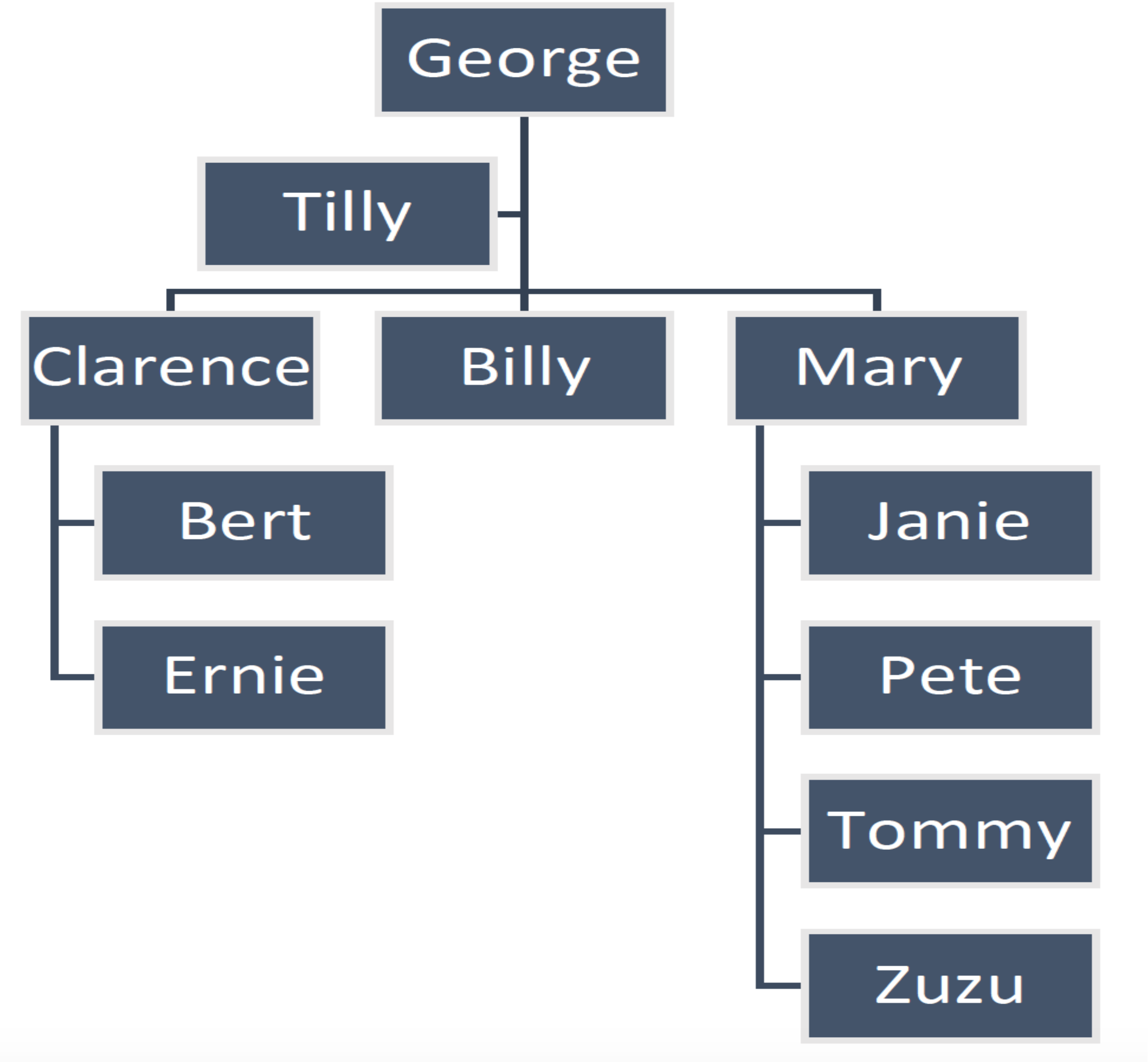
Employee Hierarchy Data

In your master data, a hierarchy is any structure where nodes have other nodes above and below them, possibly with multiple branches. One example of a master data hierarchy is employee reporting and supervisory structures like the one pictured below.



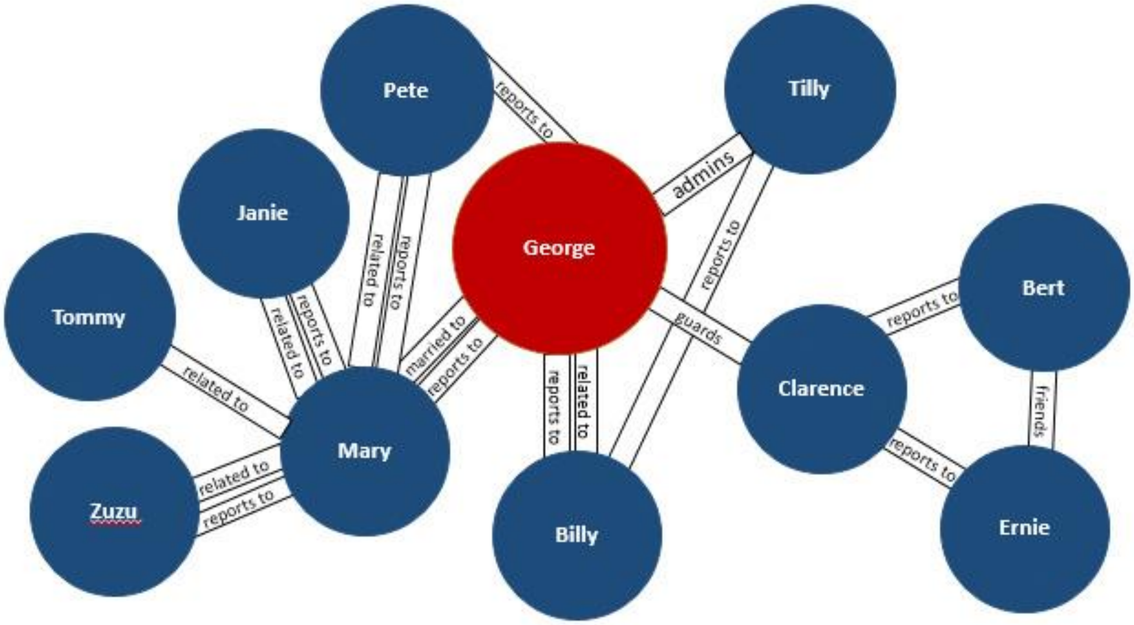
A master data hierarchy illustrating employee reporting and supervisory structures. Traditionally, this hierarchy would serve as a model in a relational database.

A small hierarchy such as the figure above is easy enough to model and maintain in a relational database. But as soon as we model a much larger set of employees, both querying and maintaining the data gets more expensive.

For example, if an employee gets a promotion, every relationship must be reset for every hierarchy in which the employee participates.

Of course, such pure hierarchies rarely exist in the real world. Employees often report to a multiple people, and sometimes reporting relationships exist only for transitional reasons (such as job shadowing or coverage).

In fact, most business hierarchies are actually networks filled with real-life complexities and many kinds of relationships.



A master data network detailing employee reporting and supervisory relationships, this time with more real-life complexity. This network would serve as a model in a graph database.

Traditional hierarchies need to be reimagined as networks that are easier and more flexible to model with a graph database as business needs change.

While the example discussed has to do with employee reporting relationships, the same principle of master data networks applies to product listings, document relationships and sales or customer data.

NODE INFORMATION

ID	NODE INFORMATION
1	George
2	Pete
3	Mary
4	Billy
5	Tilly
6	Janie
7	Tommy
8	Zuru
9	Clarence
10	Bert
11	Ernie

Relationship Between Nodes

Source	Destination	Edge Type
1	2	Reports to
1	3	Married to, Reports to
1	4	Related to, Reports to
1	5	Admins
4	5	Reports to
1	9	Guards
9	11	Reports to
9	10	Reports to
10	11	Friends
2	3	Related to, Reports to
3	6	Related to, Reports to
3	7	Related to
3	8	Related to, Reports to

The Key Challenges in Master Data Management:

Today’s enterprises are drowning in “big data” – most of which is mission-critical master data – and managing its complex relationships can be quite a challenge. Here are some of the most complicated hurdles in MDM that enterprise’s must face:

- **Complex and hierarchical datasets**

  Master data, such as organizational and product data, has deep hierarchies with top-down, lateral and diagonal connections. Managing such data models with a relational database results in complex and unwieldy code that is slow to run, expensive to build and time-consuming to maintain.

- **Real-time storage and query performance**

  The master data store must integrate with and provide data to a host of applications within the enterprise – sometimes in real time. However, traversing a complex and highly interconnected dataset to provide real-time information is a significant challenge.

- **Dynamic structure**

  Master data is dynamic in nature, making it harder for developers to design systems that accommodate its evolution.

Why Use a Graph Database for Master Data Management?

Because master data is highly connected and shared, poorly built MDM systems cost business agility in a way that ripples throughout your enterprise. Most legacy MDM systems rely on a relational database which isn't optimized for traversing relationships or rapid responsiveness.

These data connections and relationships in your master datasets are essential to competitive advantage as business analytics evolve. The good news is that graph databases are ideal for modeling, storing and querying the hierarchies, metadata and connections in your master data.

With graph databases, your master data is much easier to model, costing you fewer resources (modelers, architects, DBAs and developers) than building a relational solution. In addition, with a graph database, you don't have to migrate all of your master data into a single location. Graph relationships easily connect your siloed data between CRM systems, inventory systems, accounting and point-of-sale systems to provide a consistent vision of your enterprise data.

Conclusion

The best data-driven business decisions aren't based on stale information silos. Instead, you need real-time master data with information about data *relationships*.

Graph databases are built from the ground up to support data relationships. With more efficient modeling and querying, organizing your master data in a graph yields relevant answers faster and with more flexibility than ever before.