Otávio Santana  @otaviojava
NoSQL

01 Database
02 Doesn't use structure
03 Not Transaction
04 Base
05 Five different types
Key Value

- AmazonDynamo
- AmazonS3
- Redis
- Hazelcast
### Column Family

<table>
<thead>
<tr>
<th>HBase</th>
<th>Cassandra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scylla</td>
<td>Clouddata</td>
</tr>
<tr>
<td>SimpleDb</td>
<td>DynamoDB</td>
</tr>
</tbody>
</table>

#### Row-key
- **Apollo**
  - Duty: Sun
- **Aphrodite**
  - Duty: Love, happy
- **Ares**
  - Duty: War
- **Kratos**
  - Dead Gods: 13
  - weapon: Sword

#### Columns
Document

ApacheCouchDB

MongoDB

Couchbase

{  
  "name":"Diana",
  "duty":[
    "Hunt",
    "Moon",
    "Nature"
  ],
  "siblings":{
    "Apollo":"brother"
  }
}
Graph

Neo4j

InfoGrid

Sones

HyperGraphDB

Apollo was killed by Kratos

Ares was killed by Kratos

Kratos was killed by Apollo

Apollo was killed by Ares

Sones
Multi-Model

01. OrientDB (graph, document)
02. Couchbase (key value, document)
03. ArangoDB (document, graph, key-value)
04. Elasticsearch (document, graph)
BASE vs ACID

- Basically Available
- Soft state
- Eventual consistency

- Atomicity
- Consistency
- Isolation
- Durability
CAP

MySQL
MariaDB
mongoDB
redis
cassandra
riak

Consistency
Availability
Partition Tolerance

CA
CP
AP
JPA & JDO problem for NoSQL

- Saves Async
- Async Callback
- Time to Live (TTL)
- Consistency Level
- SQL based
- Diversity in NoSQL
The Eclipse NoSQL Solution

01. Mapping API
02. Communication API
03. No lock-in
04. Divide and Conquer

Diagram:
- DAO
- Mapping
- Communication
- ARTEMIS
- DIANA
- JNoSQL
- Data Tier
- Column
- Documents
- Key
- Graph

Key features:
- No lock-in
- Divide and Conquer
Communication Issue

BaseDocument baseDocument = new BaseDocument();
baseDocument.addAttribute(name, value);

JsonObject jsonObject = JsonObject.create();
jsonObject.put(name, value);

Document document = new Document();
document.append(name, value);

JsonObject jsonObject = JsonObject.create();
jsonObject.put(name, value);

ODocument document = new ODocument("collection");
document.field(name, value);
Eclipse JNoSQL

DocumentEntity entity = DocumentEntity.of("collection");
entity.add(name, value);
Artemis

CDI Based

Diana Based

Annotation Based

Query Method

Events to insert, delete, update

Supports to Bean Validation

Configurable and Extensible
Support to

Java

JSON

XML

YAML
NoSQL Providers
Road Map

Draft and code proposal
Community Feedback
Involve NoSQL Vendors
Involve Solution Vendors
Eclipse Project
Development
JUGs/Communities
Latin America

- Paraguay, Asunción
- Chile, Santiago de Chile
- Brasil, São Paulo
- Uruguay, Montevideo
- Argentina, Buenos Aires
- Perú, Lima
- Ecuador, Quito
- Colombia, Barranquilla
- Costa Rica, San José
- Panamá, Ciudad de Panamá
- México, Ciudad de México
- Guatemala, Ciudad de Guatemala
Oracle Code SF

- **HOL5998** Eclipse JNoSQL: One API to Many NoSQL Databases - BYOL
- **DEV6043** Let’s Make Graph Databases Fun Again with Java
- **DEV6109** Jakarta EE Meets NoSQL in the Cloud Age
Specification Process

- Java EE belongs to Eclipse Foundation
- Code First
- Move Jakarta EE forwards with new specifications
- Hopefully a new namespace until Oracle One
References

Communication API
Support to Async operations
APIs

Mapping API
Bean Validation
Events
Repository
Template

Query by text
Prepared Statement

https://projects.eclipse.org/projects/technology.jnosql
https://github.com/eclipse?q=Jnosql
https://dev.eclipse.org/mailman/listinfo/jnosql-dev
https://wiki.eclipse.org/JNoSQL
Thank you