

/RavenDB

Exploration of NoSQL
database - BDNR@M.EIC
June 7th, 2022

> Ana Cruz | up201806460@up.pt
> António Bezerra | up201806854@up.pt
> Mariana Truta | up201806543@up.pt



/TABLE OF CONTENTS



/01 /RAVENDB

- > Technology overview and history.

/02 /SETUP&DATA

- > Installation and data model.

/03 /FEATURES&API

- > Main use cases and relevant features.

/04 /LIVE_DEMO

- > RavenDB in action.

/01

/RAVENDB

Technology overview and
history.

Quick overview

- **Open-source** database with free and commercial licenses.
- Version 1.0 launched in **2010**.
- **Document-based** database.
- Provides **ACID transactions across documents and clusters**.
- Available as **DBaaS** – RavenDB Cloud.



Technical aspects

- Source-code written in **C#**.
- Data is stored as **JSON** documents.
- Supports **storing large files** through attachments.
- Extensive support for **Clustering**.
- Queries written in **own query language, RQL**.



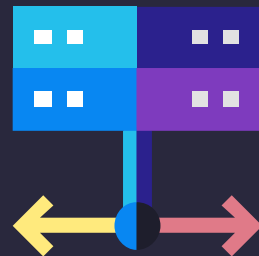
/02

/SETUP&DATA

Installation and data model.

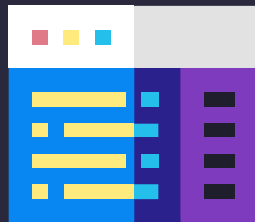
Installation

- Can be **installed** or used in a **docker-container**.
- Official docker image: *ravendb*.
- **Configurable** through the Studio, API, environment variables or files.
- **Safe by-default** - HTTPS enabled.



Data model

- **JSON** document format.
- Extensions:
 - **Counters** (distributed)
 - **Time-series** (streams of data)
 - **Attachments** (large data)
- **Revisions** with version vectors.



Data operations

- CRUD operations performed in the context of a **session/transaction**.
- **Extensions** have **specific operands** (increment/decrement counters).
- **Advanced features:**
 - ETL from external sources
 - Subscriptions
 - Backups



/03

/FEATURES&API

Main use cases and relevant
features.

Consistency



// All operations on a document or attachment using its ID are always consistent and run in an ACID transaction

Replication

- RavenDB supports replication through **clustering**.
- Clusters are **3+ nodes** working together.
- Databases may be replicated on **all nodes or a subset**.
- **ACID** properties ensured **cluster wide**.
- **Partition** may lead to conflicts.
- Conflict resolution is **customizable**.



Querying



/Raven Query Language

- > RQL allows to execute all available types of queries.



/Full-Text Search

- > Advanced full-text search operations over one or more fields.



/Spatial

- > RQL itself and our Client API contain built-in methods to help find the documents with the search areas.



/Facets

- > Perform calculations such as counts, and sums, on aggregated data and split it into defined ranges.



/MoreLikeThis

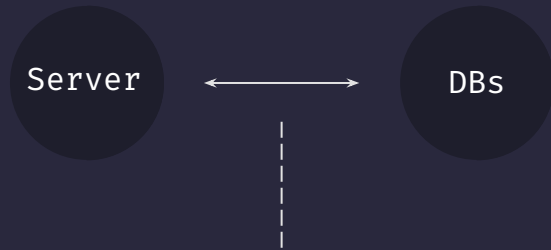
- > Built-in method to retrieve similar documents.



/Patching

- > Update documents directly on the server-side using our integrated JavaScript-based patching API, a part of RQL syntax.

API and client libraries



REST client
API

Client libraries for:



API and client libraries

Documents

Indexes

Settings

Stats

Manage database settings & contents from here

List of databases available on this server

DB-1	DB-2	DB-3
244 MBytes Up for a minute	17.28 MBytes Up for a few seconds	17.28 MBytes Up for a few seconds
0 documents Never backed up	0 documents Never backed up	0 documents Never backed up
0 indexes	0 indexes	0 indexes
Manage group	Manage group	Manage group
Disable	Disable	Disable

RavenDB studio

+ Add Task

External Replication

Replication to db3

Task Status: **Active**

Connection String: db3

Destination Database: db3

Actual Destination URL: http://localhost:8083

Topology Discovery URL: http://localhost:8083

Database Group Topology

Full screen mode

Replication to db3

TASK TYPE

- External Replication
- RavenDB ETL
- SQL ETL
- Backup
- Subscription

NODE TYPE

- Member
- Promotable
- Rehab

CONNECTION STATUS

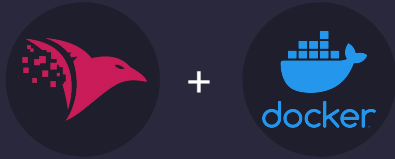
- Connection
- Partial Connection (ETL)
- Disabled Connection
- Broken Connection

/03

/LIVE_DEMO

RavenDB in action.

Prototype overview



Database

RavenDB instance
running in docker
container.



Backend

Simple node.js
backend using
Express.



Frontend

React web
application.

Data

- **Ecommerce** use scenario
- Data sourced from various product datasets
- **Clean-up and additional generation** using Python
- **6** conceptual **classes**
- **3** database **collections**

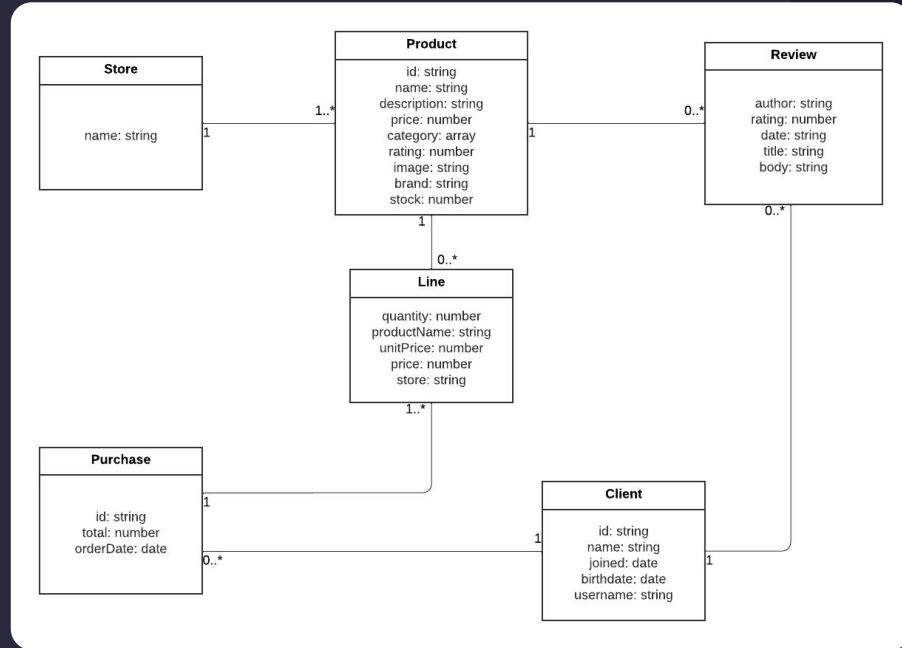


Fig.1 : Conceptual Data Model

<DEMO>

> Demonstrate Prototype and Studio. <

Conclusions

Negative

- Documentation lacking and sometimes confusing;
- Lack of documentation for all languages.

Positive

- Website and Studio intuitive and appealing;
- Versatility of features;
- High performance.

/THANKS!

Any Questions?

- > Ana Cruz | up201806460@up.pt
- > António Bezerra | up201806854@up.pt
- > Mariana Truta | up201806543@up.pt

