Topic 11 Big Data

What is Big Data?

Big data is a broad term for special sets of data which can be challenging to work with. A data set can be called big data if it matches one of three defining characteristics known as the three Vs:

- **Volume**, meaning there is a significant amount of data, making the data set too large to be stored on a traditional server or hard drive. This data must be stored across multiple servers.
- **Velocity**, meaning data is rapidly changed, deleted or created. This requires the serves hosting the data to respond quickly to a large number of requests.
- **Variety,** meaning the data consists of a large number of different data types, for example a data set made up of pictures, videos, and text files.

Conventional database are not well suited to storing big data because its unstructured nature means it does not fit neatly into the rows, tables and columns a database requires. Conventional databases also do not scale well to fit the huge data sets which store big data. It is this lack of structure which makes it the most difficult to work with and analyse big data.

Machine learning techniques have to be used to allow computers to spot patterns in the data in order to analyse and extract useful information from it.

Where data is stored on multiple different servers, the processing of it also has to be split. This is incredibly difficult with conventional programming paradigms because they would require all the computers to be synchronised together in order to prevent data being damaged or overwritten.

Functional Programming

Functional programming helps to solve this problem of processing large data sets across multiple computers. Functional programs make use of imitable data structures, and are stateless. They also support higher order functions.

The Fact-Based Model for Representing Data

The fact based model stores each piece of information as a fact, with each fact being immutable. This means that once a fact is created, it cannot be deleted or changed. This model also reduces the risk of data being lost or changed due to human error. It also does not require an index as data is simply added to the existing model.

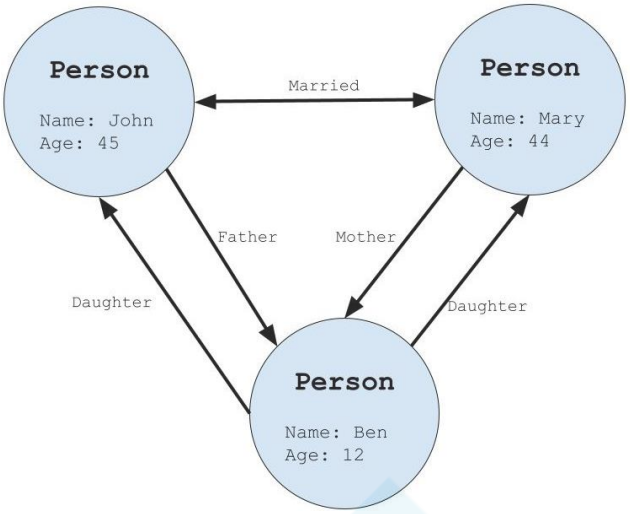
A timestamp is stored with each fact to record the time at which the fact was created. Because facts cannot be deleted it is possible to have multiple values for the same attribute, and the timestamp allows computers to see which fact is most recent and to spot patterns of changes over time.

Using Graph Schema to Represent Big Data

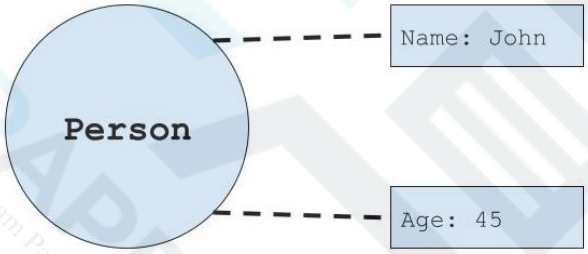
Graph schema uses graphs, made from nodes and edges, to visually show the structure of a dataset. Each node represents an entity within the dataset along with the properties of the entity. Edges meanwhile show the relationships between entities along with a description of the relationship.

Timestamps are not usually included within graph schema diagrams, as each node should contain the most recent information available.

The example below shows a graph schema with three entities and their properties, each represented as circles. Arrows between the circles show the relationships between entities.



We can also use the approach shown below to represent entities. In this case the entities properties are within rectangles joined to the entitiy with dotted lines. It is important to remember that dotted lines do not reproesent properties.



Graph schemas can be easily extended simply by adding a new node with appropriate edges and properties. Adding new data does not impact existing fact types since they are immutable. This allows graph schemas to easily adapt and capture complex evolving systems.