**ELG 5166 – Cloud Analytics**

Assignment #1

Group – 9

|  |  |
| --- | --- |
| Mohamed Elesawy | 300327237 |
| Eslam Khalaf | 300327261 |
| Ahmed Abdo Zaid | 300327306 |
| Kareem Waly | 300327303 |

Text, letter

Description automatically generated

# 

# Part 1: Questions

1. Describe briefly what a NoSQL database means. Select a NoSQL database (except MongoDB & Cassandra) and describe how this database can be used for the storage and management of big data.

A NoSQL database refers to ‘not only SQL’ that means it can store many types of data not only structured data but also semi structured data the availability is the most important factor for it. In relational databases, consistency and atomicity is the best important factor due to dealing with heavy inserting and updates queries then the performance of reading queries not good as needed because there are many joins that will slow down the performance, so in the big data needs massive reading queries that will not be suitable with relational databases, so NoSQL appears to solve massive reading big data problems.

**Riak Database**: is a particular kind of key-value store for a NoSQL database that can store any kind of data and does so over numerous nodes in what is referred to as a cluster. Data replication occurs in a cluster to support partition tolerance. This method of data replication ensures that this data is constantly accessible. Users of websites like social networking platforms or file-sharing programmers are an illustration of this. The data has been replicated among numerous nodes that extend across several servers, so it is safe in the case of a server crash, system failure, or other network anomaly. Once all nodes are successfully back online again, Riak starts to automatically allow reads and writes to and from the nodes that were not operational. In addition to extreme fault tolerance, Riak is highly scalable; it allows anyone to add new nodes with a few short commands and scale the size of the cluster to meet the data storage needs in a simple, linear, and predictable fashion, saving both time and money.

Or

**AWS DynamoDB:** is a fully managed key-value database offered by amazon web services, that delivers single-digit millisecond performance at any scale. you don’t have to configure anything because it’s a fully managed database, AWS manage scalability, configuration, replication, at rest encryption and hardware provisioning.

2. Investigate and describe one application of Big Data Analytics that was not described in

Class.

**Big data analytics in organizational performance:**

Large data samples can assist in identifying and leveraging business transformation and offer business-centric approaches and practices that give organizations or enterprises a competitive advantage by enabling them to enhance their current applications and daily operational decisions ad in creating new business values for the organization.

3. Briefly describe the transaction management features of Cassandra and MongoDB in the context of ACID vs. BASE properties.

|  |  |
| --- | --- |
| **Cassandra** | **MongoDB** |
| * Cassandra can be categorized as a BASE database. * It possesses atomicity, isolation, and long-term durability.[1] * It supports atomic(A), because inserting or modifying several columns in a row is considered a single write operation.[2] * Cassandra cannot be described as an ACID-compliant database because the ACID lacks consistency(C) the concepts of referential integrity, foreign keys, and joins.[2] * It supports isolation(I) Because it believes that writing to a row should be isolated to just the user who is writing and should not be visible to any other users until the current user has finished writing.[2] * It supports durability(D) since all writes to a replica node are stored in memory and committed to disc before being declared successful.[2] | * MongoDB is a NoSQL DB that supports ACID compliance too at the document level.[3] * MongoDB supports multi-document ACID transactions on a single replica set.[3] * The operation on a single document is atomic(A), since by using embedded documents to obtain relationships between data in a single document structure rather than normalizing across several documents and collections, the need for multi-document transactions is removed. [3] * MongoDB supports consistency(C) the most recent value will be returned by any subsequent read after a write has finished.[3] * MongoDB is isolation(I) since Data changes implemented during a transaction will not be visible outside of that transaction, until each transaction has finished processing, the data from other transactions cannot be accessed when several transactions are being carried out.[3] * It supports durability(D) as it accommodates data from heterogeneous to homogeneous.[3] |

4. You are working on a project that requires you to capture data from millions of IoT devices in people’s homes. Each IoT device uploads a JSON document with the data elements required for analytics.

1. Identify potential NoSQL databases that you can capture data from the IoT Devices

* Document Database: such as MongoDB, it’s a perfect solution for IOT app because it’s power of storing data as JSON documents**.[4]**
* Time-Series Database: such as Timestream, it designed to hold time-oriented data, which makes it ideal for time-series-dependent machine learning models, IoT applications, and audit logs**.[4]**
* Columnar Database: such as Redshift, considered as a good option because its power of ignoring the rows to directly query the columns. **[4]**

1. What are your design and analytics considerations and rationale behind your choice?

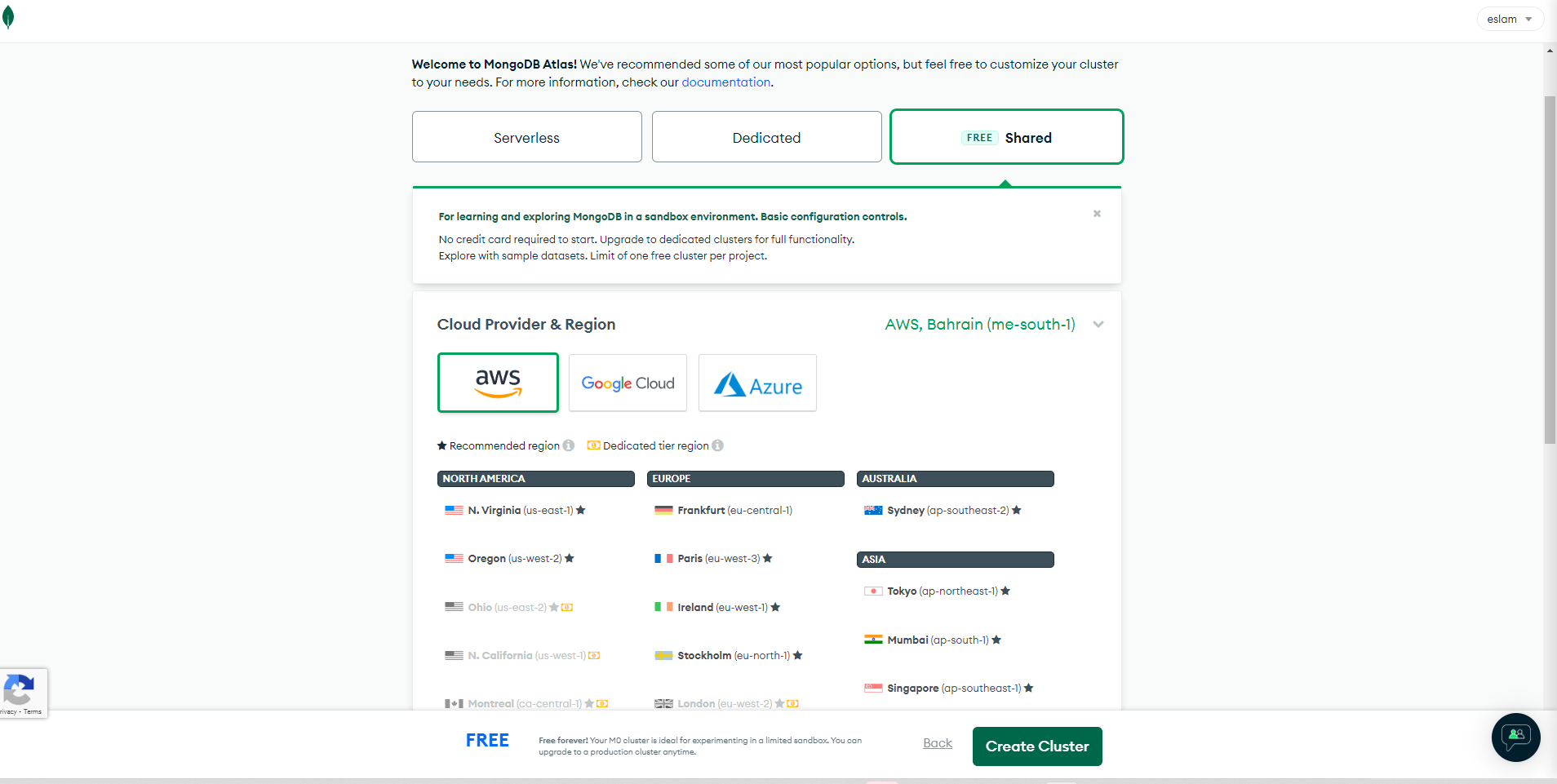
To get the most versatility out of this medium, choose a document database like MongoDB, which saves data as a JSON file and can handle the flood of data from your devices. In addition to the Columnar Database due to the fact that it may be used where analysis or grouping is being done on certain columns for the IoT applications, Time Series are an excellent choice for collecting change events in system, sensing, or storing device readings.[4]

# Part 2: NoSQL Labs

Setup (Please show evidence of your setup with screenshots)

1. **MongoDB Lab**
   1. Set an account on MongoDB Atlas - <https://cloud.mongodb.com>

Graphical user interface, text, application

Description automatically generated 

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

* 1. Load the Sample Netflix Movies Database to your Data Lake

Graphical user interface, text, application

Description automatically generated

**Answers:**

**1. Briefly describe the movies database document model.**

After loading Sample Netflix Movies Database, we noticed documents contain the following attributes, but these attributes not found in all documents but these attributes we noticed them by showing some documents in our database:

This is the data model for our document that represents that the movie document has many (awards, imdb, tomatoes) and each tomatoes has many (viewer, critic)

Diagram

Description automatically generated

**•** “\_id” with type “ObjectId” that represent the document ID of each document

• “plot” with type “String”

• “genres” with type “Array”

• “runtime” with type “Integer”

• “cast” with type “Array”

• “num\_mflix\_comments” with type “Integer”

• “poster” with type “String”

• “title” with type “String”

• “fullplot” with type “String”

• “languages” with type “Array”

• “released” with type “Timestamp”

• “directors” with type “Array”

• “rated” with type “String”

• “awards” with type “Object” that object contain the following (“wins” with type “Integer”, “nominations” with type

“Integer” and “text” with type “String”)

• “lastupdated” with type “String”

• “year” with type “Integer”

• “imdb” with type “Object” that object contain the following (“rating” with type “Double”, “votes” with type “Integer” and ” id” with type “Integer” )

• “countries” with type “Array”

• “type” with type “String”

•“tomatoes” with type “Object” that object contain the following (“viewer” with type “Object” that object contain the

following (“rating” with type “Double”, “numReviews” with type “Integer” and “meter” with type “Integer”)

“critic” with type “Object” that object contain the following (“rating” with type “Double”, “numReviews” with type “Integer” and “meter” with type “Integer”)

“rotten” with type “Integer” and

“fresh” with type “Integer” and

“dvd” with type “Timestamp” and

“lastUpdated” with type “Timestamp”).

All of this represents each document in Netflix Movies Database.

**2.**  **Filter the documents for type “movies” that are releasedbefore 1970 and rated as “PASSED”.**

{type:"movie",rated:"PASSED",released:{$lt:ISODate("1970")}}

Graphical user interface, text, application, email

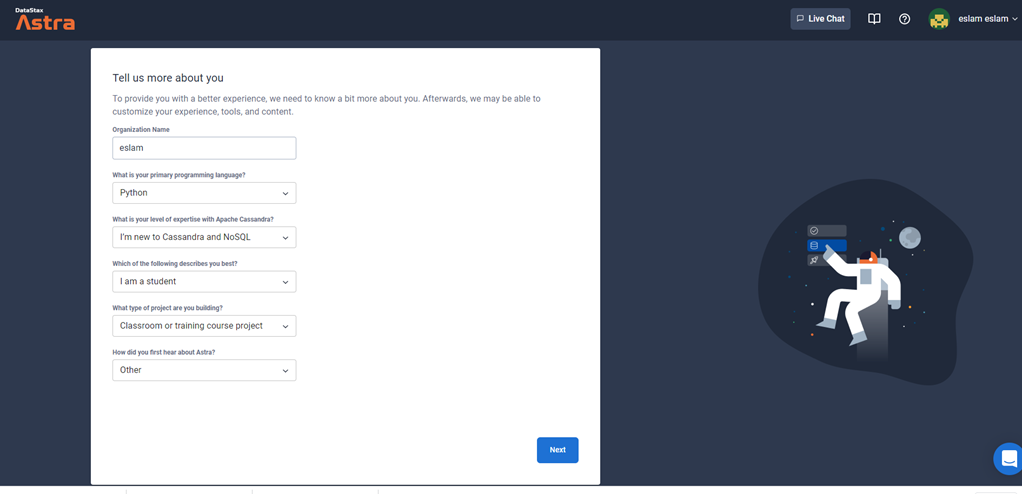
Description automatically generated

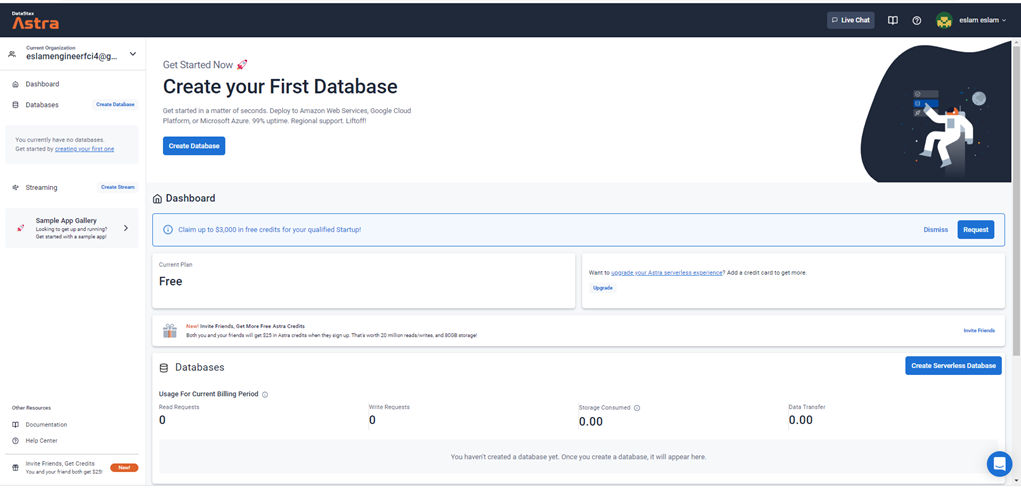
* 1. **Build an Aggregation Pipeline that shows all movies (“type”) that have won at least one award and provide the results in release year aggregate count.**

|  |  |  |
| --- | --- | --- |
| {  type:"movie",  "awards.wins":{$gte:1}  }{  \_id: "$year",  count: {  "$sum": 1  }} | Graphical user interface, application  Description automatically generated | |
| Graphical user interface, text, application  Description automatically generated | |

1. **Cassandra Lab**

Using DataStax Astra Cassandra-as-a-Service (<https://astra.datastax.com/)>





Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Create the customer tables (attached SQLite definition will serve as a guide) Review the questions in the queries section below and create one or more tables that partition and clusters data so these queries will execute without using Cassandra “ALLOW FILTERING” that scans all partitions.

Text

Description automatically generated

Load the attached data into your table(s) using the insert statements (minor modifications may be needed if your definitions include multiple tables). Please include screens counts after loading your data.

Text

Description automatically generated

Select \* from Customers limit 10; (show 10 records from customers table)

Graphical user interface, text, application

Description automatically generated

Select count (\*) from Customers;

Text

Description automatically generated

Queries:

1. Provide the query and the results (screenshots and a copy of your query) that show the customers from Rio de Janeiro, Brazil ordered by their addresses.

Our first table is Customers\_\_\_country\_city\_\_\_address that represent our first query

In our first table we used (country, city) as partition keys and (address, customerId) as cluster keys because these four values will make a unique value for each record

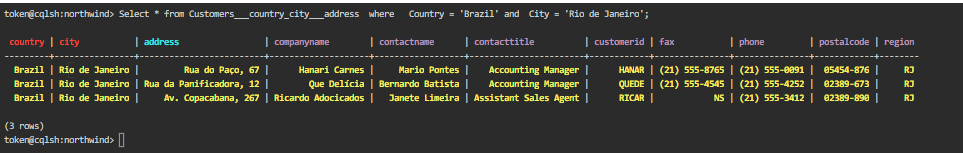
**Graphical user interface, text

Description automatically generated**

There aren’t any data yet so we will insert our data to our table that will be used in first query



Select \* from Customers\_\_\_country\_city\_\_\_address where Country = 'Brazil' and City = 'Rio de Janeiro';

****

1. Provide a list of customers that are in the Sales Manager role without forcing the scan of all partitions across all databases. The result should be ordered by their names.

Our second table is Customers\_\_\_contactname\_\_\_contacttitle that represent our second query and insert data to it

In our second table we used (ContactTitle) as partition key and (ContactName, Phone) as cluster keys because these three values will make a unique value for each record

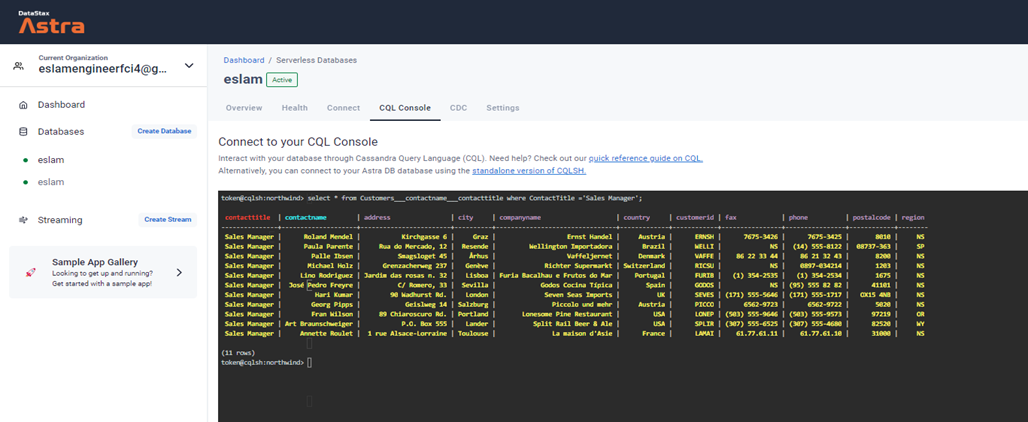
**Graphical user interface, text, application

Description automatically generated**

**Text

Description automatically generated**

select \* from Customers\_\_\_contactname\_\_\_contacttitle where ContactTitle ='Sales Manager';



# References

[1] <https://www.yugabyte.com/blog/apache-cassandra-lightweight-transactions-secondary-indexes-tunable-consistency/>

[2]<https://docs.datastax.com/en/cassandraoss/2.2/cassandra/dml/dmlTransactionsDiffer.html>

[3] <https://studio3t.com/whats-new/mongodb-acid-properties/>

[4] <https://www.verypossible.com/insights/nosql-for-iot-development-how-to-choose-the-right-database>