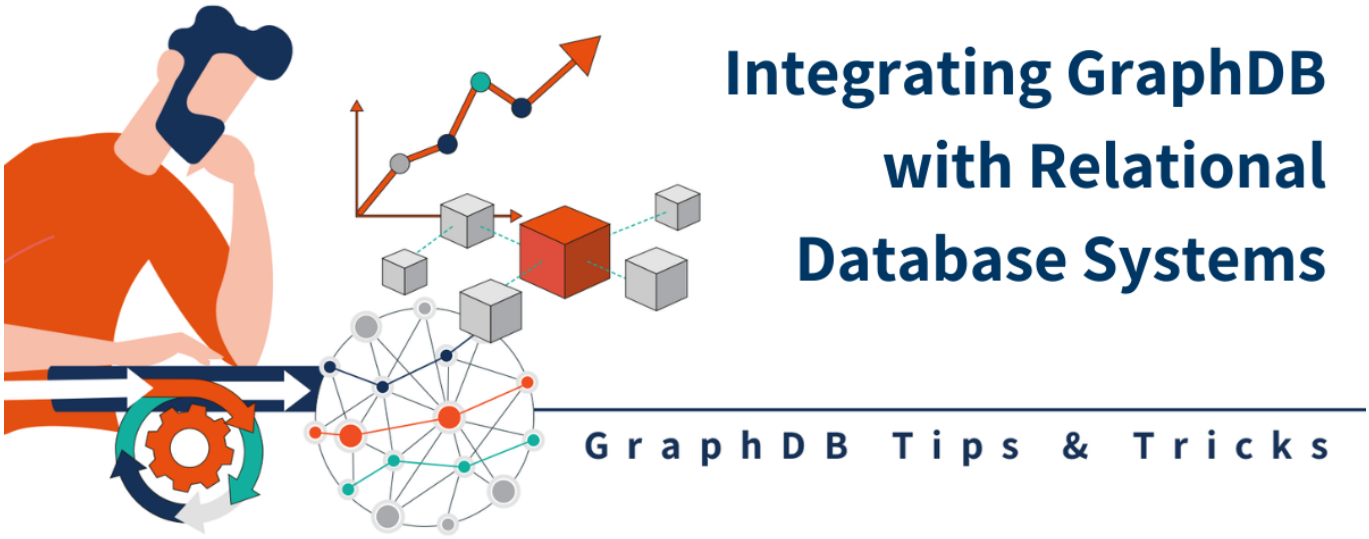
ontotext



As you might guess from its name, GraphDB stores data in a graph data structure, which is much more flexible than the rigid table structures used by relational database managers. Relational databases have been around for a long time, though, and vast amounts of data are stored in them. Applications that can take advantage of both RDF graph data and relational data can get the best of both worlds.
Two GraphDB features give you ways to do this from two different perspectives:

SQL Access Over JDBC

SQL access over JDBC lets you use a SPARQL SELECT query to define a table of data as a view that external relational database tools can access with a JDBC driver. Tableau and other Business Intelligence products that can use JDBC to access data from relational database managers can then access data in a GraphDB repository the same way.

To see this feature in action, the video below is available to demonstrate how you can share your knowledge graph data with relational applications:

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

ontotext

Relational Database Virtualization

Relational database virtualization gives your GraphDB instance access to external relational data. You create a “virtual” repository that, instead of storing RDF data, is configured to point at a database in a relational database manager such as Oracle or SQL Server so that SPARQL queries against the virtual repository are automatically translated to SQL queries against the external relational database. For the GraphDB user, the returned results look like they were delivered by a SPARQL engine and can play the same role in your applications.
The video below will demonstrate how your knowledge graph applications can take advantage of relational data:

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

ontotext

Still have no GraphDB instance?

Download Now

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