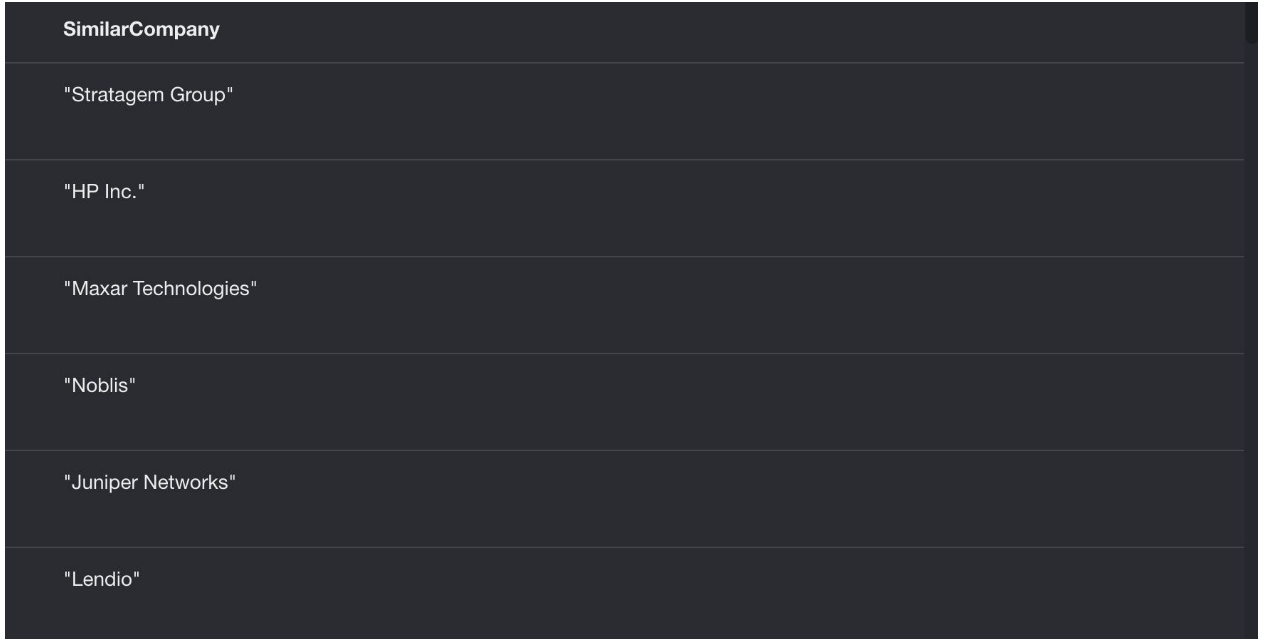
Query (example with hc1):



```
MATCH (companyX:Company {name: 'hc1'})-[:REQUIRES]->(skillX:Skill),
      (companyX)-[:HAS_SPONSORSHIP]->(sponsorshipX)
WITH companyX, collect(skillX) AS skillsX, sponsorshipX
MATCH (similar:Company)-[:REQUIRES]->(skill),
      (similar)-[:HAS_SPONSORSHIP]->(sponsorship)
WHERE similar <> companyX AND
      ALL(s IN skillsX WHERE (similar)-[:REQUIRES]->(s)) AND
      sponsorship = sponsorshipX
RETURN similar.name AS SimilarCompany
```

Question #5 and Query #5

Results:



SimilarCompany
"Stratagem Group"
"HP Inc."
"Maxar Technologies"
"Noblis"
"Juniper Networks"
"Lendio"

Thank you!