ONTOTEXT ANSWER:

Security requirements in enterprises can get very stringent. Beyond simple RBAC user management, which GraphDB fully supports, you may need to make sure users have some specific permissions. Or, there may even be a requirement that your data is compartmentalized so one attack only compromises some of your data.

You can achieve logical and physical separation, but you should do this without nested repositories. Nested repositories were an experimental feature in the GraphDB line before GraphDB 10. As a feature, it was somewhat complex to configure and manage. Therefore, in more recent versions, we deprecated this feature and removed it from the documentation. If you are still using this feature, let us know, it's interesting to hear from you. The requirements that drive nested repositories can be achieved without them.

Logical separation – GraphDB (or any graph database, really) has a way to achieve this, using named graphs – also known as contexts. Named graphs would allow you to split your data up for ingestion and query purposes while still keeping it within the same repository to allow inference, secondary indexing and plugin logic to operate on the whole dataset transparently. If you have a few named graphs, you should enable the context index for faster processing.

At the present time, we don't have security controls at this level of separation in SPARQL. There is some logic that would control security at this level in our GraphQL plugin. We aim to introduce context-level security in a later GraphDB release.

Physical separation, first degree – GraphDB has a repository concept. That roughly maps to a table/database concept from SQL. You can have multiple repositories, each with its specific read and write permissions. Then you can make cross-repository requests using standard SPARQL or FedX federation. When using federation, the individual permissions of the user would be respected. This applies to repositories in a cluster as well.

When it comes to physical separation of this degree, the index files for one GraphDB instance are on the same filesystem. Each repository has its own directory, under the same data directory parent. You can use third party solutions to encrypt your data, but if one repository is compromised, all of them would be.

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com

Physical separation, second degree – if you want to be certain that the data is separated, it is possible to use multiple GraphDB installations. There's no built-in capabilities for this in GraphDB, but cross-instance communication can be handled via the federation procedures established earlier.

Help us get better: review GraphDB on Gartner

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com