



ORACLE®

RDF Graph for Oracle NoSQL Database EE

Agenda

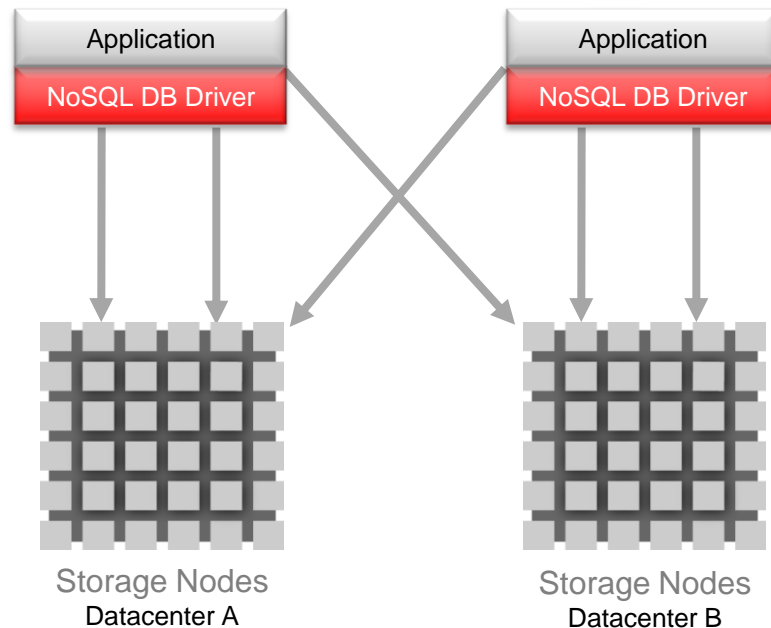
- Oracle NoSQL Database Enterprise Edition overview
- Value of a graph database
- RDF Graph for Oracle NoSQL Database Enterprise Edition
- Feature overview

Oracle NoSQL Database Enterprise Edition

Scalable, Highly Available, Key-Value Database

Features

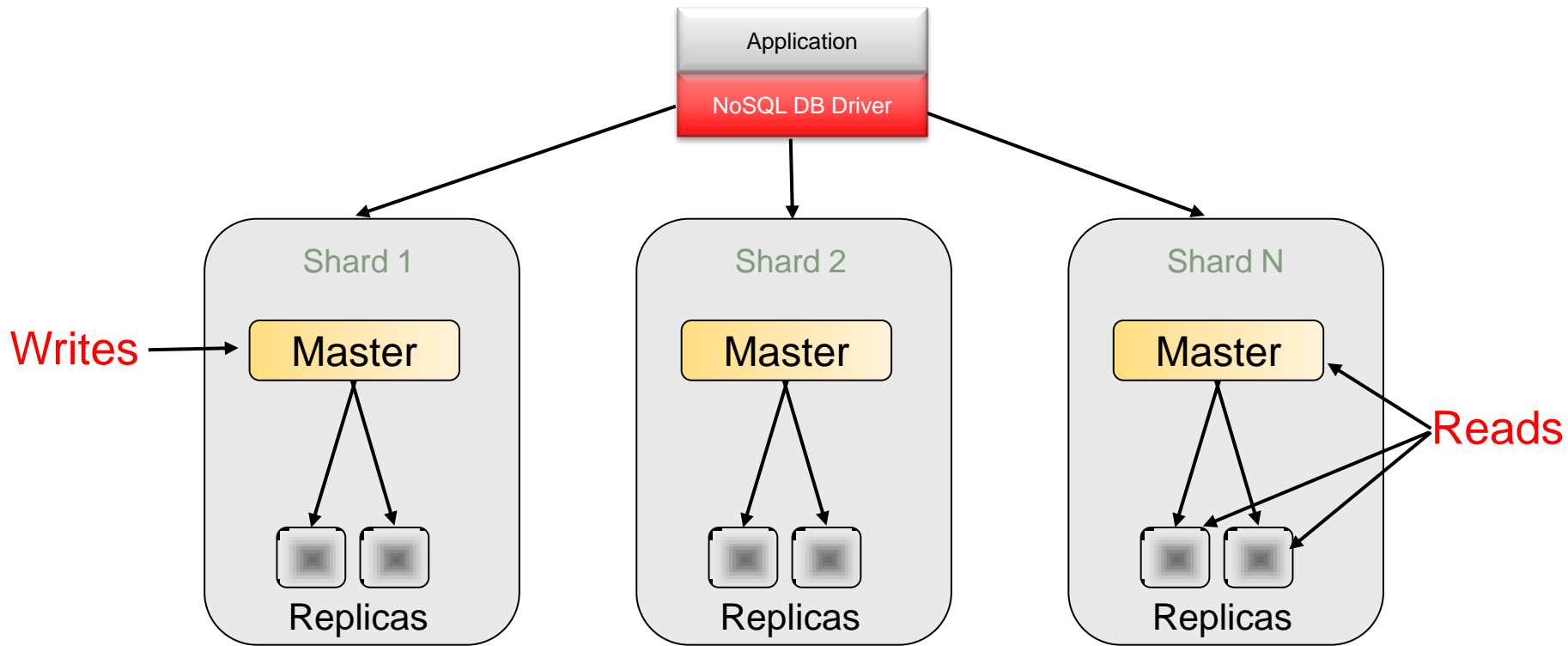
- Flexible Key-Value Data Model
- ACID transactions
- Horizontally Scalable
- Highly Available
- Elastic Configuration
- Simple administration
- Intelligent Driver
- Commercial grade software and support



Java SE 6 (JDK 1.6.0 u25)+; Solaris or Linux

ORACLE

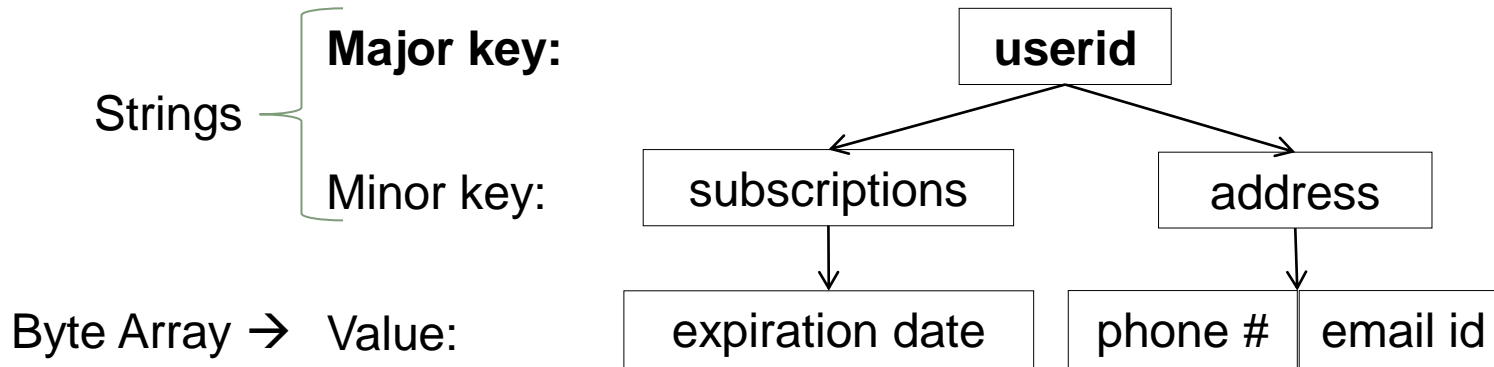
Logical Architecture – Application's view



Flexible Data Model

Key-value pairs

- Data model – key-value pair (major+minor-key paradigm)
- Simple operations – read/insert/update/delete, read-modify-write
- Scope of transaction – records within a major key, single API call
- Unordered scan of all data (non-transactional)



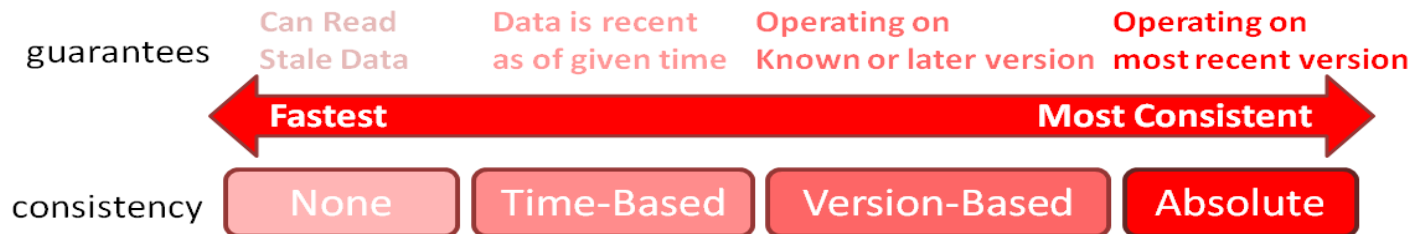
Transaction Durability and Read consistency

ACID Transactions – Configurability

- Configurable **D**urability Policy



- Configurable **C**onsistency Policy



Oracle NoSQL Database Differentiation

Integrates seamlessly with Oracle Stack (Oracle Database, ODI, OLH, CEP, RDF)

Commercial Grade Software and Support

- **General Purpose**
- **Reliable** – Based on proven Berkeley DB JE HA
- **Easy** to Install & Configure

Scalability and Availability

- **Intelligent** Oracle NoSQL DB Driver
 - Evenly distributes data
 - Ops go to fastest node
 - Bounded network hops for all operations
- **Automatic** replication and failover
- **1M+** Operations/second

Flexible Data Model

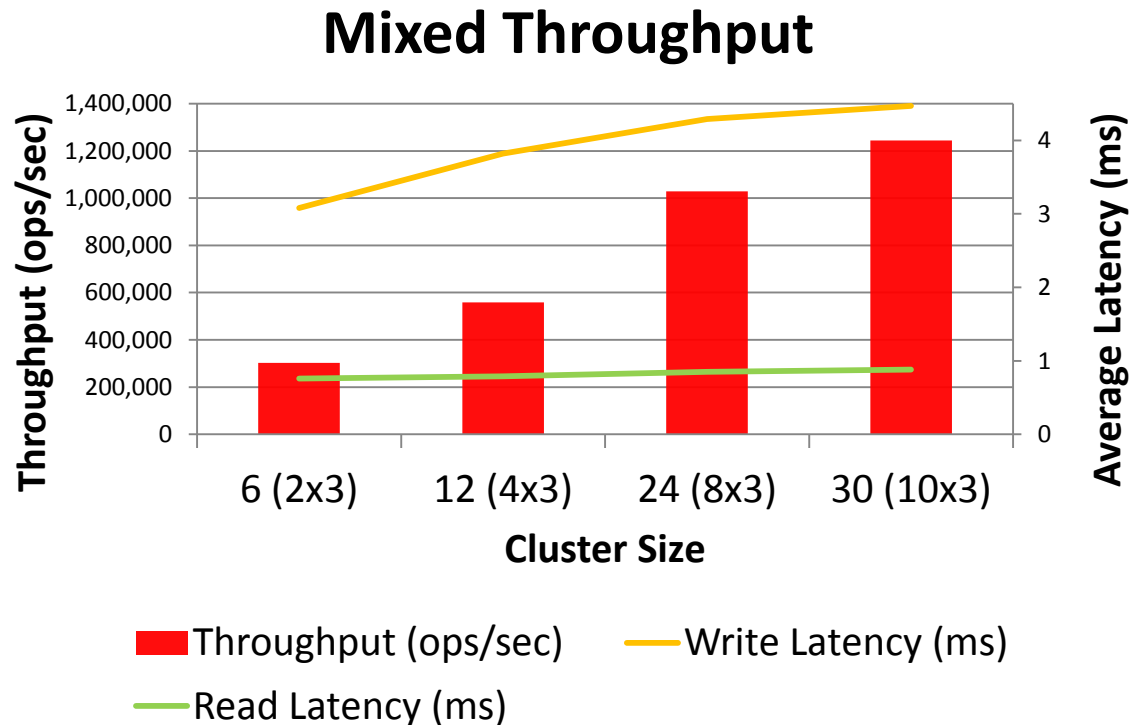
- **Flexible** Major + Minor Key-Value data structure
- **JSON** schemas
- **ACID** transactions
- **Configurable** consistency and durability

Simple Administration

- **Web-based** Console and CLI commands
- **Smart Topology** **Manages and Monitors**
 - Topology
 - Load & Performance
 - Events & Alerts
- **JMX & SNMP** Integration

Latest YCSB Benchmark Results

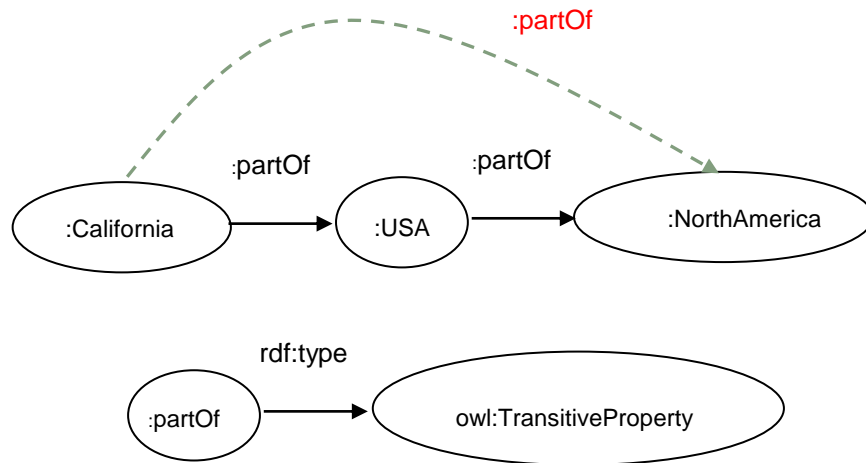
- 1.25M ops/sec
- 2 billion records
- 2 TB of data
- 95% read, 5% update
- Low latency
- High Scalability



Why a Graph Database?

Graph Database

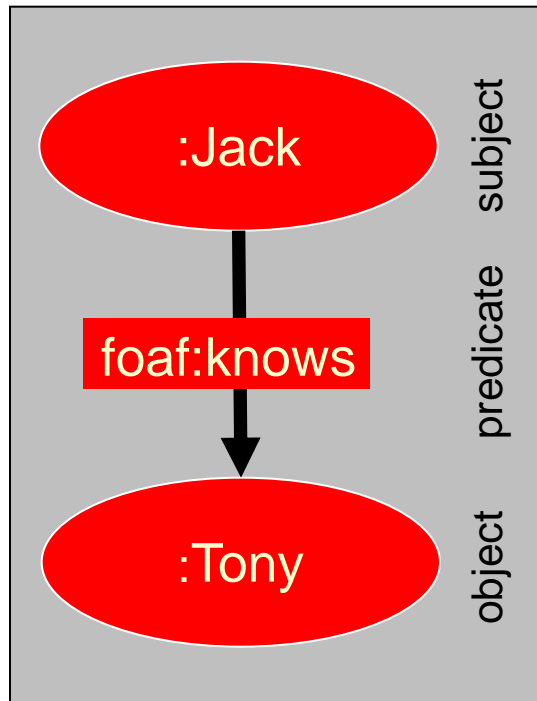
- Model data in terms of relationships
- Flexible schema evolves easily by adding new relationships
- Supports querying and discovery by graph patterns and traversal
- Enables graph analytics such as reachability, connectivity, transitivity, same as, proximity, centrality...



Query: `SELECT ?x ?y`
`FROM ...`
`WHERE { ?x :partOf ?y }`

What is RDF?

Resource Description Framework



- Basic structure is a “triple”
 - [subject] → [predicate] → [object]
- Triples are connected to form a graph
 - Graphs facilitate analysis / discovery of relationships
- RDF can be serialized into XML
 - Namespaces, for example
- Schemas need not be specified in advance
 - No schema changes to cope with foreign vocabularies
- RDF data is fully expressible as RDBMS data
 - RDBMS data is also fully expressible as RDF

RDF: Key ideas

- Based on fundamentally different Open World Assumption
 - What is unknown is undefined (not false) - that supports discovery
- Schema are flexible, evolving, can't be known in advance
 - Rich, real world relationships are modeled in the data
- Every data element is uniquely identified - supports integration
 - Data & relationships are machine-readable
- Pattern query language supports discovery workflows
- Enhance query and discovery with a standard set of related concepts that describes the relationships (an ontology)
 - Find implied relationships using rules



MANAGING RDF GRAPH DATA IN ORACLE NOSQL DATABASE EE

RDF Graph for NoSQL Database Enterprise Edition



- W3C standards compliance
- Horizontally scalable graph operations
- Develop with Apache Jena open source Java APIs for load & query
- Query with Apache Jena Joseki SPARQL end point web services
- Inference with Apache Jena & open source reasoners
- Use tools for query, visualization, and ontology engineering from open source & commercial 3rd parties with Apache Jena

Key Capabilities:

Load / Storage

- RDF data on key/value store
- ACID & BASE consistency
- Fast distributed load

Query

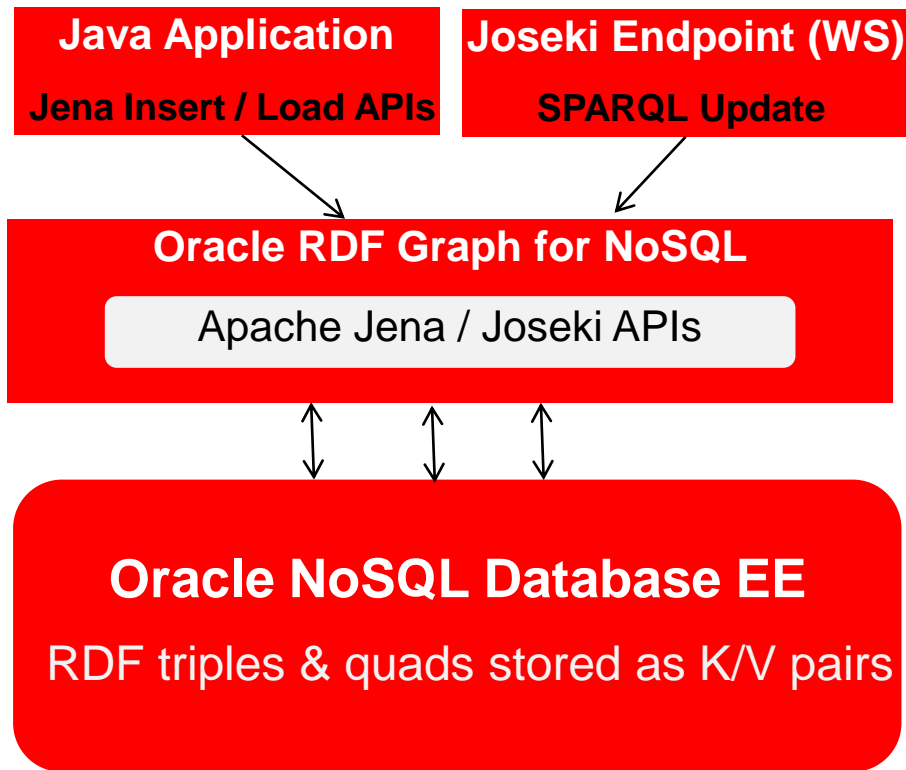
- SPARQL 1.1 Query, Update, Construct, Named graph
- Apache Jena Java APIs
- Apache Joseki SPARQL end point

Reasoning

- W3C RDFS and OWL
- Plug-in architecture

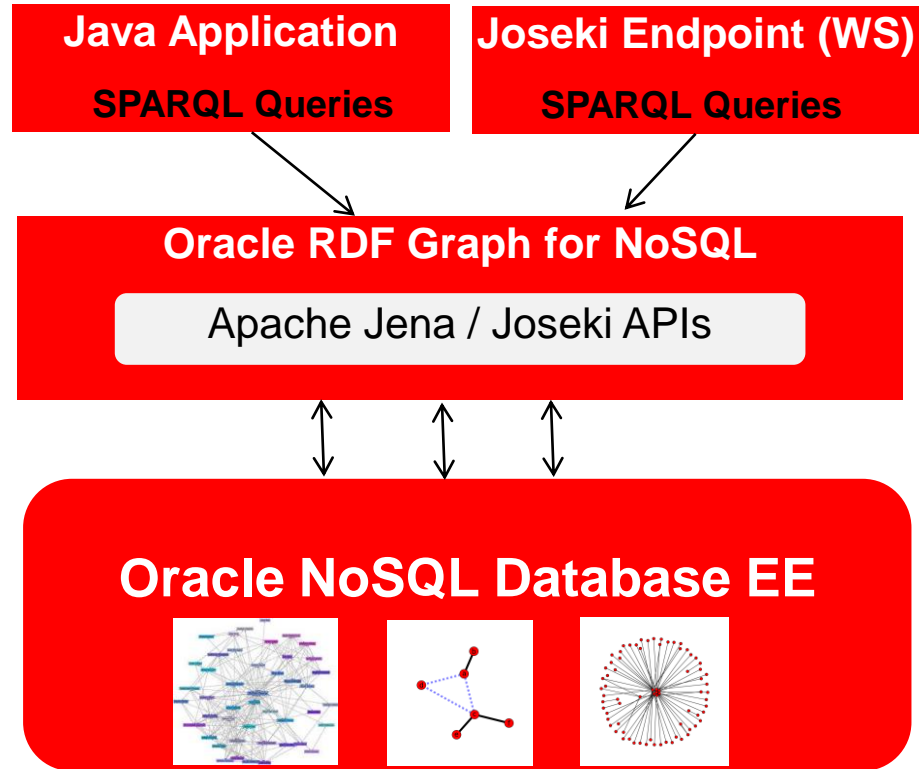
Loading RDF data in a NoSQL Graph

- Default & named graphs
- Quad format associates a triple with a named graph
- Parallel load
- Apache Jena Insert API
- Apache Jena Load API to bulk load an RDF file
- JSON (JavaScript Object Notation) data-interchange format



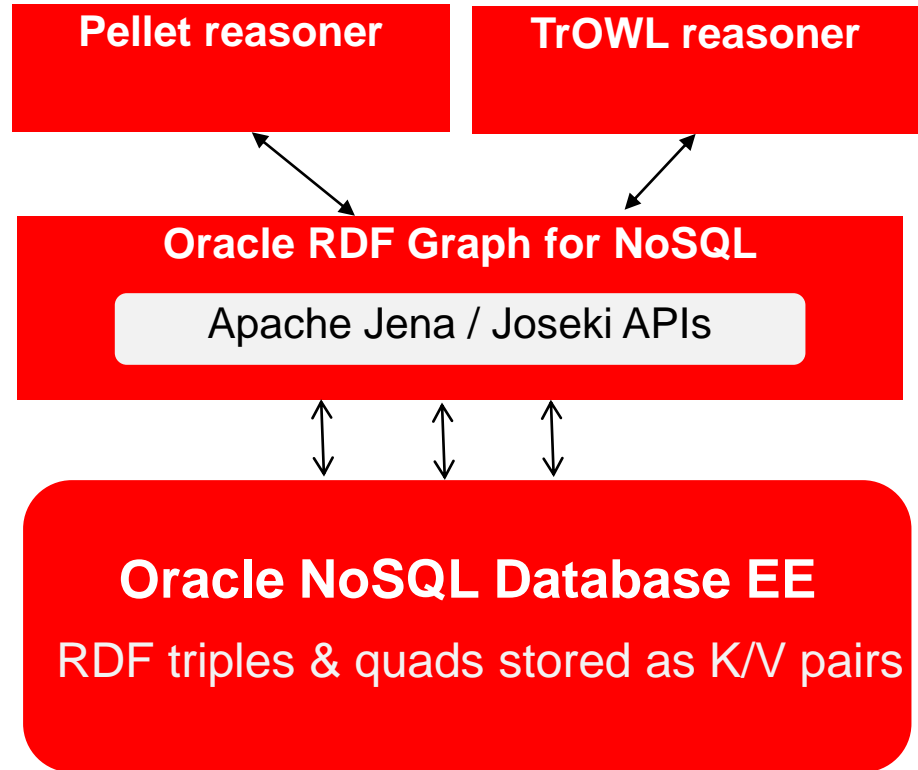
Querying RDF Graphs in NoSQL Database

- W3C SPARQL 1.1 support
- Multi-graph queries
- Parallel query
- SPARQL Update
- SPARQL Construct
- Query execution planning for optimal performance
- Joseki SPARQL endpoint for Linked Open Data services



OWL 2 Inferencing on a NoSQL Graph

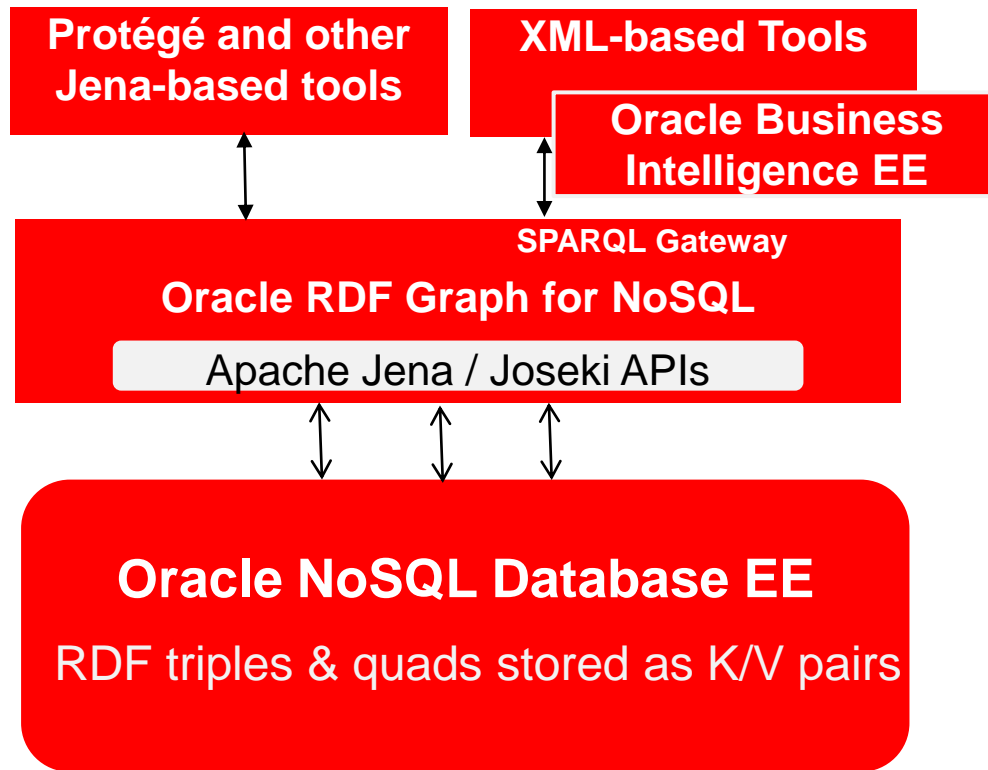
- In-memory inferencing
- Inferencing results can be queried and/or stored
- Apache Jena OntModel APIs
- Open-source Pellet reasoner
- Open-source TrOWL reasoner



Tools for Visualization, Editing and Analysis

Apache Jena and XML based:

- Oracle tools
- Third-party tools
- Open source tools
 - Visualization
 - Editing
 - analysis



Managing Graph data with Oracle NoSQL: demo

Inserting triples in a default graph



The screenshot shows a web browser window with the address bar displaying `adc2201652:8080/joseki-nosql272/update.html`. The page title is "Oracle SPARQL Update Service Endpoint using Joseki". Below the title, there is a text input area with the placeholder "Type in your SPARQL/Update request". The input area contains the following SPARQL update query:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
INSERT DATA
{ <http://example/book3> dc:title "A new book" ;
  dc:creator "A.N.Other" .
}
```

Below the input area, there is a button labeled "Perform SPARQL Update".

Managing Graph data with Oracle NoSQL: demo

Creating a named graph and inserting triples into named graph

The screenshot shows a web browser window with the address bar displaying 'adc2201652:8080/joseki-nosql272/update.html'. The page title is 'Oracle SPARQL Update Service Endpoint using Joseki'. Below the title, there is a text input area with the prompt 'Type in your SPARQL/Update request'. The input area contains the following SPARQL/Update query:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

CREATE GRAPH <http://example/bookStore>

INSERT DATA INTO <http://example/bookStore>
{ <http://example/book3>  dc:title  "Fundamentals
of Compiler Desing"
}
```

Below the input area, there is a button labeled 'Perform SPARQL Update'.

Managing Graph data with Oracle NoSQL: demo

Loading an RDF file into a specified graph (named graph or default)



The screenshot shows a web browser window with the address bar displaying `adc2201652:8080/joseki-nosql272/update.html`. The page title is "Oracle SPARQL Update Service Endpoint using Joseki". Below the title, there is a text input area with the placeholder "Type in your SPARQL/Update request". Inside this area, the following SPARQL update query is entered:


```
PREFIX host: <http://adc2201652:8080/joseki>

LOAD host:family.rdf INTO <http://example/family>
```

Below the text input area, there is a button labeled "Perform SPARQL Update".

Managing Graph data with Oracle NoSQL: demo

Modifying graph (triples) data



The screenshot shows a web browser window with the address bar displaying `adc2201652:8080/joseki-nosql272/update.html`. The page title is "Oracle SPARQL Update Service Endpoint using Joseki". Below the title, there is a text input area with the placeholder "Type in your SPARQL/Update request". The input area contains the following SPARQL update query:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

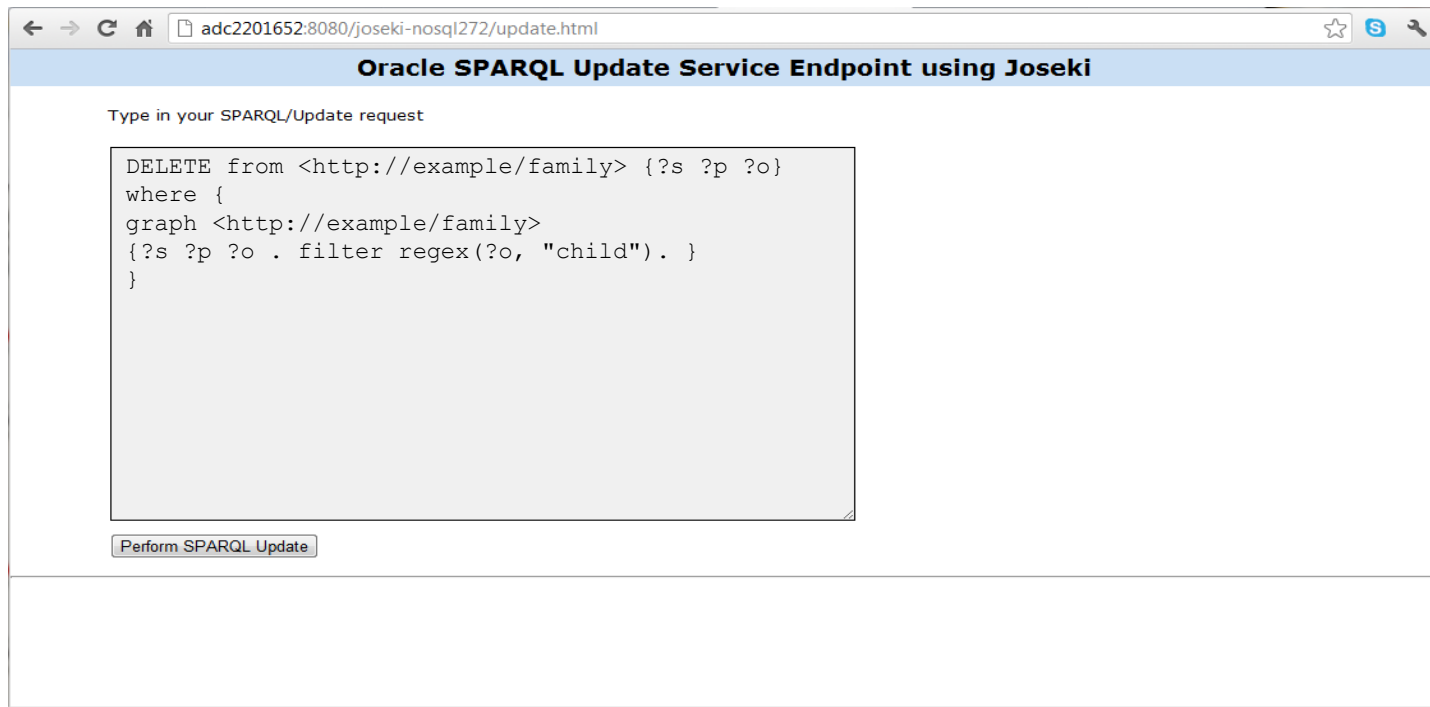
DELETE DATA FROM <http://example/bookStore>
{ <http://example/book3> dc:title "Fundamentals of
Compiler Desing" }

INSERT DATA INTO <http://example/bookStore>
{ <http://example/book3> dc:title "Fundamentals of
Compiler Design" }
```

Below the input area, there is a button labeled "Perform SPARQL Update".

Managing Graph data with Oracle NoSQL: demo

Deleting triples from a specified graph (named graph or default)



The screenshot shows a web browser window with the address bar displaying `adc2201652:8080/joseki-nosql272/update.html`. The page title is "Oracle SPARQL Update Service Endpoint using Joseki". Below the title, there is a text input area with the placeholder "Type in your SPARQL/Update request". Inside this area, a SPARQL DELETE query is entered: `DELETE from <http://example/family> {?s ?p ?o} where { graph <http://example/family> {?s ?p ?o . filter regex(?o, "child"). } }`. Below the input area is a button labeled "Perform SPARQL Update".

← → ↺ ⬆️ `adc2201652:8080/joseki-nosql272/update.html` ☆ S 🔧

Oracle SPARQL Update Service Endpoint using Joseki

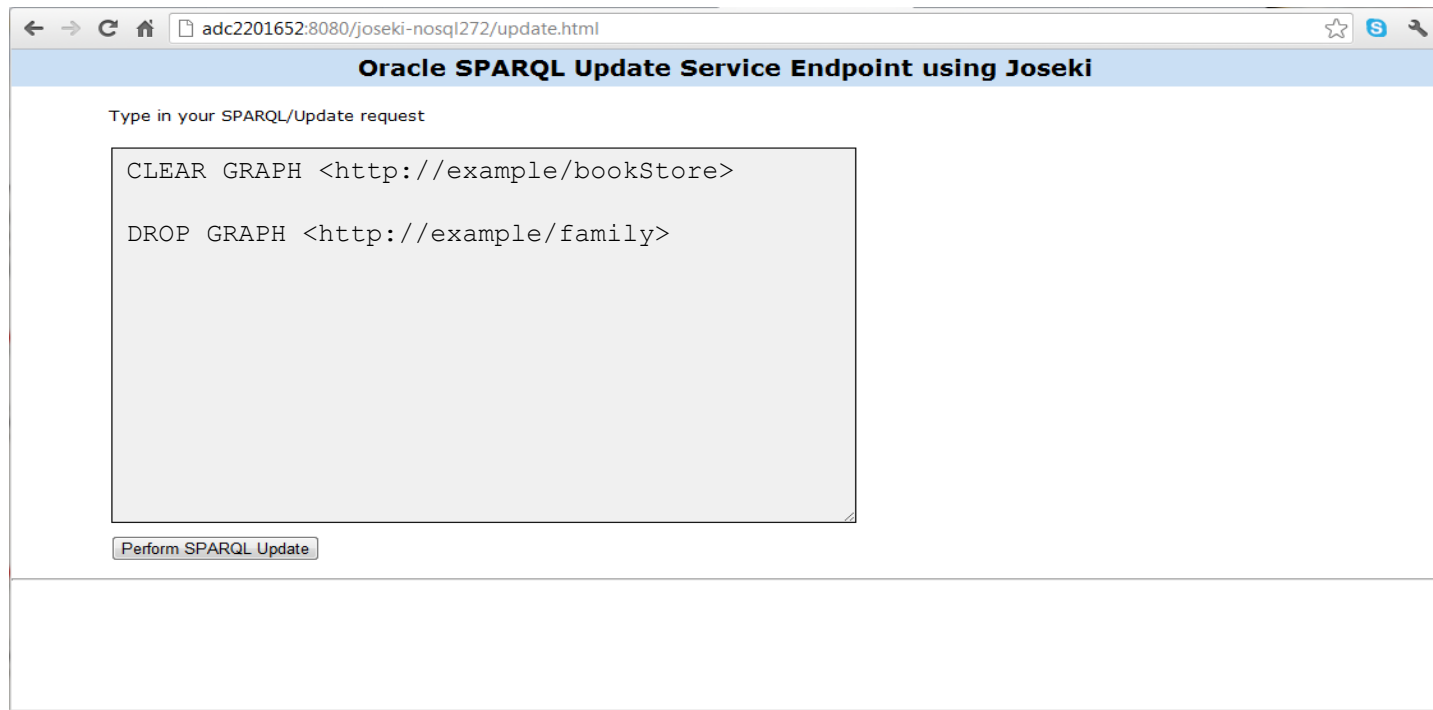
Type in your SPARQL/Update request

```
DELETE from <http://example/family> {?s ?p ?o}
where {
  graph <http://example/family>
  {?s ?p ?o . filter regex(?o, "child"). }
}
```

Perform SPARQL Update

Managing Graph data with Oracle NoSQL: demo

Clearing/removing a named graph



The screenshot shows a web browser window with the address bar displaying `adc2201652:8080/joseki-nosql272/update.html`. The page title is "Oracle SPARQL Update Service Endpoint using Joseki". Below the title, there is a text input area with the placeholder "Type in your SPARQL/Update request". Inside this area, the following SPARQL update queries are entered:

```
CLEAR GRAPH <http://example/bookStore>  
  
DROP GRAPH <http://example/family>
```

Below the text input area, there is a button labeled "Perform SPARQL Update".

Summary

RDF Graph for Oracle NoSQL Database Enterprise Edition

- Standards-based: W3C RDF, SPARQL, OWL
- Store & query RDF graph data in k/v NoSQL database
- SPARQL endpoint & 3rd party tools / technologies for...
 - Visualization, querying, inferencing, ontology editing (via Jena)
- Horizontal scalability
- Use for high-volume simple queries and updates

Q&A

Hardware and Software

ORACLE®

Engineered to Work Together

ORACLE®

ORACLE®