The major advantages of NoSQL Storage techniques over SQL Based Storage is that there are no complex relations as the ones between tables in relation database management systems. This means that the concept of joining various tables is not used in NoSQL Storages. There are no inherent schemas in a NoSQL Storage, hence both structured and unstructured data can be stored with ease in a NoSQL Storage. NoSQL have a high level of flexibility and Scalability with respect to adding data and work very effectively when working in a Distributed-Computing Environment. Also, as the size of the application goes on increasing, the NoSQL storage becomes a more cost-effective solution as compared to SQL based storages which consume a lot of resources for storage and processing.

The group of NoSQL databases collectively known as aggregate stores (term coined by Martin Fowler) includes key-value stores, wide-column stores and document stores. Most NoSQL systems are aggregate-oriented, grouping the data based on a criterion and the database type (such as document store, key-value pair, etc). This model provides only simple, limited operations and only forms one dedicated view of your data. Focusing on one aggregate at a time allows users to easily spread many chunks of data across a network of machines along the aggregate dimension (for instance, the Document in document databases), but that means that other projections and perspectives have to be computed by crunching or duplicating your data. Graph databases, on the other hand, handle fine-grained networks of information, providing any perspective on your data that fits your use case. The well-known and trusted transactional guarantees from relational systems also protect updates of the graph data, conforming to ACID standards.

When you shop on Amazon or watch series on Netflix, those platforms search through your viewing and purchasing history, and people deemed to be like you, to recommend items you might want to view or purchase next. Netflix looks at IMDB. Amazon looks at what people who have purchased the item you just bought. For example, if you buy an iPhone then it might suggest an iPhone cover. A graph database is an online database management system with Create, Read, Update and Delete (CRUD) operations working on a graph data model. Unlike other databases, relationships take first priority in graph databases. This means your application doesn’t have to infer data connections using things like foreign keys or out-of-band processing, such as MapReduce. Graph databases are built for use with transactional (OLTP) systems and are engineered with transactional integrity and operational availability in mind. Hence, a Graph based Database is the best option for such a case scenario.

Whenever we need to choose a storage method for any application, we need to follow the basic ACID (Atomicity, Consistency, Isolation and Durability) Principle. Apart from these, Performance, Robustness, Reliability, Scalability, Stabilization and Recovery Time are also important factors that need to be considered when we need to make a choice of Data Storage Systems. But, choosing the best option becomes very difficult if we evaluate every option with all the above factors. Hence, the best choice for any use case scenario by default depends on the Type of Data and Application. If the Application demands a Storage with very less Stabilization and Recovery Time, we have to go for a system which allows less customization with respect to the storage data types and structures and vice-versa. Hence, the application and use case decides the trade off between the various choices.

References:

https://neo4j.com/developer/graph-db-vs-nosql/

https://www.bmc.com/blogs/graph-database-use-cases/

https://neo4j.com/why-graph-databases/