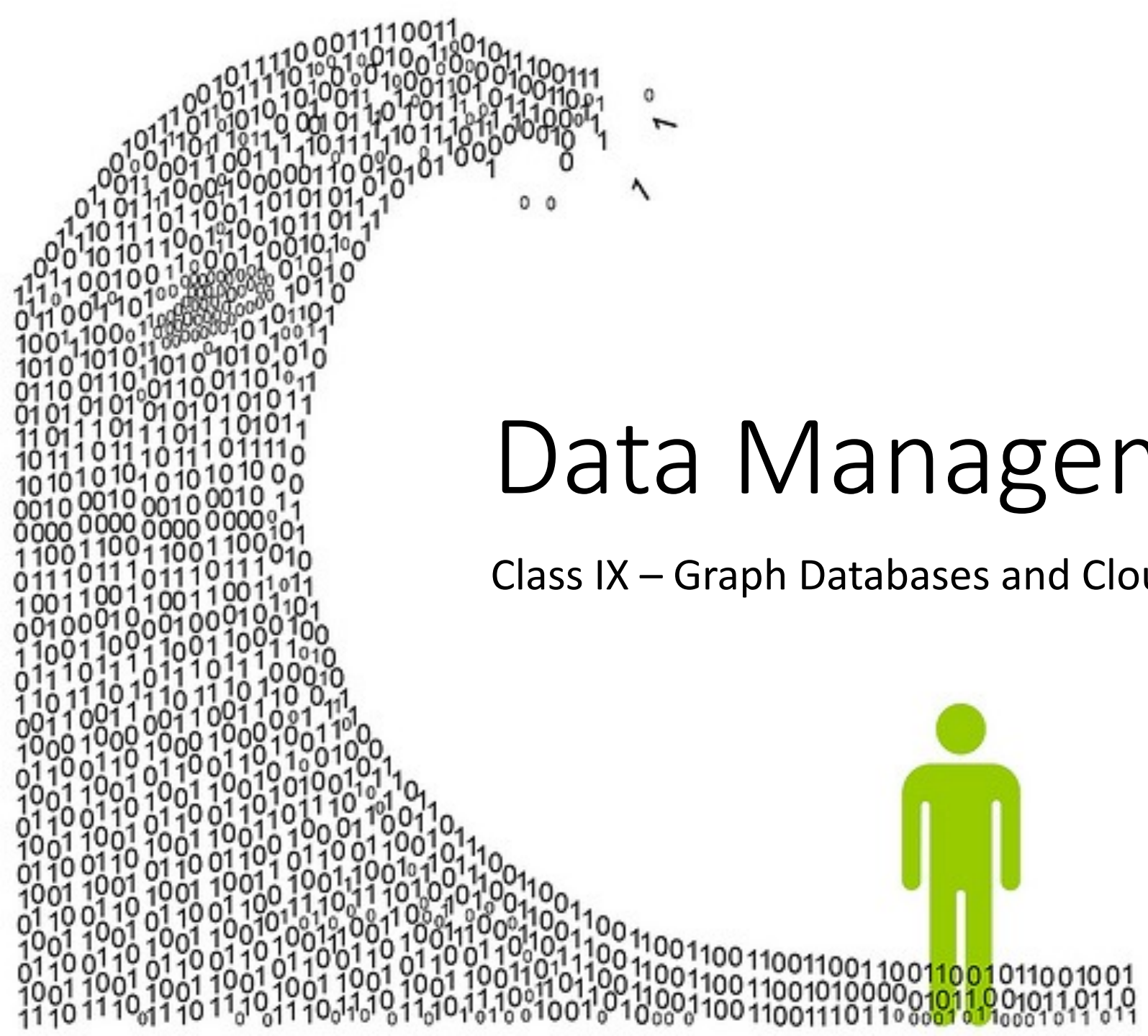
Data Management

Class IX – Graph Databases and Cloud Services



Revision

- We have now learned how to use two very different types of database – SQL and NoSQL. We've imported our own research data into each of them and run various kinds of search queries.
- We've also seen how to use indexing to speed up database operations, and how to stream data so we can analyse data sets that are too big for the computer's memory.

Database Best Practices

- Finally for our section on SQL and NoSQL, I thought we should talk a little bit about how to most effectively use databases.

- There are some things about databases which are different to how you're used to doing things, and these can be a bit counter-intuitive.

1) Storage is Cheap

- A lot of people have the instinct to try to minimise the amount of data they store – they avoid using too many tables, for example.
- In reality, **storage is cheap**. Even your smartphone probably has at least 64Gb of storage – enough to store vast amounts of text data. Cloud services only charge a few cents a month for storing gigabytes of data.
- So… Don’t be afraid to store *lots* of tables, and in general, *never delete data*. Instead of deleting data from tables, make a column called “deleted” (or whatever you like) and set it to True when you don’t want to use that data. You never know if it might be useful in future.

2) Processing Takes Time

- Storage is cheap… But processing always takes time, even on the fastest computer.
- Your objective should be to make sure that you never have to recalculate the same analysis result twice.
- So, when you run an analysis – especially one that takes time, like tokenisation – you can often benefit by making a new table and storing your analysis results.
- Next time, you just look up the result in the database, rather than having to run the analysis all over again!

3) Back up, back up, and back up some more.

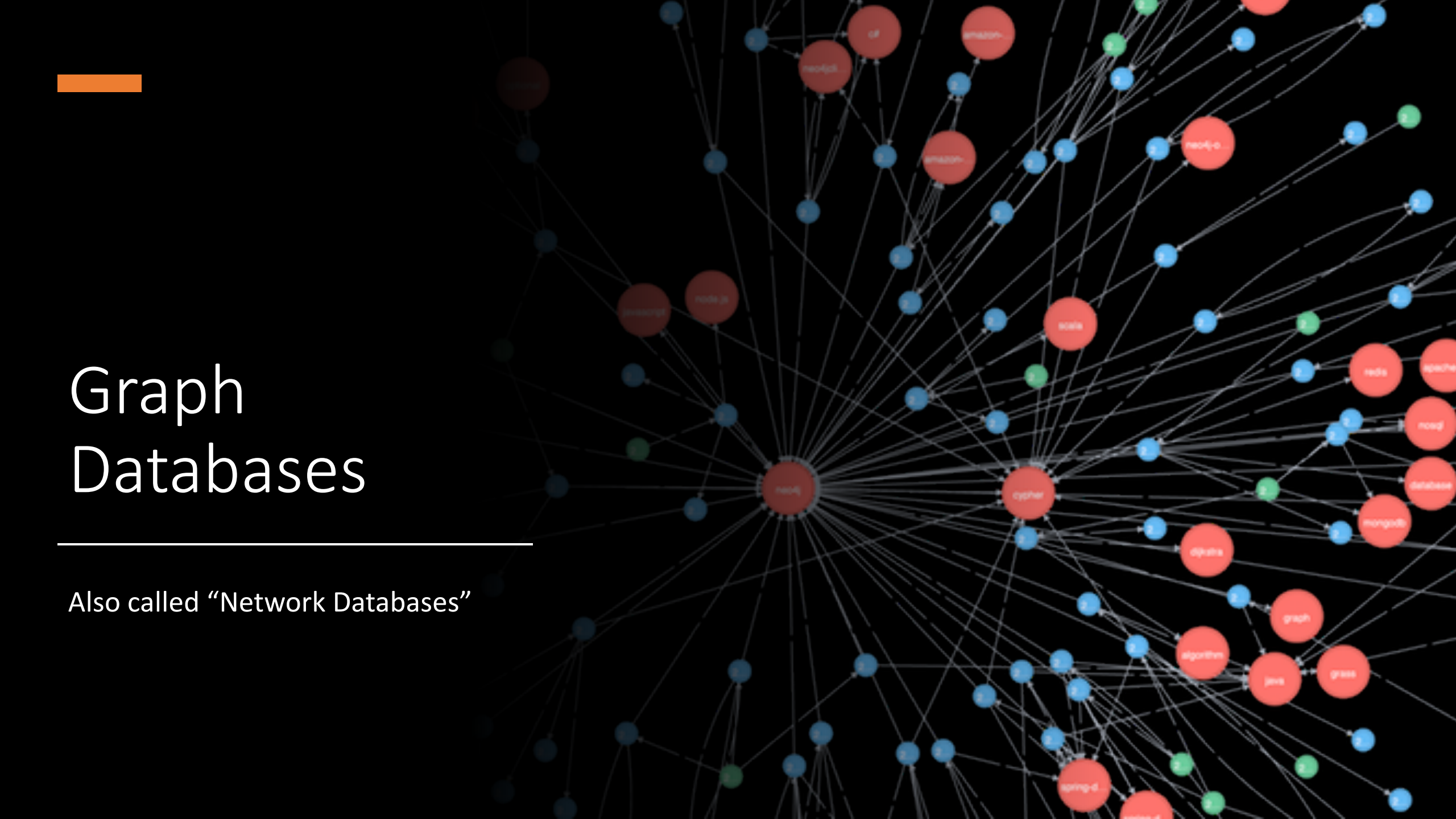
- Databases do a lot of clever stuff to keep your data safe… But ultimately, if they're running on your laptop, they can't rescue you if you drop the laptop in a lake, or it gets stolen in a bar.

- MySQL provides a command line program called 'mysqldump' which will 'dump' your databases and tables out into a backup file – which you can then save on a network drive, an external hard disc etc.
  - MongoDB gives you a similar tool called 'mongodump'.

- You can download tools to do this automatically every few days. It's a really, really good idea to do so!

Graph Databases

Also called “Network Databases”



Graph Databases

- SQL databases are incredibly powerful and fast when storing regular, well-ordered tables of data; NoSQL databases are extremely flexible and capable of storing all sorts of data types together easily.
- In recent years, though, one specific type of data has become incredibly important for social science research: network data, also known as **graph data**.

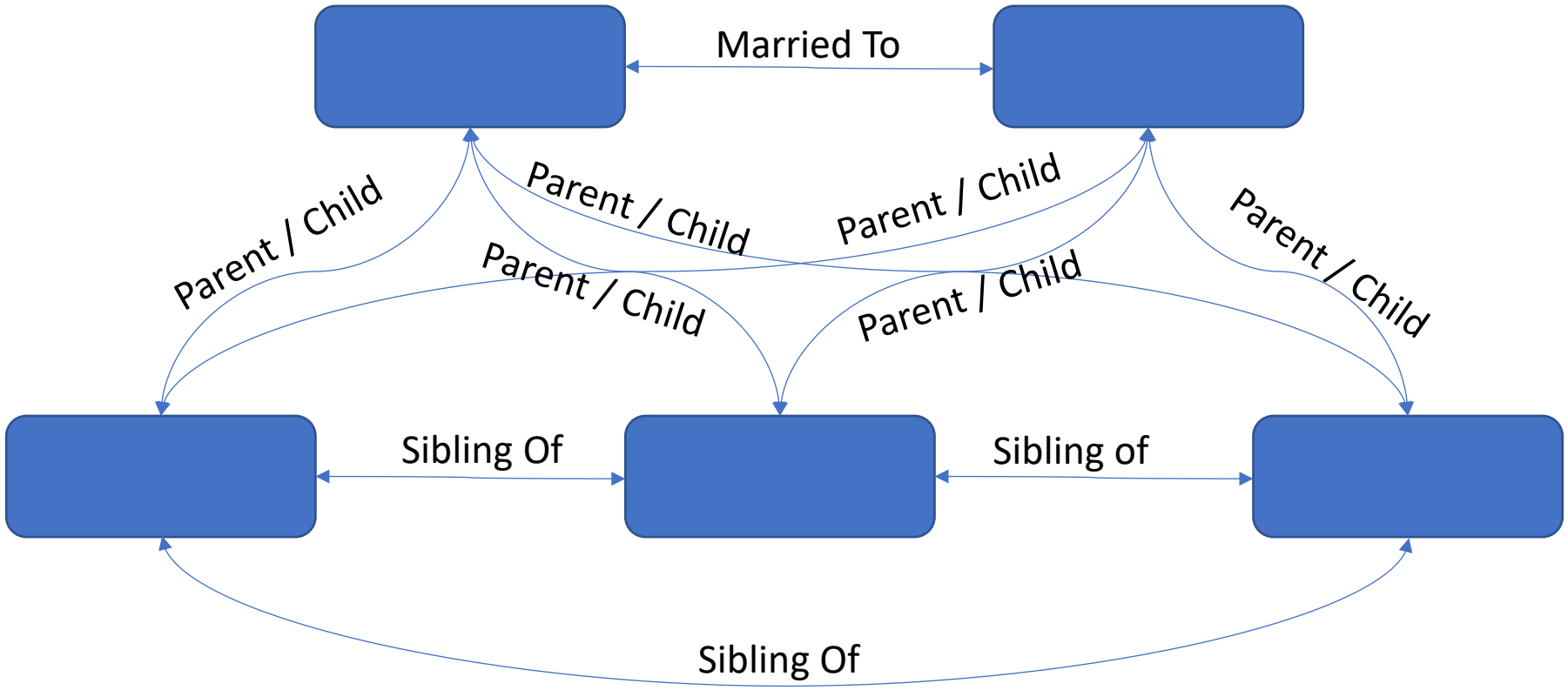
The Structure of Graph Data

- While SQL tables are made of rows and columns, and NoSQL documents are made of keys, values, lists and sub-documents, Graph Data is made of **nodes** and **edges**.

- A “node” is a point on the graph.
- An “edge” is a connection between two points.

- Each of these items can have its own associated information, made of keys and values – like a NoSQL document.

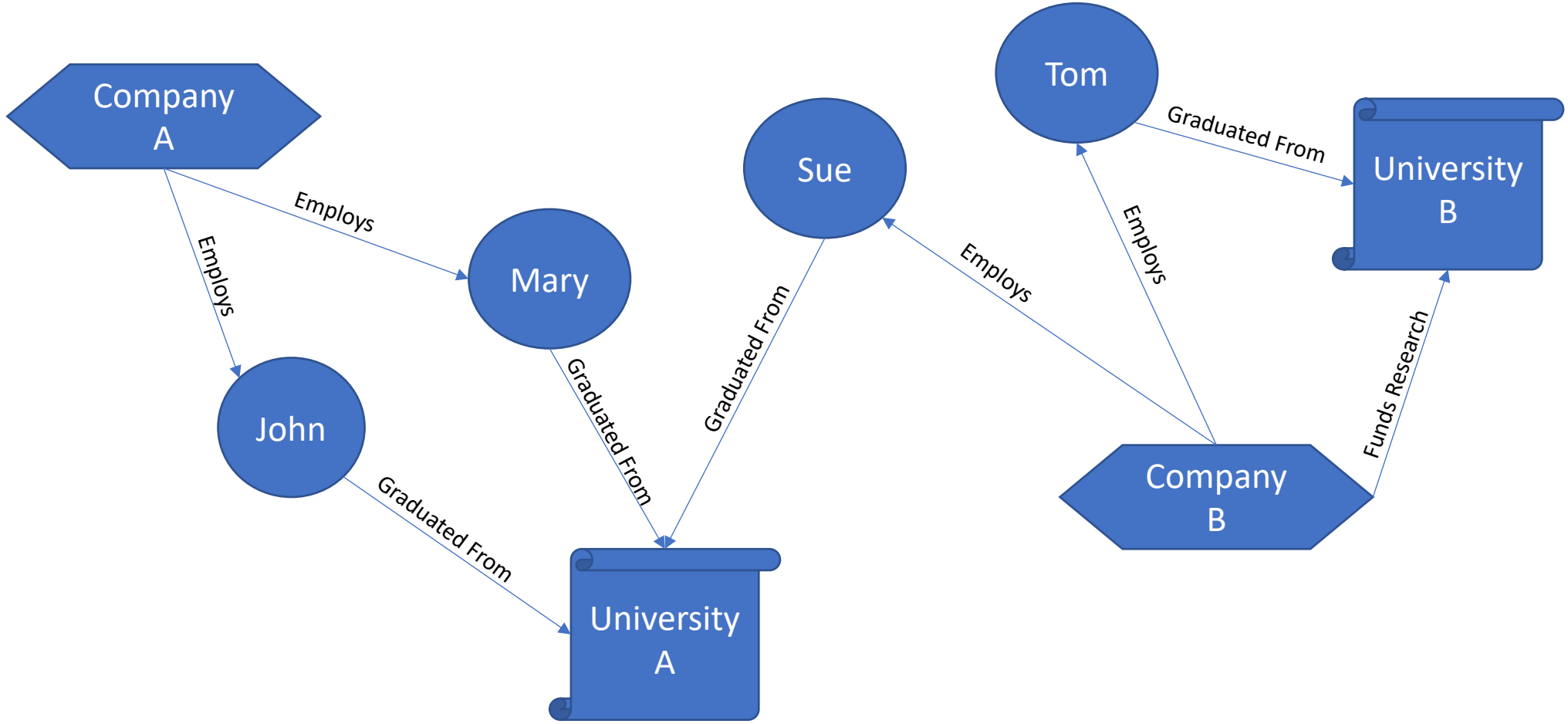
Nodes and Edges: Example



More Complex Graphs

- In the previous graph, each node was of the same **kind** – it represented an individual person (i.e. a member of a family).
  - The node might contain very different information about each individual, but fundamentally each node represents the same broad concept – i.e. a person.
- However, the edges were not all the same; each edge could represent a different kind of relationship.
  - Edges can also include lots of different information; in some cases they can also be “weighted”, using numeric variables. For example: a count of how many times two users in a network have interacted with each other.
- We can also have graphs where there are many different kinds of node.

Graph with Multiple Node Types



Graph Database Software

- The most popular graph database at the moment is Neo4J, which uses a graph database language (similar to SQL for regular column/row databases) called “Cypher”.
- However, graph databases are very new, and unlike with SQL, it’s hard to say what language or products will be popular in future.
- Also… While graph databases are very powerful and useful, many researchers still prefer to use more traditional tools (like SQL databases!) for graph-based data.
  - Traditional databases are more familiar for many people, they’re older and more established, and in many cases, they’re higher performance.

Let’s try out Neo4J

- Rather than installing and trying out Neo4J directly, we’re just going to look at some demonstrations of it in class.
- If you do want to try using it, you can either install it and use it directly from the Neo4J Browser (as I’ll be doing in the demonstration)…
- … or you can try controlling it directly from R (as we did with MySQL and MongoDB) using the package **neo4r**.

So why wouldn’t you use a Graph Database?

- As you can see, graph databases let you explore your data in a lot of very powerful and intuitive ways very easily.
- They’re especially good for projects where you’re investigating relationships between entities of different kinds.
- **However**: They’re not very performant (meaning they can be very slow with large amounts of data) and are generally not well suited to extremely large data sets.
  - **This includes social media data, if you‘re collecting large-scale data.**
- For those cases, you’re often better off using SQL (which is very high performance) and combining it with graphing tools in R or Python.

Cloud Services

“Just someone else’s computer”

- Yes, the Cloud is really just a huge number of powerful computers belonging to Google, Amazon, Microsoft and so on.
- However, that simple description ignores the huge benefit of storing and processing your data on “someone else’s computer”.
- Those computers are:
  - Much faster than your laptop;
  - More secure and regularly backed up than your laptop;
  - Have far more memory and storage than your laptop;
  - and are connected to the Internet at much higher speed than your WiFi connection.

Cloud Service Providers

- There are a lot of different Cloud Service providers out there. The three big ones are:
  - Amazon (Web Services, or AWS)
  - Google (Cloud Platform, GCP)
  - Microsoft (Azure)

- All of those companies offer basically the same kinds of cloud services – and there are many smaller companies, such as Digital Ocean, which also provide specific kinds of cloud service.
- My research teams use Google Cloud, but AWS and Azure are also good options.

What is a “Cloud Service”?

- Any company’s Cloud is made up of a host of different “services”. Services include things like:
  - Servers and ‘virtual machines’ where you can run code – doing analytics, running a website, etc.;
  - Data storage services like databases or long-term data archives;
  - API services giving access to advanced technologies like AI and machine learning, language translation, image or speech recognition, etc.
  - Support services like security, communications etc., which help to tie together all those other services and allow them to work together.

Databases in the Cloud

- What we're interested in in this class is the potential for storing, accessing and sharing data in the Cloud.
- There are two very different approaches to running a database in the Cloud…
  - You can run standard database software on a Cloud server – so for example, you could have MySQL or MongoDB running on a server provided by Amazon or Google. This would let all of your team access it from anywhere in the world.
  - You could use a native Cloud database. You'd never know or care about where or how your data is stored – just put data in and take it out, trusting the Cloud company to worry about all the rest.

When to use the Cloud

- Cloud services are a great option in certain circumstances.
  - If your data are simply too large to fit onto a standard laptop, huge Cloud databases can be a great solution to the problem.
  - If your data needs to be worked on by a lot of different people around the world, running a database in the Cloud is an effective solution.
  - If you have at least one team member who is very literate about IT and programming (maybe this could be you?), and can set up Cloud access, security etc. for everyone else, then using a Cloud service can be a great idea.
  - If you need to do something like running a web service (e.g. creating an interface for student RAs to work on tasks), Cloud services are ideal.

When **not** to use the Cloud

- The Cloud is a popular buzzword… but it's not always the answer to a research team's problems.
  - If you don't have someone who is a decent programmer and can get Cloud services working well for your team, trying to do so will just cause extra problems for you in the long run.
  - If your data are small enough to just share on Dropbox, you don't need the Cloud (well, Dropbox kind of *is* a Cloud service…).
  - If your data are extremely sensitive – e.g. commercial or private information – then you probably shouldn't put it in the Cloud unless you're *incredibly* confident about your IT security skills.

Today’s Assignment

- For today’s assignment, I want you to be somewhat creative.

- Think about your capstone project, or the data you downloaded from Kaggle, as a starting point.

- Now I want you to imagine that you have as much budget / resources as you need to do the research project of your dreams on that topic.

  You can recruit colleagues from other institutions or other countries, work with the help of research assistants, and use whatever IT / server / cloud resources you need.

Today's Assignment

- **Tell me about the project you'd like to do in that situation**.

- For now, don't worry about what kind of data management you'd need. I want you to design your ideal research project – think about the team you'd need, the data you'd need, the kind of analysis you'd need to do. Describe this project in a few paragraphs.