**About Neo4j:**

Neo4j is the world’s leading graph data platform. 75% of the world’s Fortune 100 companies use Neo4j for their mission critical applications. We’re building the technology that enables connected data to solve the world’s most pressing problems, such as helping to cure cancers, address diabetes, and get humans to Mars. We're proud to fight fraud, crush pandemics, and even help journalists uncover the truth. Find out more at neo4j.com and follow us at @Neo4j.

**Our Vision:**

At Neo4j, we have always strived to help the world make sense of data.

As business, society and knowledge become increasingly connected, our technology promotes innovation by helping organizations to find and understand data relationships. We created, drive and lead the graph database category, and we’re disrupting how organizations leverage their data to innovate and stay competitive.

## **Job Overview:**

Are you at the end of your studies and you want to immerse yourself in graph technology? We are now looking for students who want to do their Master’s Thesis alongside us at Neo4j!

As part of Neo4j engineering in London, you will work together with a diverse team of talented colleagues from around the world. You will receive advice and continuous support from us - we are experts within graph technology and positioned to help you perform to the best of your ability.

**Past Thesis Ideas**

**Random Generation of Semantically Valid Cypher Queries**  
Database management systems (DBMS) are integral tools at the center of many software applications, which means that these applications are deeply dependent on the correctness of their DBMS. In recent years, graph DBMSs have seen a significant rise in popularity, but they have not gotten the same amount of academic attention when it comes to testing as their relational counterparts. The most popular graph DBMS is called Neo4j and it has its own query language called Cypher. In this thesis, we present a tool that generates random semantically correct Cypher queries. This query generator has a versatile set of use-cases and is built to be configurable, and in this thesis we have focused on using it for random testing of the Neo4j DBMS. Random testing of a DBMS means generating random but correct queries, executing them on the database and then checking whether the output is incorrect, which can be accomplished in a few different ways. We found 25 confirmed bugs in Neo4j with our tool which suggests that it works well for the purpose of random testing of graph DBMSs. 21 of these bugs are already fixed, which suggests that the tool can find significant errors and not just irrelevant edge cases.  
  
**Finding Candidate Node Pairs for Link Prediction at Scale**  
There are methods for inferring whether a pair of people are likely to become friends or whether two kinds of drugs are likely to interact if consumed simultaneously. The methods solve the problem of link prediction, i.e. answer the question "Is a link (friendship, interaction) likely to form between two particular nodes (people, drugs)?". Generalizing the problem to graphs translates it to predicting if particular node pairs are likely to form links. As predicting links between all possible node pairs is computationally infeasible for larger graphs, methods for narrowing down the search space are required to efficiently solve the problem. We propose a novel algorithm, DAPPR, for resolving this issue and compare it against an existing solution LinkWaldo, along with breadth first search and a variant of KNN. The algorithms are evaluated by their ability of finding hidden edges on on real-world graphs, and it is shown that DAPPR outperforms all compared algorithms.

**Preserving Availability in a Consensus Module Using Back Pressure:**  
In distributed systems, the consensus algorithm Raft is used to replicate a globally ordered log of entries. However, members that fall behind in replicating the log entries can cause system write unavailability. One reason for this write unavailability is that Raft needs a majority of members to replicate a log entry, before it is accepted into the system.  
  
**Row vs. column data layout in a graph database query engine:**  
This thesis aims to examine if there is any performance improvement to be gained by changing the memory layout from row-wise to column-wise inside of the Neo4j query engine. In order to test this a column-wise representation was created along with new implementation for a few operators to better leverage the potential of the new memory layout, such as using SIMD. This change means that the query execution strategy is changed from the current approach, which relies upon fusing and compilation, to a vectorized approach instead.  
  
**Categorization of Cypher Queries to Improve Benchmark Coverage for Graph Databases:**  
Benchmarks are often used to find regressions to avoid performance dropping over time. To make benchmarks relevant for a product, the benchmarks should mirror the users’ needs and uses of functionality. To achieve this, user data can be used as a foundation when creating new benchmarks and thus improving the coverage. This thesis was carried out at Neo4j who develops the most frequently used graph database. Using data from their database as a service (AuraDB), we focused on finding a way to improve the coverage of the benchmark suite run by them.

**GraphBLAS:**  
Connected data can be found in a variety of places, ranging from social networks to protein structures. Thereby, graph algorithms can be used to analyze the connected data. To allow users to implement their own graph algorithms, different graph computational models evolved. One model, that recently gained more popularity, is the graph computational model GraphBLAS. GraphBLAS enables users to define graph algorithms based on linear algebra. This thesis evaluates GraphBLAS in the context of existing graph algorithm libraries in Java.  
  
**MAGPIE:** **A Maintainable Graph Pattern Indexing Engine:**  
Even though the benefits of using a path index for evaluating pattern matching queries have been proven empirically by years of research, no industrial graph database has adopted the path index as a core feature. This project summarizes the design of MAGPIE, the first Maintainable Graph Pattern Indexing Engine, which is tailored to be used in the industrial graph database.  
  
**We tackle challenges in:**

* Concurrency and parallelism
* Distributed systems and fault tolerance
* Language design and type systems
* Performance tuning and benchmarking
* Cloud architecture and service design
* Site Reliability Engineering and cloud automation
* Continuous Integration and Continuous Delivery
* Graph algorithms and machine learning

**Please send us a description in English of:**

* Your area of study
* Your thesis idea and area of engineering that it corresponds to
* If you are not completely sure, that is okay - please let us know if you would like to find out more information

**Why Join Neo4j?**

Neo4j is one of the 20 most popular databases in the world, and the leader in the rapidly emerging Graph Database category. We’re building the technology that enables connected data to solve the world’s most pressing problems. You’ll work on unique engineering projects that challenge what’s possible.

We have customers in every industry across the globe, and our products are a proven product/market fit. Joining our team is an opportunity to shape the future of data and analytics.

We pride ourselves on being a welcoming team. Most of us would agree that one of the best parts of working at Neo4j is the people we get to work with.

**We encourage you to apply even if you don’t 100% fit the qualifications above**. Why? Because we are committed to building an environment that fosters belonging, which means valuing intellectually honest discussions and being receptive to new ideas and perspectives.

One of our central objectives is to provide an inclusive, diverse, and equitable workplace for everyone to develop their potential and have a positive, career-defining experience.

**Neo4j Values:**

Neo4j is a Silicon Valley company with a Swedish soul. We foster collaboration and each of us is empowered to contribute and put our innovative stamp on projects. We hire candidates who reflect the following Neo4j core values:

(we)-[:VALUE]->(relationships)

(we)-[:FOCUS\_ON]->(:UserSuccess)

(we)-[:THRIVE\_IN]->(:Culture {type: [‘Open’, ‘Inclusive’})

(we)-[:ASSUME]->(:Intent {direction:’Positive’})

(we)-[:WELCOME]->(:Discussions {nature: ‘IntellectuallyHonest’})

(we)-[:DELIVER\_ON]->(ourCommitments)

Neo4j is committed to protecting and respecting your privacy. Please read the privacy notice regarding Neo4j's recruitment process to understand how we will handle the personal data that you provide.

More information at www.neo4j.com.