



Nicht-Standard Datenmodelle und Datenbanken

1. Einführung
2. RDF und Semantic Web
3. **Graphdatenbanken**

Not only SQL

Relationale Datenbanken sind für große Datenmengen **oder** viele gleichzeitige Zugriffe optimiert. Das Internet hat den Bedarf an Big Data Lösungen in dem die Datenmenge groß **und** die Anzahl der Zugriffe hoch ist

- **Nicht-Relational:** Keine Tabellenstruktur, ermöglicht agile Verarbeitung großer Datenmengen.
- **Verschiedene Datenmodelle:** Umfasst Key-Value-Stores, Dokumenten-Datenbanken, Wide-Column-Stores und Graphdatenbanken.
- **Skalierbarkeit:** Hoch skalierbar, geeignet für Verteilung auf viele Server.
- **Flexibilität:** Schema-los, bietet Flexibilität bei Datenspeicherung und -manipulation.
- **Leistung:** Schnelle Schreib- und Lesevorgänge, ideal für Echtzeitanwendungen.
- **Big Data & Echtzeit-Web-Apps:** Ideal für Big Data und schnelle Webdienste.
- **Konsistenzmodell:** Nutzt "eventuelle Konsistenz" für hohe Verfügbarkeit.

NoSQL: Datenbank-Typen

Key-Value-Store:

Vereinfachte Datenstruktur, die Daten als Schlüssel-Wert-Paare speichert.
Ideal für Anwendungen, die schnelle Lese- und Schreibzugriffe benötigen.
Beispiele: Redis, DynamoDB.

Dokumenten-Datenbanken:

Speichert Daten als "Dokumente", meist im JSON-Format.
Flexible Schemata, einfach hinzuzufügen und zu ändern.
Ideal für Anwendungen, die komplexe, hierarchische Datenstrukturen verwalten.
Beispiele: MongoDB, CouchDB.

Wide-Column-Stores:

Speichern Daten in Spalten statt in Zeilen, gut für die Analyse großer Datenmengen.
Hoch skalierbar und effizient bei der Speicherung großer Datenmengen.
Beispiele: Cassandra, HBase.

Graphdatenbanken:

Verwenden Graphenstrukturen mit Knoten, Kanten und Eigenschaften, um relationale Daten darzustellen.
Ideal für Anwendungen, die komplexe Beziehungen und Verbindungen analysieren.
Beispiele: Neo4j, Blazegraph, Amazon Neptune -> Gremlin/ApacheTinkerpop

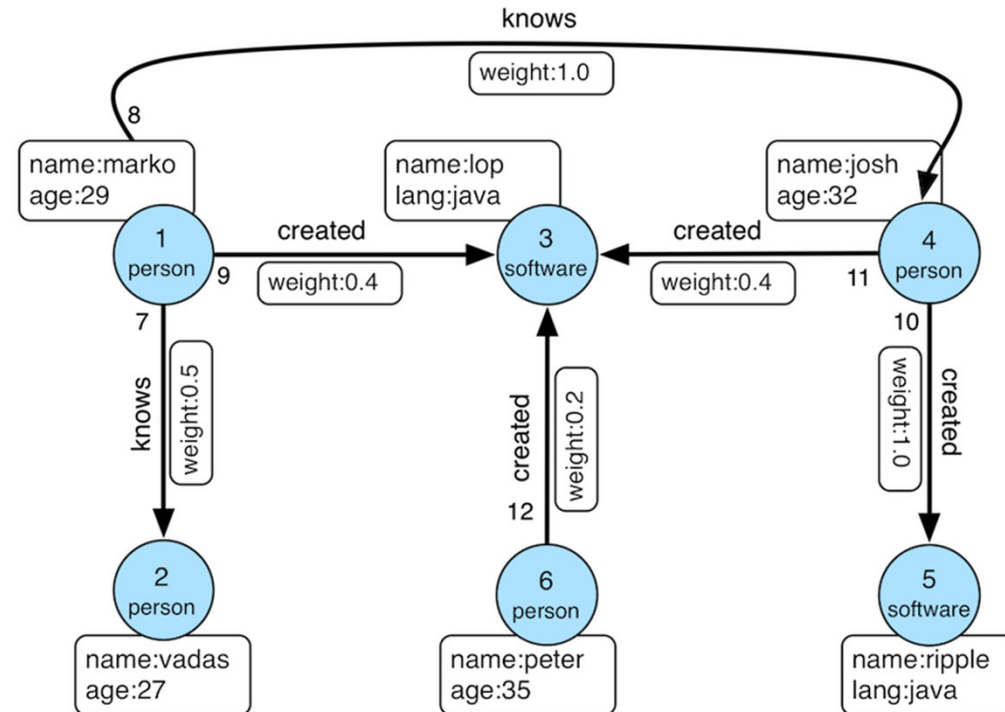
Datenmodell: Property Graph

Knoten

- Repräsentiert eine *Entity*
- Hat 0 oder mehr *Label*
- Hat 0 oder mehr *Properties*

Kanten

- Strukturieren den Graphen (semantischer Kontext)
- Haben einen Typ
- Haben 0 oder mehr *Properties*
- Setzen Knoten zueinander in Relation
- Haben einen Start und Ende-Knoten



Eigenschaften

- Key-Value Paare
- Repräsentieren Daten
- Struktur „String key; typisierter Wert“

This is a preview - click here to buy the full publication

**INTERNATIONAL
STANDARD**

**ISO/IEC
9075-16**

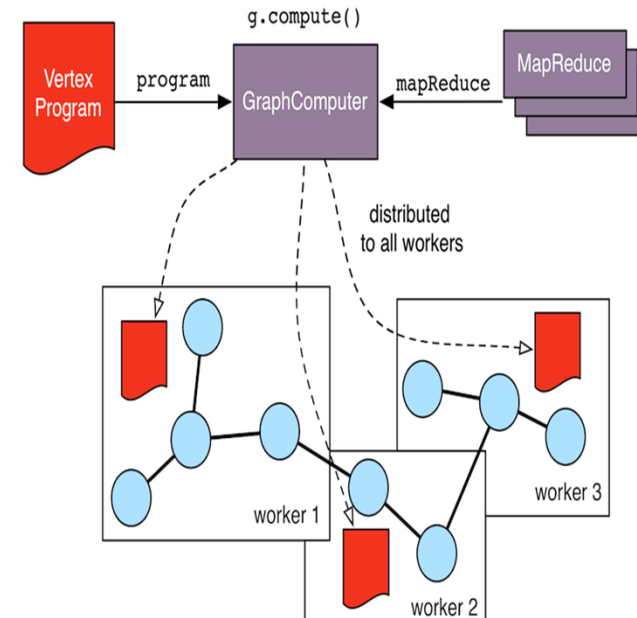
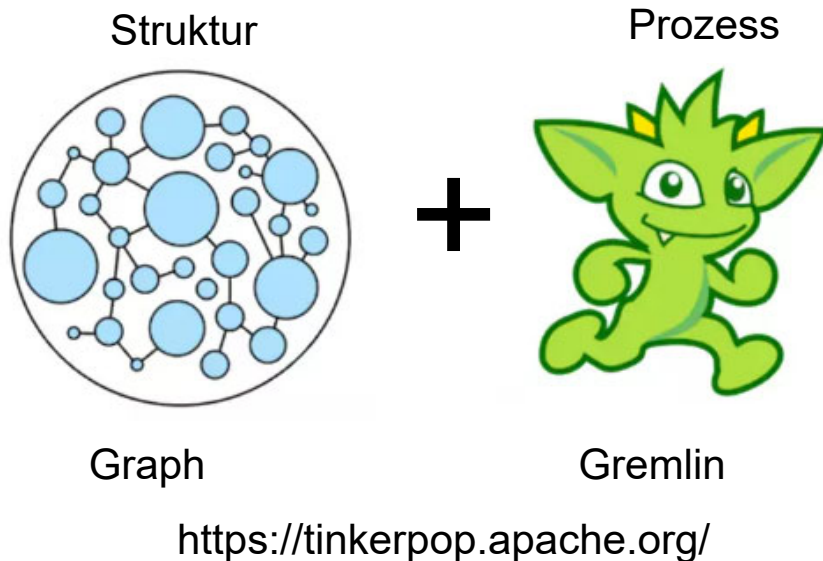
First edition
2023-06

**Information technology — Database
languages SQL —**

Part 16:
Property Graph Queries (SQL/PGQ)

Apache Tinkerpop/Gremlin

- Open Source Graph Computing Framework
- Gremlin ist die Graph-Verarbeitungssprache



Vertex-Programm: Berechnung auf jedem Knoten (Vertex)

GraphComputer: Steuert Ausführung von Vertex-Programmen

MapReduce: "Map" führt Berechnungen auf Knotenebene aus, und "Reduce" aggregiert diese Ergebnisse

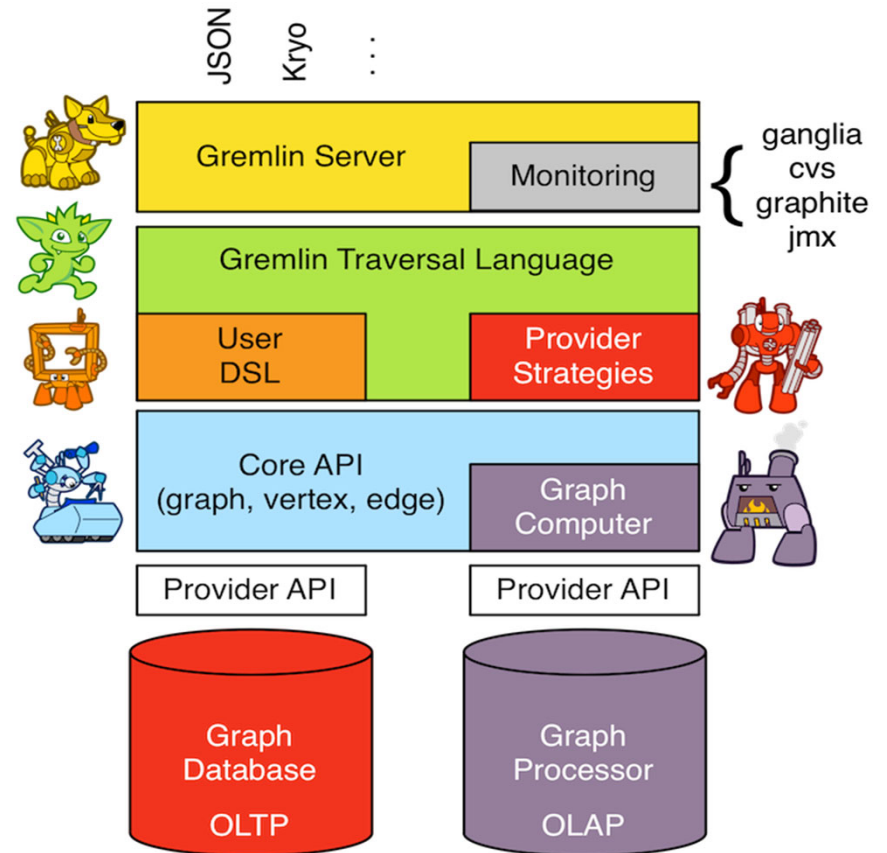
Apache Tinkerpop/Gremlin Architektur - sowohl für OLAP als auch OLTP geeignet

OLTP (Online Transaction Processing):

- Echtzeit-Transaktionsbedürfnisse
- optimiert für schnelle und zuverlässige Verarbeitung kleiner Datenmengen.
- Bankwesen oder Einzelhandelsverkäufe.

























OLAP (Online Analytical Processing):

- Systeme für Datenanalyse konzipiert,
- optimiert um komplexe Abfragen auf großen Datenmengen zu bewältigen.
- Verwendet in Business Intelligence und Data Warehousing
- Schwerpunkt auf Datenaggregation, multidimensionaler Analyse

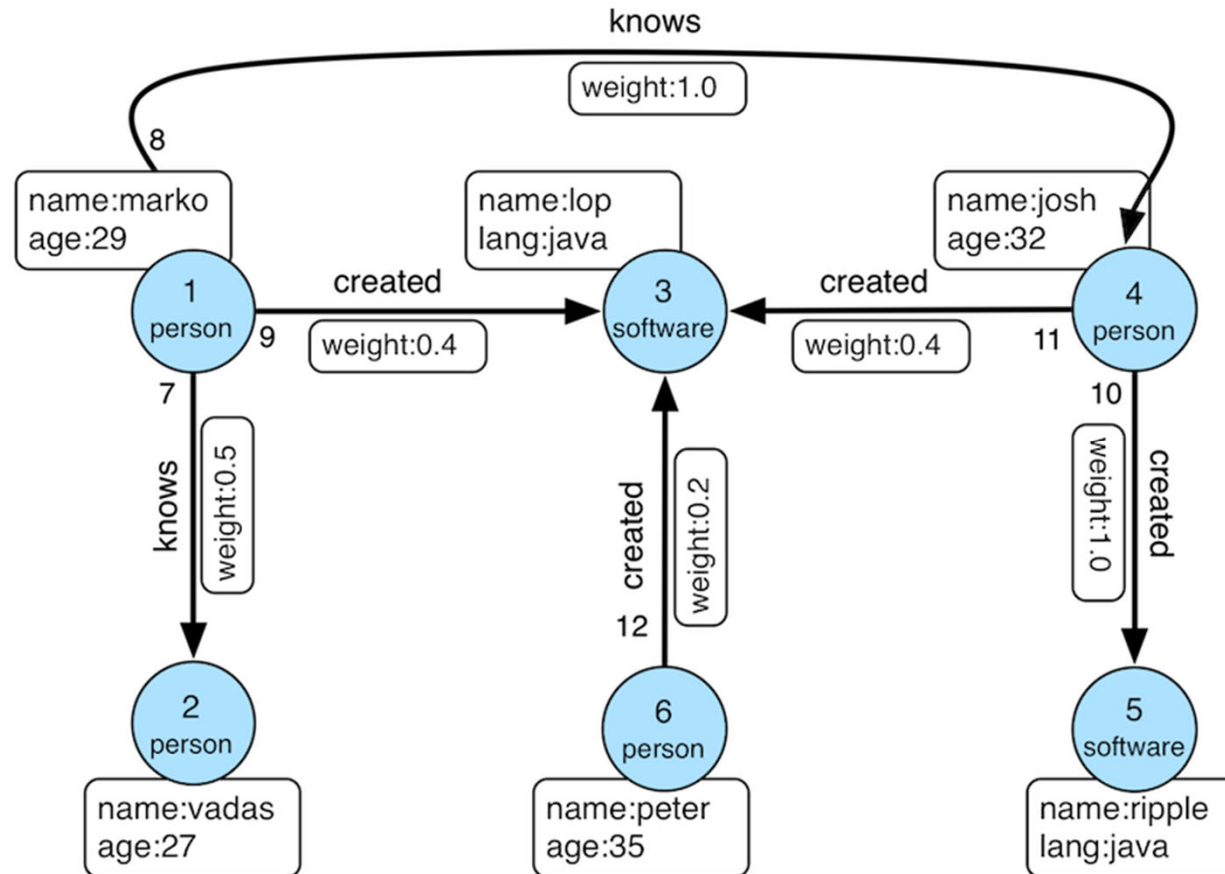


Apache Tinkerpop/Gremlin

- <https://tinkerpop.apache.org/providers.html>

 Alibaba Graph Database A real-time, reliable, cloud-native graph database service that supports property graph model.	 Amazon Neptune Fully-managed graph database service.	 ArcadeDB Multi-Model Database Supporting Graphs, Key/Value, Documents and Time-Series.	 HugeGraph A high-speed, distributed and scalable OLTP and OLAP graph database with visualized analytics platform.	 IBM Db2 Graph Graph database using IBM® Db2®.	 JanusGraph Distributed OLTP and OLAP graph database with BerkeleyDB, Apache Cassandra and Apache HBase support.
 ArangoDB OLTP Provider for ArangoDB.	 Bitsy A small, fast, embeddable, durable in-memory graph database.	 Blazegraph RDF graph database with OLTP support.	 Neo4j OLTP graph database (embedded and high availability).	 OrientDB OLTP graph database	 OverflowDB In-memory graph database with low memory footprint
 CosmosDB Microsoft's distributed OLTP graph database.	 ChronoGraph A versioned graph database.	 DSEGraph DataStax graph database with OLTP and OLAP support.	 Apache S2Graph OLTP graph database running on Apache HBase.	 Sqlg OLTP implementation on SQL databases.	 Stardog RDF graph database with OLTP and OLAP support.
 Hadoop (Spark) OLAP graph processor using Spark.	 HGraphDB OLTP graph database running on Apache HBase.	 Huawei Graph Engine Service Fully-managed, distributed, at-scale graph query and analysis service that provides a visualized interactive analytics platform.	 Tibco Graph Database Combined OLTP and OLAP features in a single enterprise-grade database.	 TinkerGraph In-memory OLTP and OLAP reference implementation.	 Unipop OLTP Elasticsearch and JDBC backed graph.

Der apache tinkertop „modern“ Beispiel Graph



Graph Definition

$G=(V,E)$

V=Menge der Vertices (Knoten)

E=Menge Edges (Kanten)

Grundprinzip: Graph Transversals

Graph Traversal

