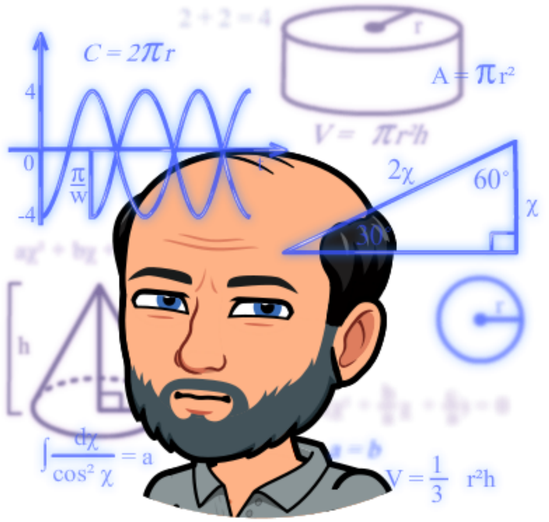
So, what about JSON in my database?



Thomas Hütter

Data Saturday Rheinland
St Augustin 2023

What about JSON in my database?

Thomas Hütter, Diplom-Betriebswirt

- Developer for ERP apps, SQL scripts and BI stuff
- Worked at consultancies, ISVs, end user companies
- 1995: SQL Server, 2001: Nav/BC, 2014: R, 2020: Power*
- Speaker at SQL / data / dev events around Europe

@DerFredo https://twitter.com/DerFredo
de.linkedin.com/in/derfredo
https://techhub.social/@DerFredo

SWECON

Developer!
DeveloperDeveloper
UK Community Events

sqlbits

{ } NDC
Conferences

DATA
INNOVATION
SUMMIT
2023

Pure expertise at the
SQL Server Konferenz 2018
FEB 26th - 28th, 2018 | DARMSTADT
Sign up now at sqlkonferenz.de

< >

Agenda

- JSON…
  - where it comes from
  - what is it used for
  - what it looks like
- How SQL Server handles JSON
- Comparing with Azure Cosmos DB and MongoDB
- Conclusions
- Credits & resources

JSON… why bother?

- Original motivation: pre-study for a project
- Receive JSON data regularly and automated from an external source
- Data amount: x.000 records/documents per day, each < 10 kB
- Store and later retrieve/analyze the data
- Mainly on-premises environment, but cloud an option

- Project eventually didn’t materialize („maybe later…“)
- Therefor no real proof of concept, no cost comparisons
- ~~Author trashed the original slide deck~~ 🙄

JSON… where it comes from

- In the early 2000s, a stateless, real-time server-to-browser data-interchange communication protocol was needed
- First specified by Douglas Crockford as a lightweight alternative to XML
- JSON = JavaScript Object Notation („hipster's XML“)
- Based on a subset of JavaScript, but language-independent data format
- Relatively easy to read and write, as it is a plain-text format
- Easy to parse and generate by any programming language
- Current specifications: RFC 8259, IETF Standard 90, ECMA-404
- „The software should be used for good, not evil“ is part of the license

JSON… what is it used for

- „Why JSON? - JSON is everywhere!“ (Jovan Popovic, PM at Microsoft)
- transferring data between web servers and applications
- IoT devices send semi-structured data in JSON format
- log and telemetry data may come JSON-formatted
- cloud services use JSON format internally (Azure Stream Analytics)
- user settings are stored in json files (e.g. VS Code or Azure Data Studio)
- popular data format for *NoSQL* databases
- …

JSON... what it looks like (see also json.org)

- A JSON object is an unordered collection of name-value pairs,
  enclosed in curly brackets `{` and `}` aka braces
- Name-value pairs are separated by a comma `,`
- Names are strings surrounded by double quotes `"  "` and followed by a colon `:`
- For the values there is a limited choice of data types:
  - numbers (signed, decimal fractions, exponential notation)
  - strings (Unicode, double-quote delimited, backslash escaping supported)
  - boolean values (`true` or `false`)
  - empty/unknown value using `NULL`
  - array = ordered list of elements, comma-separated within `[ ]`
  - another object, so nesting is possible
- Throw in whitespace (incl. linefeeds) almost everywhere

JSON… what it looks like

- `{"event": "Sat Rheinland 2023", "location": "St Augustin"}`

- ```
  {
   "event": "Sat Rheinland 2023",
   "location": "St Augustin"
  }
  ```

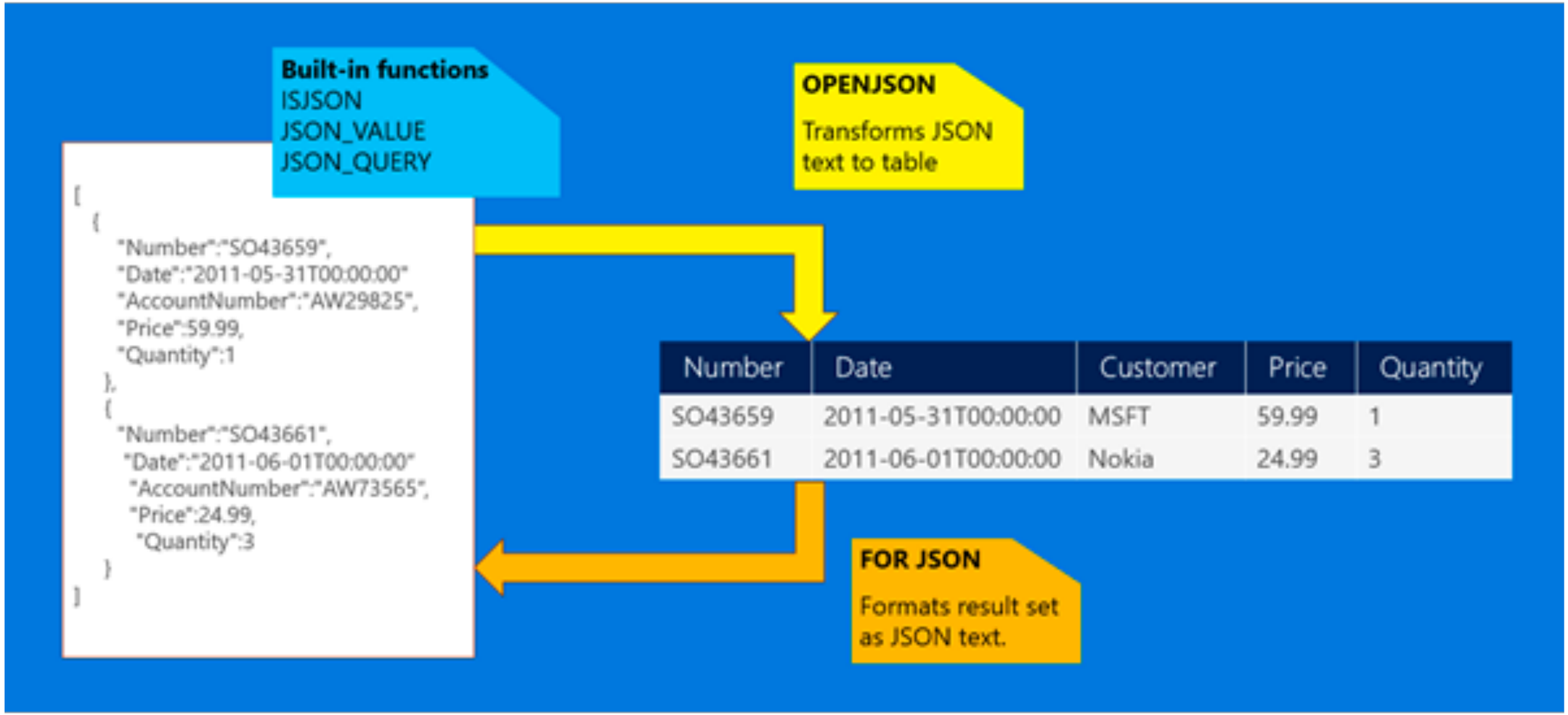
- `{"event": "Sat Rheinland 2023"; "location": "St Augustin"}`

- `{event: "Sat Rheinland 2023", location: "St Augustin"}`
  ```

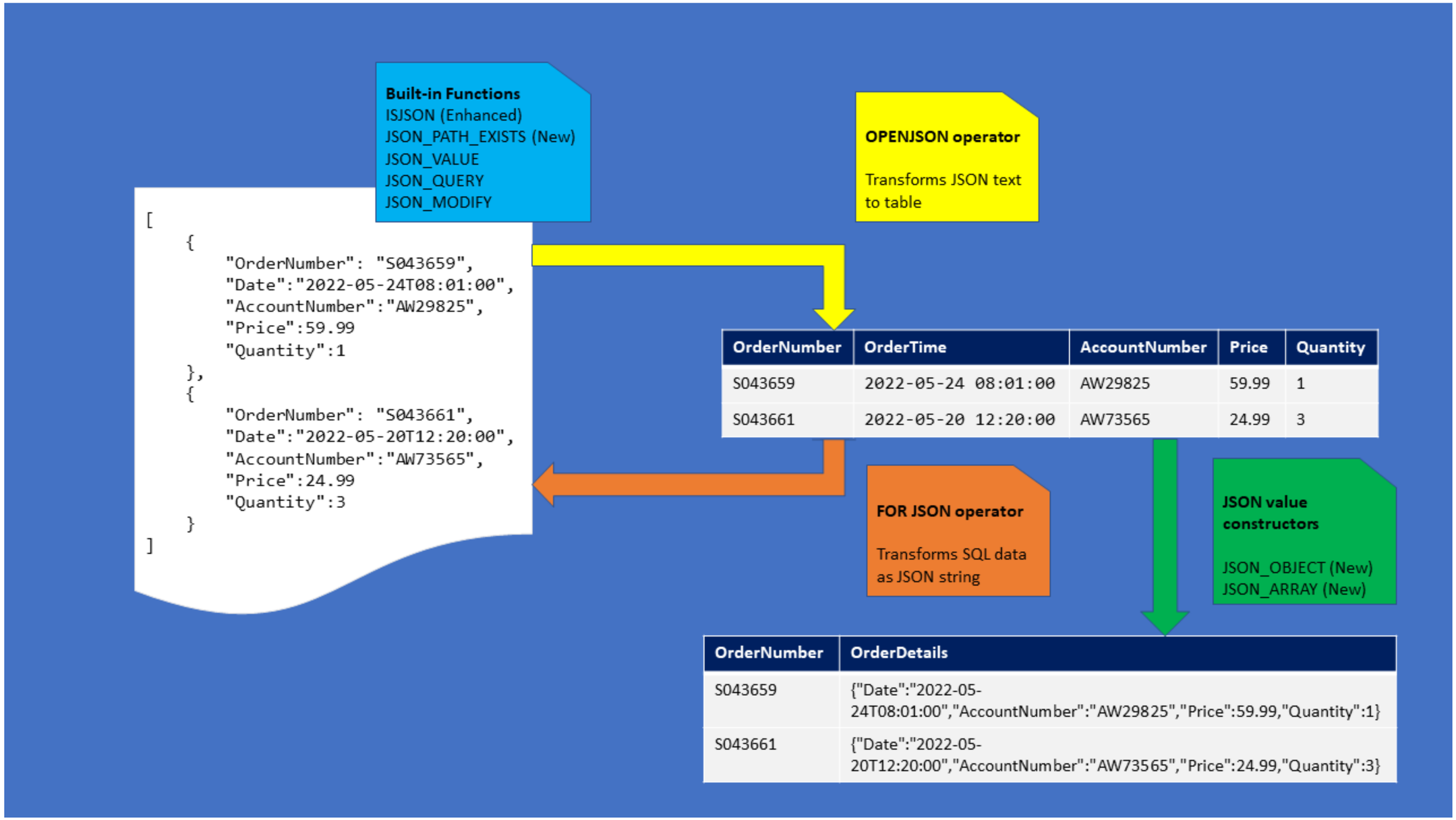
JSON… what it looks like

- ```
  {
  "ID": 1,
  "CustomerName": "Tailspin Toys (Head Office)",
  "CustomerCategoryName": "Novelty Shop",
  "Contact": {
              "Phone": "(308) 555-0100",
              "Fax": "(308) 555-0101"
             },
  "CityName": "Lisco"
  }
  ```
- ```
  {
  "PersonID": 10,
  "FullName": "Stella Rosenhain",
  "OtherLanguages": ["Dutch", "Finnish", "Lithuanian"]
  }
  ```

How SQL Server < 2022 handles JSON



How SQL Server 2022 handles JSON



How SQL Server handles JSON

Constructing JSON from relational table data

- `SELECT … FROM … FOR JSON AUTO | PATH`
  `AUTO` formats the JSON output automatically,
  `PATH` allows for nesting of complex objects
- `ROOT(…)` adds a single top-level element to the output
- `INCLUDE_NULL_VALUES`
  specify this to include JSON properties for NULL values in your output
- `WITHOUT_ARRAY_WRAPPER`
  removes the square brackets surrounding the JSON output by default
- `JSON_OBJECT()` and `JSON_ARRAY()`
  new functions in SQL Server 2022 to generate JSON output

How SQL Server handles JSON

Storing JSON data in SQL Server

- No dedicated data type for JSON data in SQL Server (… yet 😉)
- Recommended ways of storing:
  `NVarChar(4000)` fits on one data page (performance benefit) or
  `NVarChar(Max)` Unicode data with a size of up to 2GB
- Since this is a text data type that is supported in all subsystems of SQL Server, you can take advantage of column store indexes, memory optimized tables, external files, Polybase, …

How SQL Server handles JSON

Retrieving JSON data from SQL Server

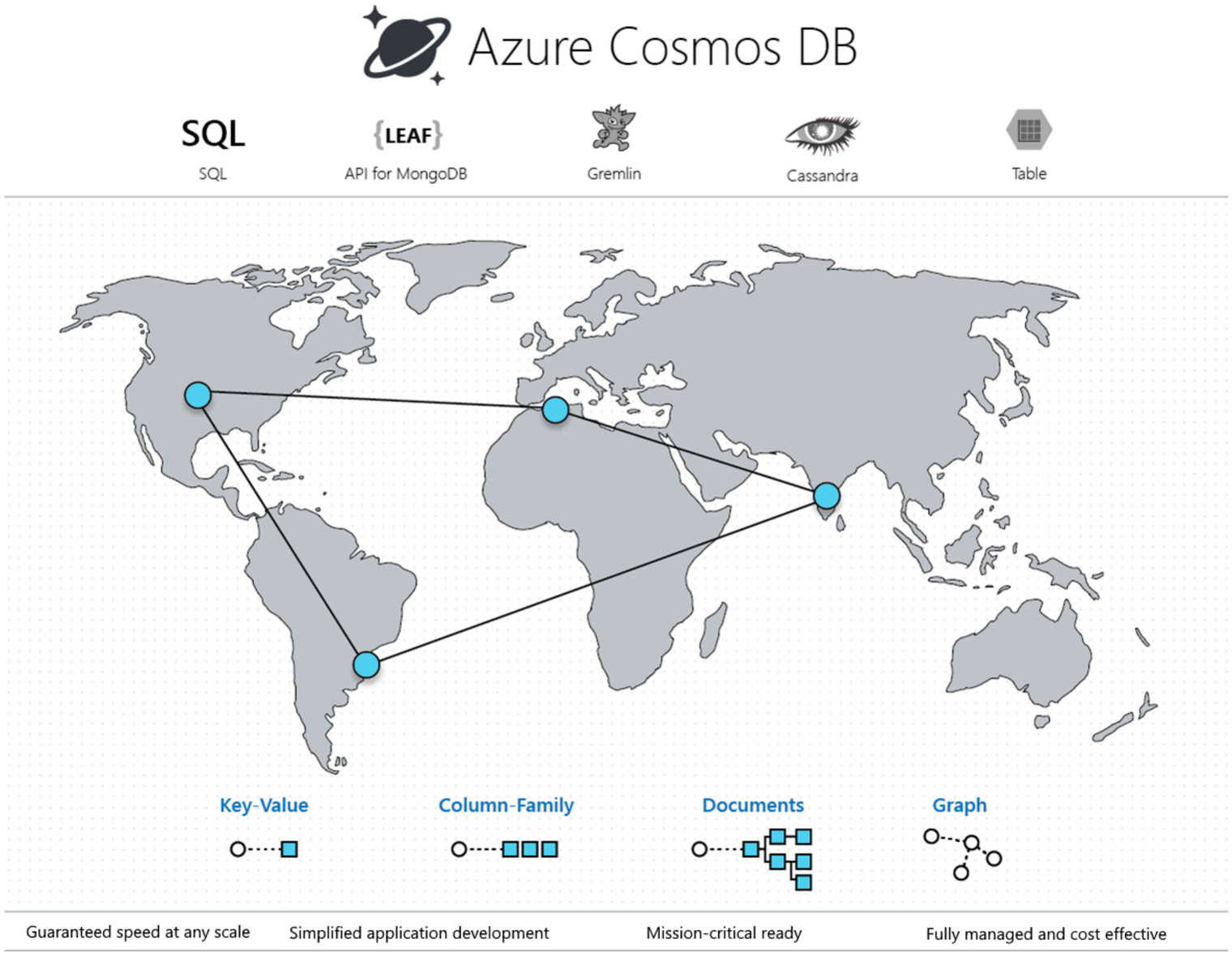
- `ISJSON()` checks for valid JSON syntax, additional parameters in SQL 2022:
  `ISJSON(SomeVar, VALUE|OBJECT|ARRAY|SCALAR)`
- New in SQL 2022: `JSON_PATH_EXISTS(SomeVar, '$.SomePath')`
- `OPENJSON()` to transform (simple) JSON to relational format,
  by default returns key, value and type of your JSON data
- `JSON_VALUE()` extracts one scalar value from a JSON object/text,
  returns a single text value of type NVarChar(4000)
- `JSON_QUERY()` to extract an array or an object from a JSON string
- Results can be further processed just like T-SQL

How SQL Server handles JSON

Modifying JSON data within SQL Server

- `JSON_MODIFY()` updates the value of a property in a JSON string and returns the updated JSON string
- can be used to modify existing values or append new key-value pairs
- in *strict* mode, the referenced property must exist
- in *lax* mode (default):
  - if the property does not exist, JSON_Modify tries to insert the value on the specified path
  - the specified key will be deleted if the new value is NULL

Comparing with Azure Cosmos DB and MongoDB

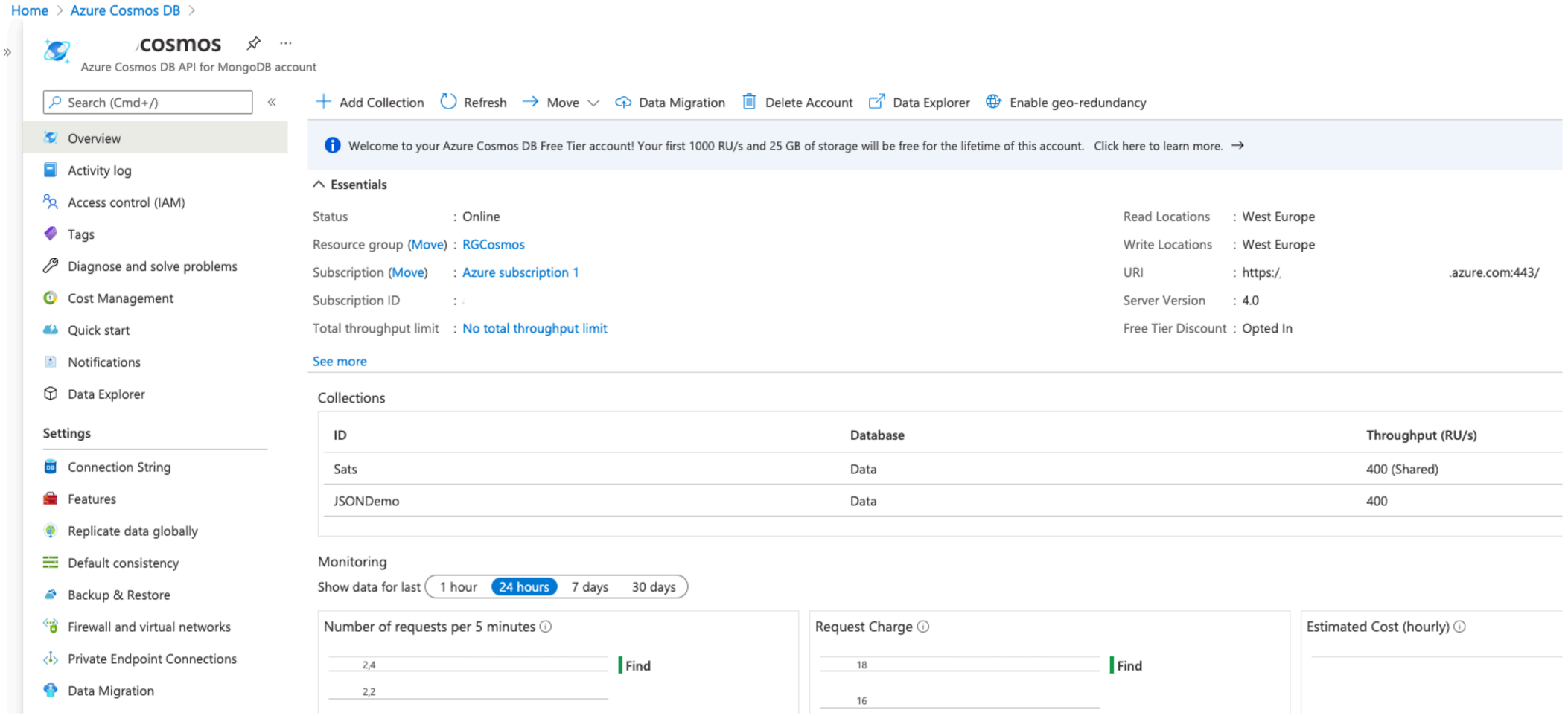


Comparing with Azure Cosmos DB and MongoDB

- Cosmos DB is a distributed multi-model NoSQL database service, exposing the data model (items in containers) through several APIs
- Developed by *Microsoft Corporation*, Redmond, Washington/USA
- Initial release in May 2017
- License: Proprietary; free tier 1000 RU/s and 25 GB of storage (as of June 2023)
- APIs available:
  native Core(SQL) API - enables queries in SQL syntax
  MongoDB API - wire-compatible to MongoDB Server version 4.2/[5.0] (as of June 2023)
  Cassandra API - enables Apache Cassandra apps
  Gremlin API - graph database
  Table API - overcoming limitations of Azure Table Storage

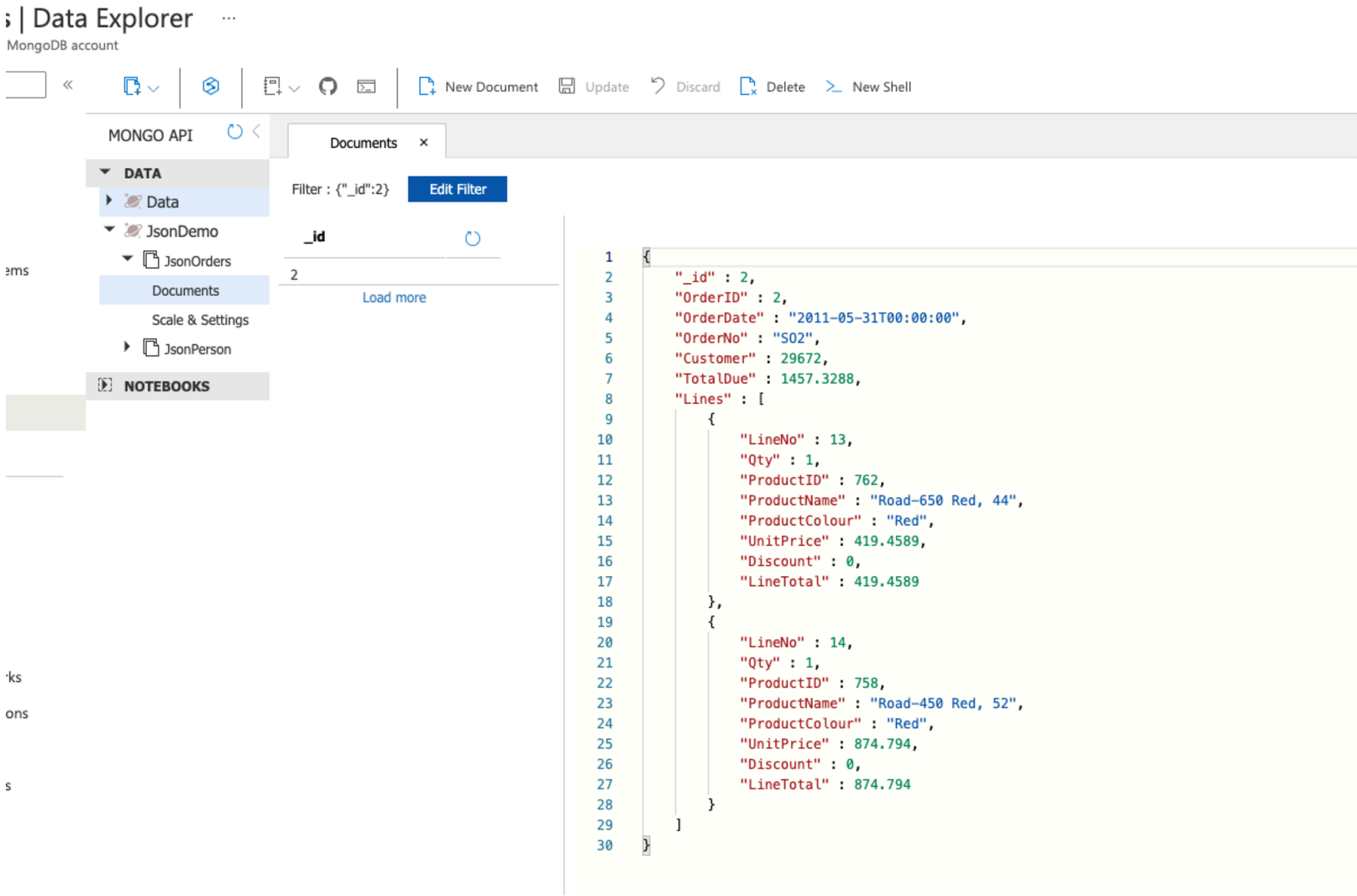
Comparing with Azure Cosmos DB and MongoDB

- Managing your Azure CosmosDB

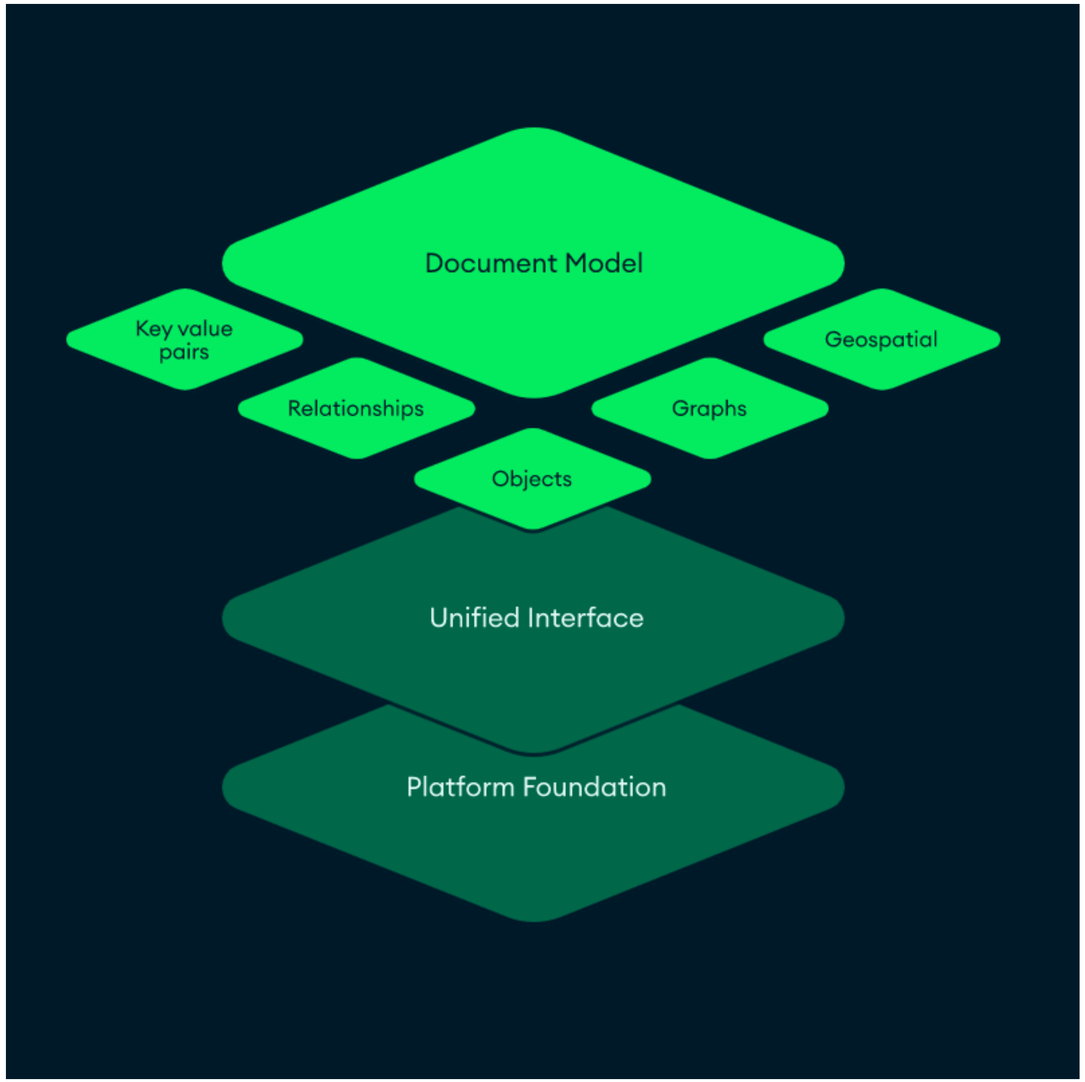


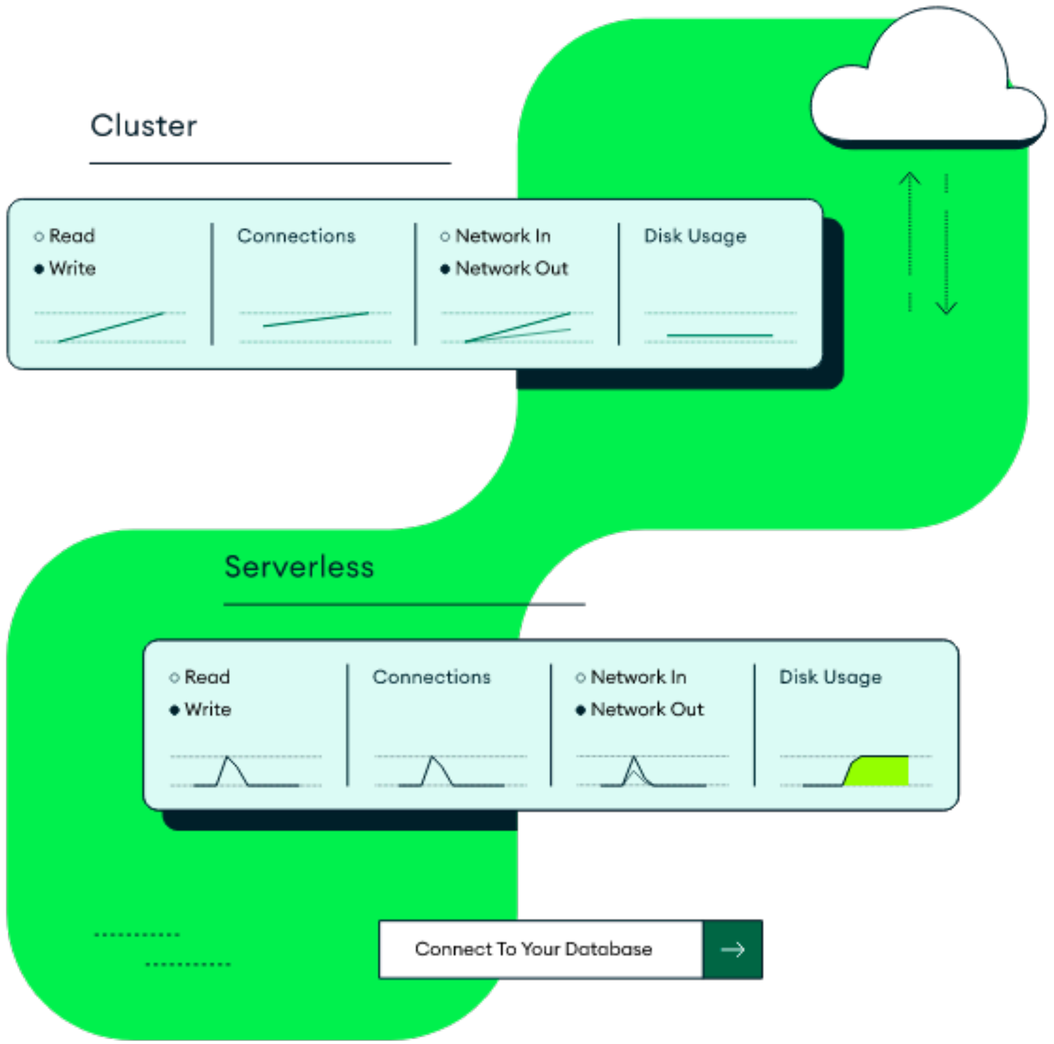
Comparing with Azure Cosmos DB and MongoDB

- Accessing data in Azure CosmosDB via Data Explorer



Comparing with Azure Cosmos DB and MongoDB



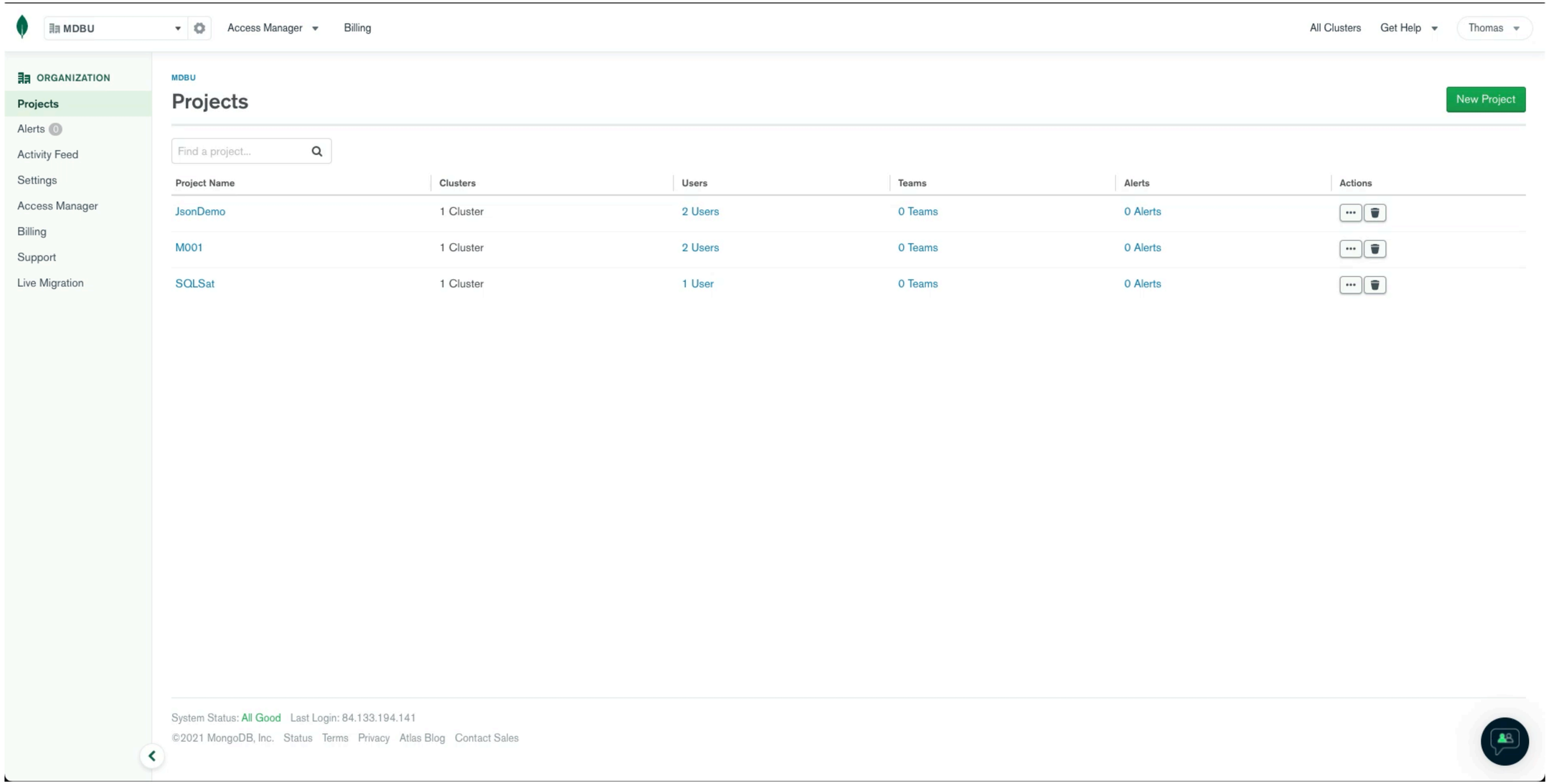


Comparing with Azure Cosmos DB and MongoDB

- MongoDB is a multi-platform document-oriented NoSQL database program using JSON style documents with optional schemas
- Developed by *MongoDB Inc.* (former *10gen Inc.*), New York City/USA
- Initial release in Feb 2009, current stable version (as of June 2023) 6.0.6
- License: SSPL (Server-side public license), source available
- Editions available:
  Community Server - free (for Linux, MacOS, Windows)
  Enterprise Server - commercial
  MongoDB Atlas - on AWS, Microsoft Azure, Google Cloud Platform

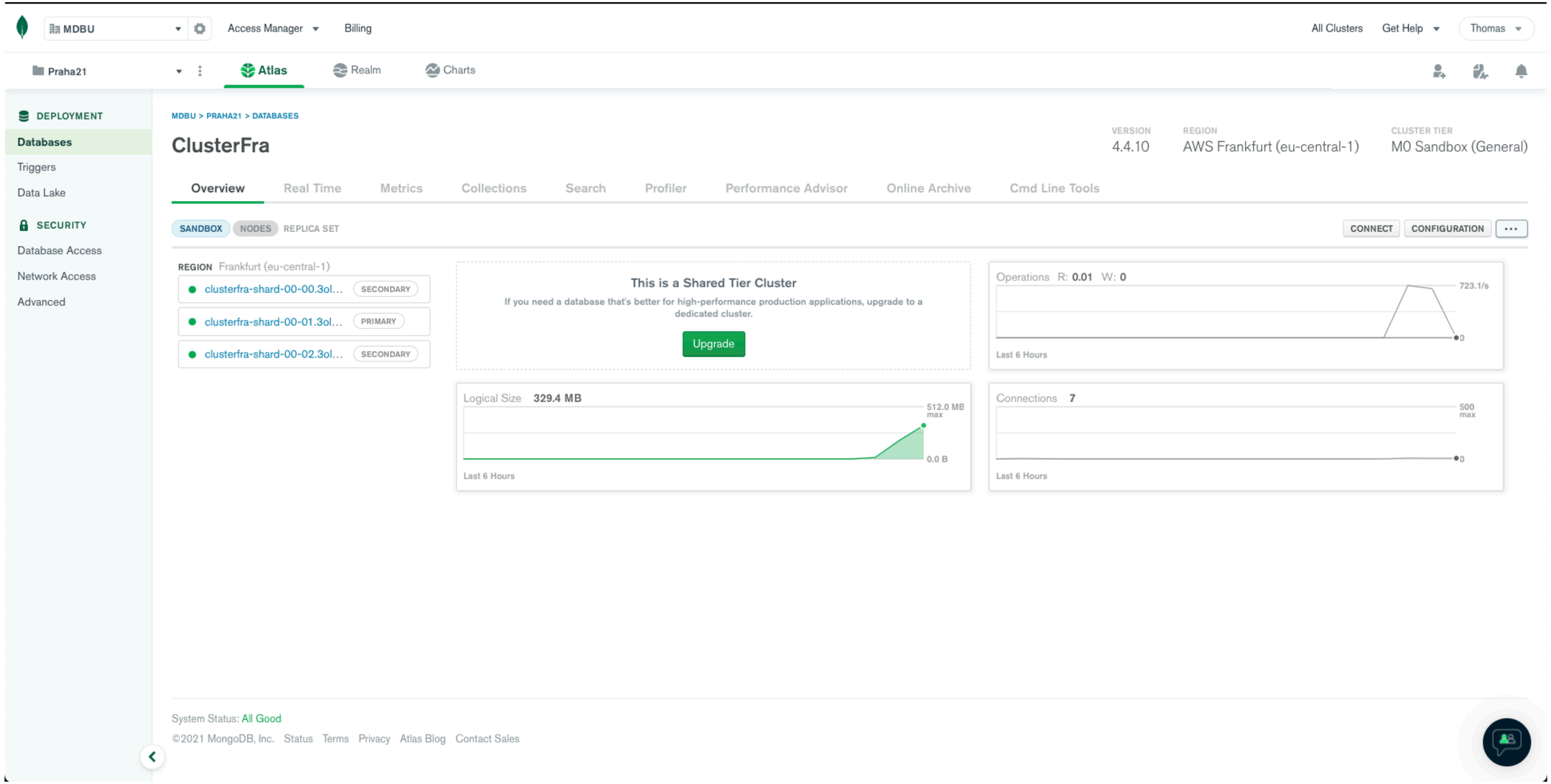
Comparing with Azure Cosmos DB and MongoDB

- Provisioning a MongoDB environment



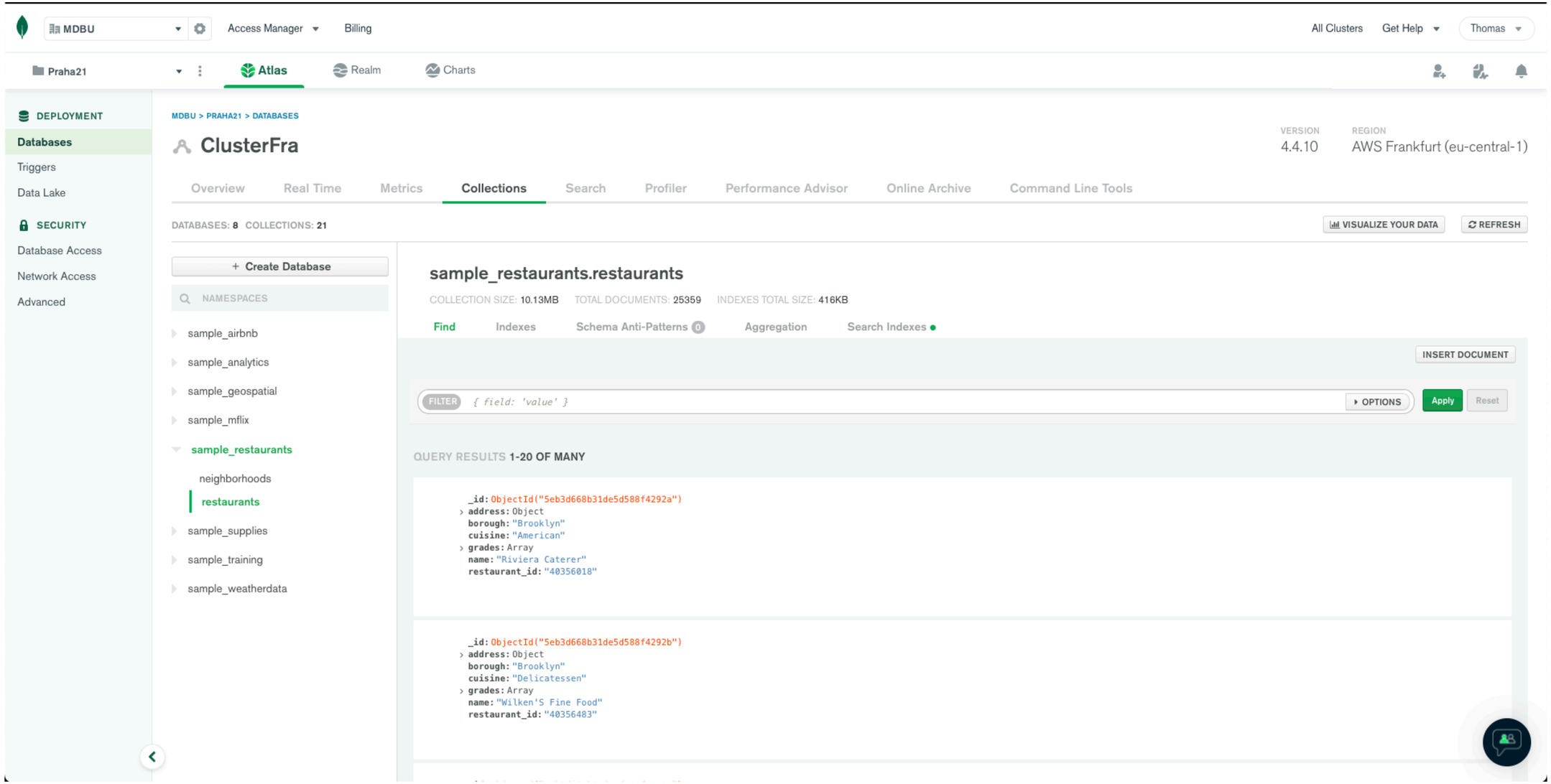
Comparing with Azure Cosmos DB and MongoDB

- Creates a 3-node cluster



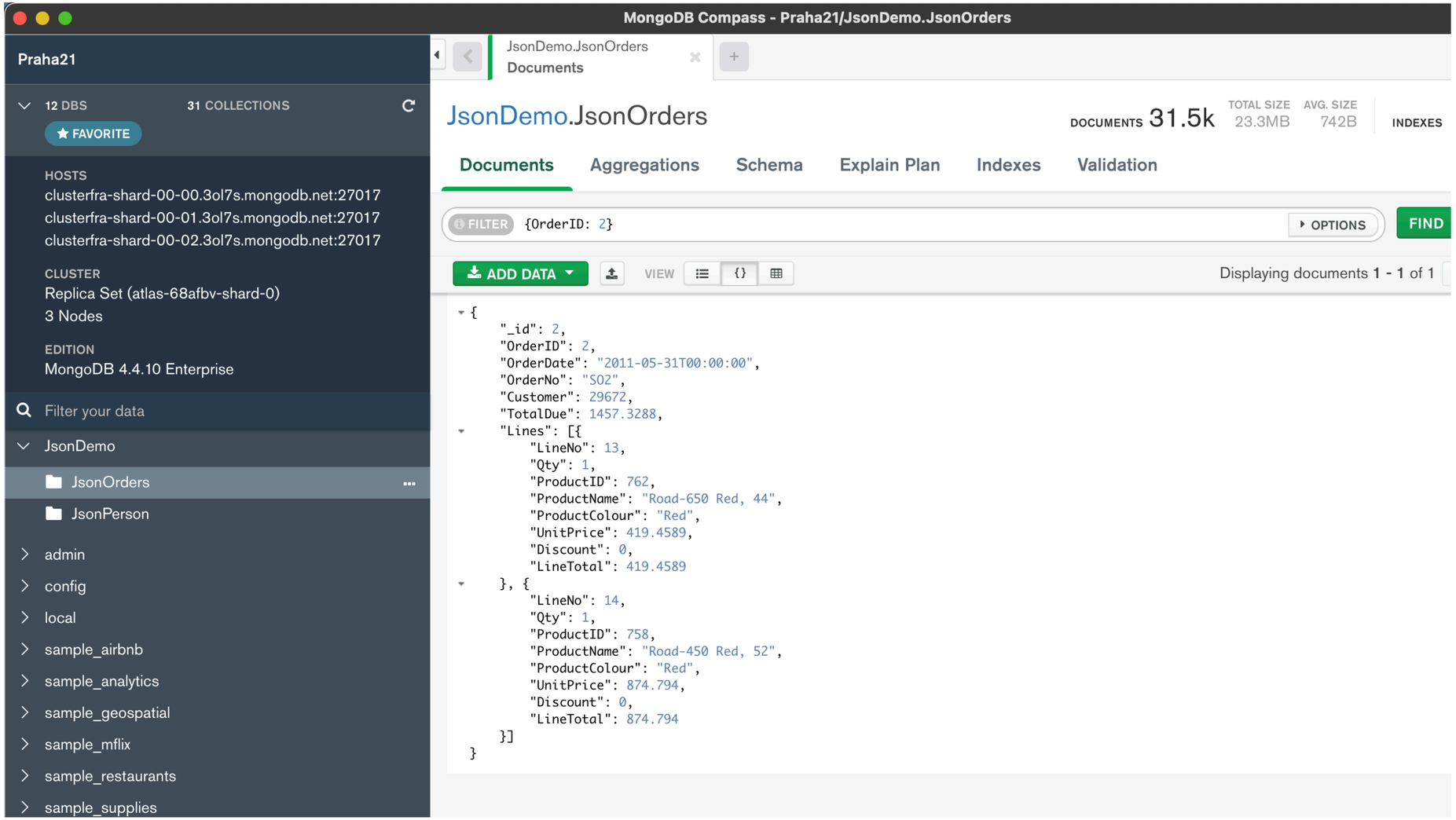
Comparing with Azure Cosmos DB and MongoDB

- Installed the sample databases



Comparing with Azure Cosmos DB and MongoDB

- Using MongoDB Compass to access your data



Comparing with Azure Cosmos DB and MongoDB

- Atlas - cloud version of the MongoDB database on AWS, Azure, Google
  - Realm - build mobiles apps, web sites, services to connect to MongoDB
  - Charts - build dashboards and charts on your data
  - Data Lake - „serverless“, scalable analyzing query engine
- Enterprise Server
  - commercial edition, includes additional capabilities
- Community Server
  - on-premises version of MongoDB Atlas
- MongoDB Compass
  - the GUI client/tool for local installation

Comparing with Azure Cosmos DB and MongoDB

	SQL Server	CosmosDB	MongoDB
Primary DB model	Relational	Document/Graph/ Key-value/Table	Document Store
Initial release, license	1989, commercial	2017, commercial	2009, open source
Cloud-based/on-prem	yes / yes	yes / no	yes / yes
Operating system	Windows, Linux, Docker	(hosted)	Windows, Linux, MacOS
Supported program. languages	~ 12	~ 6 + Mongo-driven	~ 30
DB-ranking (*) overall/ document stores	3 / -	27 / 4	5 / 1
Partitioning/replic ation	yes / yes	yes / yes	yes / yes
Consistency	immediate	immediate, eventual	immediate, eventual
Scalability	on-prem vs Azure	„instant, automatic“	vertical & horizontal

(*) according to db-engines.com, June 2023

Conclusions for handling JSON in your database

- **SQL Server**
  - on-premises or in Azure Cloud, probably „already there“
  - no dedicated (optimized) data type for handling JSON yet
  - integrates well with relational data queries
- **Azure Cosmos DB** (with Mongo API)
  - MS cloud only, no on-premises solution
  - integrates well with other Azure services
  - will scale „mostly without“ manual input
- **MongoDB**
  - on-premises, or in one of the big 3 cloud environments
  - „the original“ when it comes to JSON-style document store
  - scalable from free to Enterprise Server, not automatically

Credits & resources

- General info: https://www.json.org/json-en.html, https://en.wikipedia.org/wiki/JSON
- SQL Server: https://docs.microsoft.com/en-us/sql/relational-databases/json/json-data-sql-server
- Cosmos DB: https://docs.microsoft.com/en-us/azure/cosmos-db/introduction https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-query-working-with-json, https://docs.microsoft.com/en-us/azure/cosmos-db/mongodb/mongodb-introduction
- MongoDB: https://docs.mongodb.com/
- Comparisons: https://db-engines.com/en/system/Microsoft+Azure+Cosmos+DB;Microsoft+SQL+Server;MongoDB, https://sourceforge.net/software/compare/Azure-Cosmos-DB-vs-MongoDB-vs-SQL-Server/, https://www.trustradius.com/compare-products/azure-cosmos-db-vs-mongodb-atlas

What about JSON in my database?

Thank you for your time and interest & keep in touch:

- @DerFredo https://twitter.com/DerFredo
- de.linkedin.com/in/derfredo
- https://techhub.social/@DerFredo

This file and the demo scripts can be found at:
https://bit.ly/DerFredoRhein23