



What is Virtuoso, and why is it important?



Virtuoso is a uniquely designed multi-function platform, referred to as a Universal Server, comprising Data Access, Integration, Virtualization, and Management capabilities.

Virtuoso achieves all of this with unrivalled performance, scalability, and cost-effectiveness, without compromising data security.

Platform Design Principles (1)

- Use of the Abstract Relational Model for Structured Data Definition
- Use of Hyperlinks that function as Super Keys for Entity, Entity Type, and Entity Relationship Type Identification
- Use of 3-Tuples, 4-Tuples, n -Tuples, for Structured Data Representation

Platform Design Principles (2)

- Negotiable Structured Data Serialization via HTTP Protocol
- Declarative Query Language (SQL and SPARQL) support for Data Definition and Manipulation operations
- Native Support for Open Data Access Protocols (ODBC, JDBC, and HTTP) as a Data Access Consumer and/or Provider



Platform Design Principles (3)

- Built-in Extract, Load, and Transform (ELT) and/or Extract, Transform, and Load (ETL) functionality
- Built-in Reasoning & Inference usable via both SQL and SPARQL query languages
- Built-in Entity Relationship Graph and Network Analytics usable via both SQL and SPARQL query languages



What Problem Does It Solve?



Prevalence and persistence of Data Silos arising from an Application-centric approach to software solution development and deployment



How Costly is Virtuoso?

Virtuoso is extremely cost-effective across both Sticker Price and Total Cost of Ownership (TCO) dimensions.

Exploiting Virtuoso's unique functionality doesn't require an expensive and high-risk "rip & replace" approach, due to OpenLink Software's core philosophy of "maximum incorporation with minimum disruption," enabled by deep and innovative support of open standards.

Existing SQL-compliant apps and services are augmented through the same ODBC or JDBC access they've always used, while newer HTTP-based solutions simply leverage hyperlinks.



Cost-Effectiveness — Live Examples

There are 24/7-live instances of DBpedia-Live (legacy), DBpedia-Live (new), and Wikidata Instances, supporting ad-hoc SPARQL Queries with unpredictable —

- **Query Complexity**
- **Solution Size**
- **Concurrent Users**

Host Machine Configuration
(One machine supports all of these live instances — simultaneously!)

- **CPU** — 2x Intel® Xeon® CPU E5-2630 0 @ 2.30GHz
- **CPU Cores** — 24
- **Memory** — 378GB
- **Storage** — 4x Crucial M4 SSD @ 500 GB

How Do I Get Going?

- On Premise — standard installer and Free 30-day Evaluation licenses from our Download Site
- Docker Container — various containers from Docker Hub
- Cloud — Bring-Your-Own License (BYOL) and Pay-As-You-Go (PAGO) instances in the Amazon Web Services (AWS) or Microsoft Azure Clouds



Post installation

You are immediately positioned to complete the following tasks:

- Use ETL or ELT operations for progressive Database or Knowledge Graph population
- Attach external ODBC- or JDBC-accessible Data Sources for Data Virtualization
- Use existing ODBC-, JDBC-, or HTTP-compliant tools to operate on Databases and Knowledge Graphs

ADDITIONAL INFORMATION

- [Virtuoso Home Page](#)
- [Virtuoso Weblog](#)
- [Virtuoso Community Forum](#)
- [What is the Business Benefit of Virtuoso?](#)
- [Five Simple Steps to Experience the Power of a Knowledge Graph, using Virtuoso](#)
- [Hyperlinks as Powerful Data Source Names](#)
- [Entity Relationship & Network Analytics using Virtuoso](#)
- [Virtuoso Unique Value Proposition](#)
- [Here's Why The Abstraction Relational Model is so important](#)
- [What is the Virtuoso Sparger Middleware?](#)
- [What is a Virtuoso SPARQL Endpoint, and why is it important?](#)
- [Conceptual Relational Data Virtualization, using Existing Open Standards](#)
- [Magic Sets and Custom Inference Rules in Virtuoso 8.x](#)

