**3.8 Session 8 reflection**

In your profession, you will need to continuously learn new types of databases and tools that operate or interact with them. Most of the time, there will be no course or textbook at hand, just the documentation of the new tools you are facing and whatever instruction you happen to stumble upon on some internet forum.

In this reflection, we are purposefully emulating that experience, for which there is no example code that populates or queries a graph database so that you start from scratch, like in the real world. Please read all of the questions before you start working on the responses. Remember to clearly cite all your sources, even when you did not end up copying anything verbatim from them. Also show your code, either by including snippets in the responses or linking files in a GitHub repository or similar.

Each adequate response is worth one point.

1. Identify and describe a scenario in which there is a clear need to use a graph database (hopefully something other than a social network site, but just do that if you can think of nothing else).
2. Create a plain-text file with example data that represents a property graph, either in CSV or JSON, with at least three vertices and at least two edges.
3. Populate a Neo4j database with that file, preferably using the command-line shell utility.
4. Write at least one relevant query for the newly created database and show its results on the interactive shell.
5. Read through the Drivers & Language Guides section in Neo4j documentation and pick a programming language to use in the next step; discuss the reasons behind your choice.
6. Connect to the database and execute one or more of the queries from the fourth step through the language you chose in the fifth step.
7. Try to think of a real-world scenario in which it would not make sense to limit to one kind of a database (relational, noSQL, or graph) but instead two or more kinds are needed for an efficient implementation. Describe the scenario you thought of or, if you could not think of one, explain why you think it is unlikely to be useful to combine two or more kinds of databases.
8. Pick a job-posting website (like LinkedIn or Indeed, for example) and search for jobs that require database skills. Visualize in a chart of your choice the popularity of the different database technologies among the results.
9. Do you perceive any differences in the salary range (when indicated) for the positions you examined in step 8, depending on which database technologies are requested? If salaries are rarely indicated, why do you think that is?