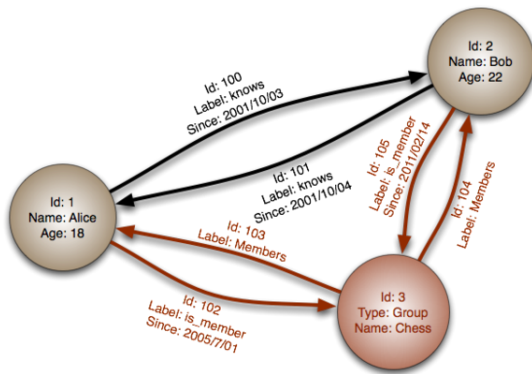


# Graph database

In computing, a **graph database** is a database that uses graph structures for semantic queries with nodes, edges and properties to represent and store data.



Graph databases employ nodes, properties, and edges.

## 1 Structure

Graph databases are based on graph theory. Graph databases employ nodes, properties, and edges.

- Nodes represent entities such as people, businesses, accounts, or any other item you might want to keep track of.
- Properties are pertinent information that relate to nodes. For instance, if “Wikipedia” were one of the nodes, one might have it tied to properties such as “website”, “reference material”, or “word that starts with the letter 'w'”, depending on which aspects of “Wikipedia” are pertinent to the particular database.
- Edges are the lines that connect nodes to nodes or nodes to properties and they represent the relationship between the two. Most of the important information is really stored in the edges. Meaningful patterns emerge when one examines the connections and interconnections of nodes, properties, and edges

## 2 Properties

Compared with relational databases, graph databases are often faster for associative data sets and map more directly to the structure of object-oriented applications.

They can scale more naturally to large data sets as they do not typically require expensive join operations. As they depend less on a rigid schema, they are more suitable to manage ad hoc and changing data with evolving schemas. Conversely, relational databases are typically faster at performing the same operation on large numbers of data elements.

Graph databases are a powerful tool for graph-like queries, for example computing the shortest path between two nodes in the graph. Other graph-like queries can be performed over a graph database in a natural way (for example graph’s diameter computations or community detection).

## 3 Graph database projects

The following is a list of several well-known graph database projects:

## 4 Graph database features

The following table compares the features of the above graph databases.

## 5 Distributed Graph Processing

- Angrapa - graph package in Hama, a bulk synchronous parallel (BSP) platform
- Apache Hama - a pure BSP(Bulk Synchronous Parallel) computing framework on top of HDFS (Hadoop Distributed File System) for massive scientific computations such as matrix, graph and network algorithms.
- Blazegraph - A RDF/graph database capable of clustered deployment. Blazegraph supports high availability (HA) mode, embedded mode, single server mode and has available commercial licenses. As of version 1.3.1, it supports the Blueprints API and Reification Done Right (RDR).
- Cyclops - A computation and communication efficient graph processing system with significantly low communication cost.

- Faunus - a Hadoop-based graph computing framework that uses Gremlin as its query language. Faunus provides connectivity to Titan, Rexster-fronted graph databases, and to text/binary graph formats stored in HDFS. Faunus is developed by Aurelius.
- FlockDB - an open source distributed, fault-tolerant graph database based on MySQL and the Gizzard framework for managing Twitter-like graph data (single-hop relationships).
- Giraph - a Graph processing infrastructure that runs on Hadoop (see Pregel).
- GraphBase - Enterprise Edition supports embedding of callable Java Agents within the vertices of a distributed graph.
- Graph Engine - A free distributed, in-memory, large graph processing engine, previously known as Trinity.
- GoldenOrb - Pregel implementation built on top of Apache Hadoop
- GraphLab - A framework for machine learning and data mining in the cloud
- GraphX - GraphLab built on the Spark cluster computing system. Dr. Joseph Gonzalez is the project lead, the creator of GraphLab.
- HipG - a library for high-level parallel processing of large-scale graphs. HipG is implemented in Java and is designed for distributed-memory machine
- IBM System G Graph Analytics Toolkit - A comprehensive graph analytics library consisted of network topological analysis tools, graph matching and search tools, and graph path and flow tools. It has been applied to various use cases and industry solutions.
- Imitator - A reliable distributed graph processing system with replication-based fault-tolerance.
- InfiniteGraph - a commercially available distributed graph database that supports parallel load and parallel queries.
- JPreGel - In-memory java based Pregel implementation
- Lionsgate - A distributed, browser-based graph database developed by Semblent.
- KDT - An open-source distributed graph library with a Python front-end and C++/MPI backend.
- Mizan - An optimized Pregel clone that can be deployed easily on Amazon EC2, local clusters, stand-alone Linux systems and supercomputers (IBM BlueGene/P). It utilizes runtime graph repartitioning between iterations to provide dynamic load balancing for better algorithm performance.<sup>[11]</sup>
- OpenLink Virtuoso - the shared-nothing Cluster Edition supports distributed graph data processing.
- Oracle Spatial and Graph - loading, inferencing, and querying workloads are automatically and transparently distributed across the nodes in an Oracle Real Application Cluster, Oracle Exadata Database Machine, and Oracle Database Appliance.
- Phoebus - Pregel implementation written in Erlang
- Pregel - Google's internal graph processing platform, released details in ACM paper.
- Powergraph - Distributed graph-parallel computation on natural graphs.
- PowerLyra - A distributed graph analytics based on GraphLab using differentiated graph computation and partitioning on skewed (e.g. power-law and bipartite) graphs (dynamically applying different computation and partition strategies for different vertices).
- PowerSwitch - Adaptive Prediction and Mode Switch (Sync & Async) on Graph-parallel Computation
- Sedge - A framework for distributed large graph processing and graph partition management (including an open source version of Google's Pregel)
- Signal/Collect - a framework for parallel graph processing written in Scala
- Sqrrl Enterprise - distributed graph processing utilizing Apache Accumulo and featuring cell-level security, massive scalability, and JSON support
- Titan - A distributed, disk-based graph database developed by Aurelius.
- Parallel Boost Graph Library (PBGL) - a C++ library for graph processing on distributed machines, part of Boost framework.
- Weaver - A fast and scalable graph store designed specifically for dynamically-changing graphs.

## 6 GPGPU Graph Processing

- Medusa - A framework for graph processing using Graphics Processing Units (GPUs) on both shared memory and distributed environments. Medusa allows users with no GPU programming expertise to leverage GPUs for graph processing.

## 7 APIs and Graph Query/Programming Languages

- Bounds Language - terse C-style syntax which initiates concurrent traversals in GraphBase and supports interaction between them.
- Blueprints - a Java API for Property Graphs from TinkerPop and supported by a few graph database vendors.
- Blueprints.NET - a C#/.NET API for generic Property Graphs.
- Bulbflow - a Python persistence framework for Rexster, Titan, and Neo4j Server.
- Cypher Query Language - a declarative graph query language for Neo4j that enables ad hoc as well as programmatic (SQL-like) access to the graph
- Dave - a declarative graph query language for Semblent - Lionsgate
- Gremlin - an open-source graph programming language that works over various graph database systems.
- Neo4jClient - a .NET client for accessing Neo4j.
- Neography - a thin Ruby wrapper that provides access to Neo4j via REST.
- Neo4jPHP - a PHP library wrapping the Neo4j graph database.
- NodeNeo4j - a Node.js driver for Neo4j that provides access to Neo4j via REST
- Pacer - a Ruby dialect/implementation of the Gremlin graph traversal language.
- Pipes - a lazy dataflow framework written in Java that forms the foundation for various property graph traversal languages.
- Pixy - a declarative graph query language that works on any Blueprints-compatible graph database
- PYBlueprints - a Python API for Property Graphs.
- Pygr - a Python API for large-scale analysis of biological sequences and genomes, with alignments represented as graphs.
- Rexster - a graph database server that provides a REST or binary protocol API (RexPro). Supports Titan, Neo4j, OrientDB, Dex, and any TinkerPop/Blueprints-enabled graph.
- RDFSharp - a .NET API for modeling RDF graphs, storing them on many SQL databases (Firebird, MySQL, PostgreSQL, SQL Server, SQLite) and querying them with SPARQL.

- SPARQL - a query language for databases, able to retrieve and manipulate data stored in Resource Description Framework format.
- SPASQL - an extension of the SQL standard, allowing execution of SPARQL queries within SQL statements, typically by treating them as subquery or function clauses. This also allows SPARQL queries to be issued through “traditional” data access APIs (ODBC, JDBC, OLE DB, ADO.NET, etc.)
- Spring Data Neo4j - an extension to Spring Data (part of the Spring Framework), providing direct/native access to Neo4j
- Oracle SQL and PL/SQL APIs] - have graph extensions for Oracle Spatial and Graph.
- Styx (previously named Pipes.Net) - a data flow framework for C#/.NET for processing generic graphs and Property Graphs.
- Thunderdome - a Titan Rexster Object-Graph Mapper for Python (no longer maintained)
- Mogwai - a Titan Rexster Object-Graph Mapper for Python - Forked from Thunderdome
- Rexpython - a Titan Rexpython connection handler for Python.

## 8 See also

- NoSQL
- Document-oriented database
- Structured storage
- Object database
- Resource Description Framework (RDF), framework to express node-edge graphs
- Graph transformation for a complementary topic (rule based in memory manipulation of graphs instead of transaction safe persistence).
- RDF Database

## 9 References

- [1] <http://brightstardb.com/blog/2013/02/brightstardb-goes-open-source/>
- [2] <http://sparsity-technologies.com/#sparksee>
- [3] <http://www.graphengine.io>
- [4] <http://infogrid.org/wiki/Docs/License>

- [5] MapGraph: A High Level API for Fast Development of High Performance Graph Analytics on GPUs Retrieved Aug 7, 2014.
- [6] Parallel Breadth First Search on GPU Clusters Retrieved Aug 7, 2014.
- [7] DB-Engines Ranking of Graph DBMS. Retrieved July 19, 2013.
- [8] OpenLink Software. "Clustering Deployment Architecture Diagrams for Virtuoso (Release 6 and later, Commercial Edition only)". *Virtuoso Open-Source Wiki*. OpenLink Software. Retrieved 2014-05-01.
- [9] <http://sparsity-technologies.com#sparksee>
- [10] <http://grid.hust.edu.cn/triplebit/>
- [11] <http://dl.acm.org/citation.cfm?id=2465369>

## 10 Text and image sources, contributors, and licenses

### 10.1 Text

- **Graph database** *Source:* [https://en.wikipedia.org/wiki/Graph\\_database?oldid=674061526](https://en.wikipedia.org/wiki/Graph_database?oldid=674061526) *Contributors:* Jonik, Jni, Giftlite, JimD, Kaldari, Stesmo, Gary, Diego Moya, Sbrunner, MacTed, BD2412, Josh Parris, A.K.Karthikeyan, Intgr, Morpvh, SamJohnston, Pelister, Egbert J. van der Haring, Cedar101, Ffangs, Frap, Russorat, E40, Lguzenda, RichMorin, ProfessorBaltasar, TempestSA, Espeed, GauravJShah, Michael A. White, Vonmoltke, Stybn, Magioladitis, Vishal0soni, Ksinker, Crasmnky, Agavenwurm, AlanUS, Drq123, Stokito, Cnorvell, Trivialist, Pombredanne, Pointillist, Triadic2000, Miami33139, Jacobito, MystBot, Addbot, Thomas888b, Jncraton, Laaknor-Bot, JakobVoss, Legobot, Aglnl, Yobot, Bunnyhop11, AnomieBOT, Fraktalek, Materialschemist, Xqbot, El33th4x0r, MuffledThud, FrescoBot, Nawroth, Sae1962, LittleWink, Binshao, Ajmagnifico, JnRouvignac, Thinxer, Magnuschr, J12t, Zelevin, Yanivby, Lillem4n, Ahzf, EmausBot, Colinnui, Bolerio, Zorclub27, Starboy8, ZéroBot, Pholding, Sberriman, Tuhl, Andrearatto, DamarisC, Thoughtpuzzle, Tgrota, 0x24a537r9, Germanviscuso, TTJDenman, Luisbargu, Katkov Yury, Danim, Pereb, Taneltammet, Aldonline, 4368a, BG19bot, Freshn-fruity, Stott.parker, Wbeaureg, RecaiAlkan, Fceller, Syhuang1988, Chmarkine, Dreamingxk89, Luebbert42, Praveensripati, Codename Lisa, Shengqiyang, Electro rick, Forssam, Preillyme, Tsm32, Mhgrove, Rzicari, Lesser Cartographies, Kiryakov ak, MoSarwat, Ppr15, Lambdazen, Elykahn1, Jmesney, Khayyatzy, Dindin89, Yinglongxia, Applelittle, Brbebee, Platinummonkey, Azanebrain, Caffeinewriter, Jsaarela, Endyke, Danistrebel, Nachivpn, Ktraison and Anonymous: 135

### 10.2 Images

- **File:GraphDatabase\_PropertyGraph.png** *Source:* [https://upload.wikimedia.org/wikipedia/commons/3/3a/GraphDatabase\\_PropertyGraph.png](https://upload.wikimedia.org/wikipedia/commons/3/3a/GraphDatabase_PropertyGraph.png) *License:* CC0 *Contributors:* Originally uploaded on en.wikipedia *Original artist:* Originally uploaded by Ahzf (Transferred by Obersachse)

### 10.3 Content license

- Creative Commons Attribution-Share Alike 3.0