University of Ottawa

Faculty of Engineering

School of Electrical Engineering and Computer Science

ELG 5166 – Cloud Analytics

Assignment #1

Personal Ethics & Academic Integrity Statement

Student name: Esraa Ahmed Abdelhakam Abo wadaa Student ID: 300327225

Student Name: Abdallah Mohamed Mahmoud Mohamed Ragab Student ID: 300327288

Student Name: Hosam Mahmoud Ibrahim Mahmoud Student ID: 300327269

Student Name: Sondos Mohammed Hussein Ali Student ID: 300327219

By typing in my name and student ID on this form and submitting it electronically, I am attesting to the fact that I have reviewed not only my work but the work of my team member, in its entirety.

I attest to the fact that my work in this project adheres to the fraud policies as outlined in the Academic Regulations in the University's Graduate Studies Calendar. I further attest that I have knowledge of and have respected the "Beware of Plagiarism" brochure for the university. To the best of my knowledge, I also believe that each of my group colleagues has also met the aforementioned requirements and regulations. I understand that if my group assignment is submitted without a completed copy of this Personal Work Statement from each group member, it will be interpreted by the school that the missing student(s) name is confirmation of the nonparticipation of the aforementioned student(s) in the required work.

We, by typing in our names and student IDs on this form and submitting it electronically,

- warrant that the work submitted herein is our own group members' work and not the work of others
- acknowledge that we have read and understood the University Regulations on Academic Misconduct
- acknowledge that it is a breach of University Regulations to give or receive unauthorized and/or unacknowledged assistance on a graded piece of work

A. Part 1

I. Describe briefly what a NoSQL database means.

NoSQL (nonrelational database) is a database design approach that allows for data storage and querying outside of the traditional structures found in relational databases. It accesses and manages data using a variety of data models. These databases are designed specifically for applications that require high data volumes, low latency, and flexible data models, which are accomplished by relaxing some of the data consistency constraints of other databases.

NoSQL databases offer a number of data models, including key-value, document, and graph, that are optimized for performance and scalability.

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

HBase:

HBase is a distributed database, capable of hosting very large tables because it is layered on commodity hardware Hadoop clusters, it stores the data in HDFS-indexed files. It is highly configurable, allowing for a great deal of flexibility in dealing with massive amounts of data. Because HBase is a columnar database, all data is stored in tables with rows and columns, much like relational database management systems (RDBMSs), It stores the data in cells in decreasing order (using the timestamp) so that a read always returns the most recent values first.

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

Transportation:

Applications of Big Data in the Transportation Industry:

Big Data is used by governments for a variety of purposes, including traffic control, route planning, intelligent transportation systems, and congestion management by predicting traffic conditions. It also can be used for a variety of purposes, such as route planning to save on fuel and time, as well as for travel arrangements for tourism.

Challenges:

1. Location-based social network data and high-speed telecommunications data have an impact on travel behavior

2- Transport demand models continue to be built on the poorly understood new social media structure.

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

- **Casandra**
  - Cassandra does not provide ACID transactions but can be tuned to support "AID" transactions among "ACID" because it does not support foreign keys which means that data will be eventually consistent.
  - But Cassandra let the users control their consistency Level by letting them decide whether some requests would be completed on only one node, or waiting until all nodes respond.
  - and also, Cassandra supports BASE transactions.

- **MongoDB**
  - MongoDB has always provided transactional guarantees on single-document operations, and nowadays with the release of MongoDB 4.0, it's now supporting Multi-document ACID transactions with snapshot isolation.
  - and also, MongoDB supports BASE transactions.

I. 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 to capture data from the IoT devices**

   The NoSQL database selected is MongoDB.

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

   We selected MongoDB for the following reasons:

   - There is no need to define schemas first so it can receive data from different IoT devices with different data structures with no need to define a schema.

   - MongoDB has the ability to scale out fast and inexpensive as we collect data from millions of IoT devices it will be there need to scale out.

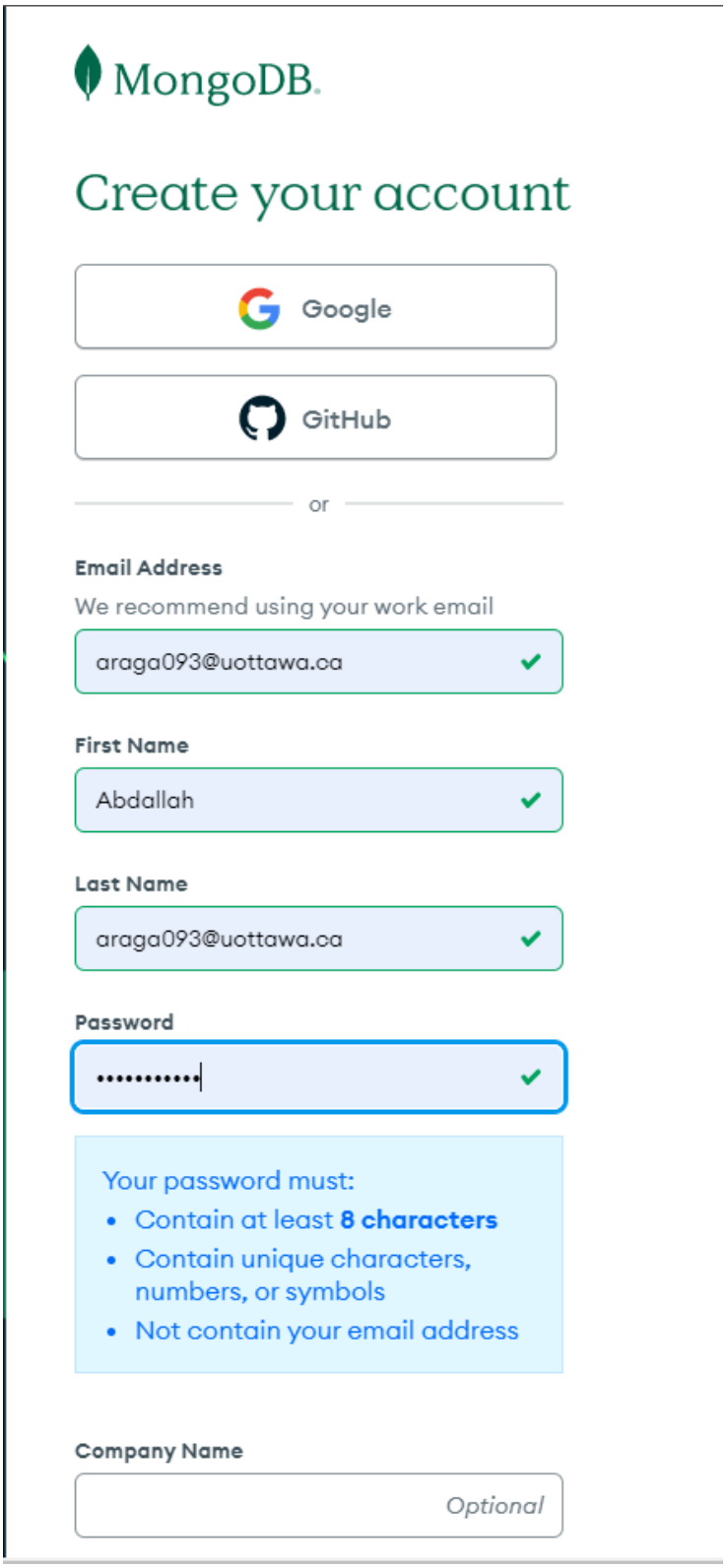
   - MongoDB provides the ability to perform analysis in real-time without affecting the availability of the database

   - Also MongoDB provides replication for the data so it doesn't have a single point of failure.

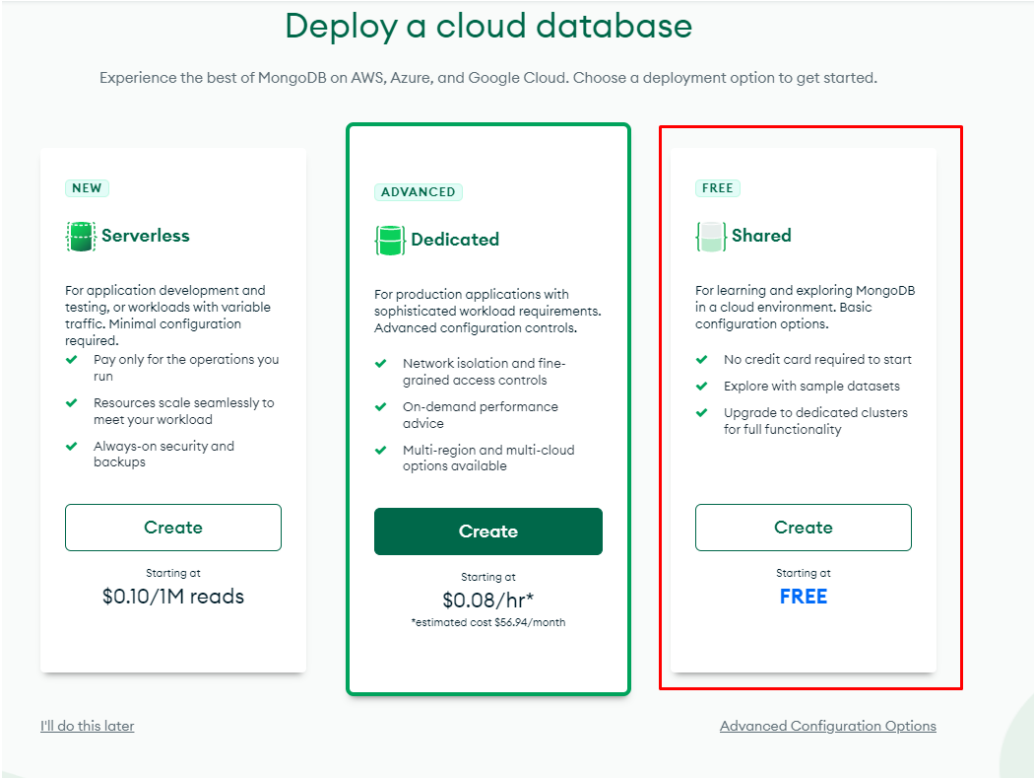
B. MongoDB Lab

I. Create an Account

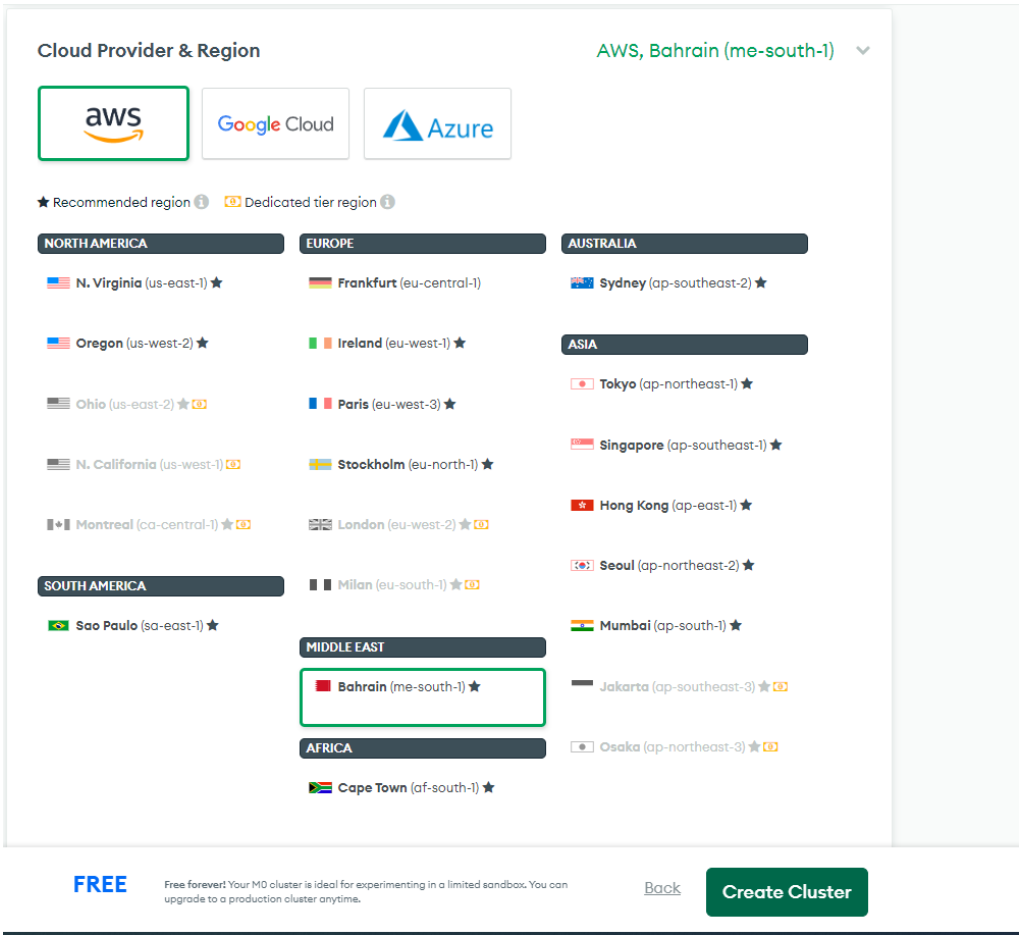
First, we have created an account via https://account.mongodb.com/account/register.



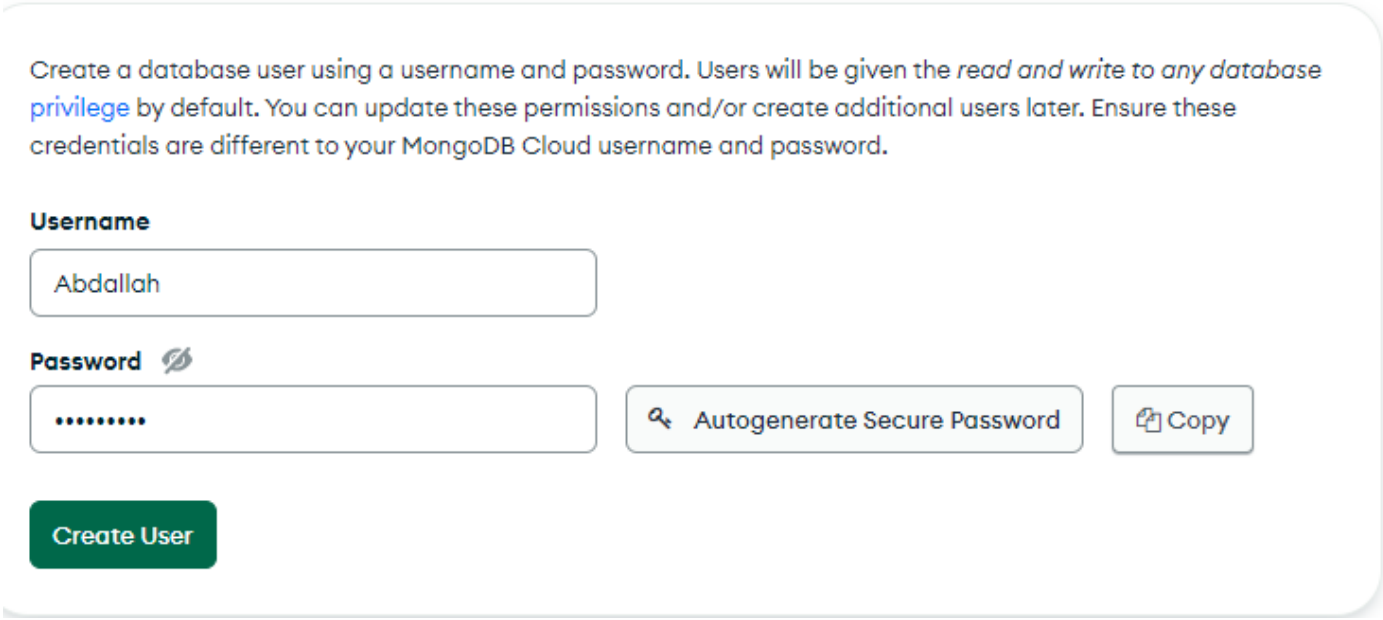
- After verification of the email we created the cluster.



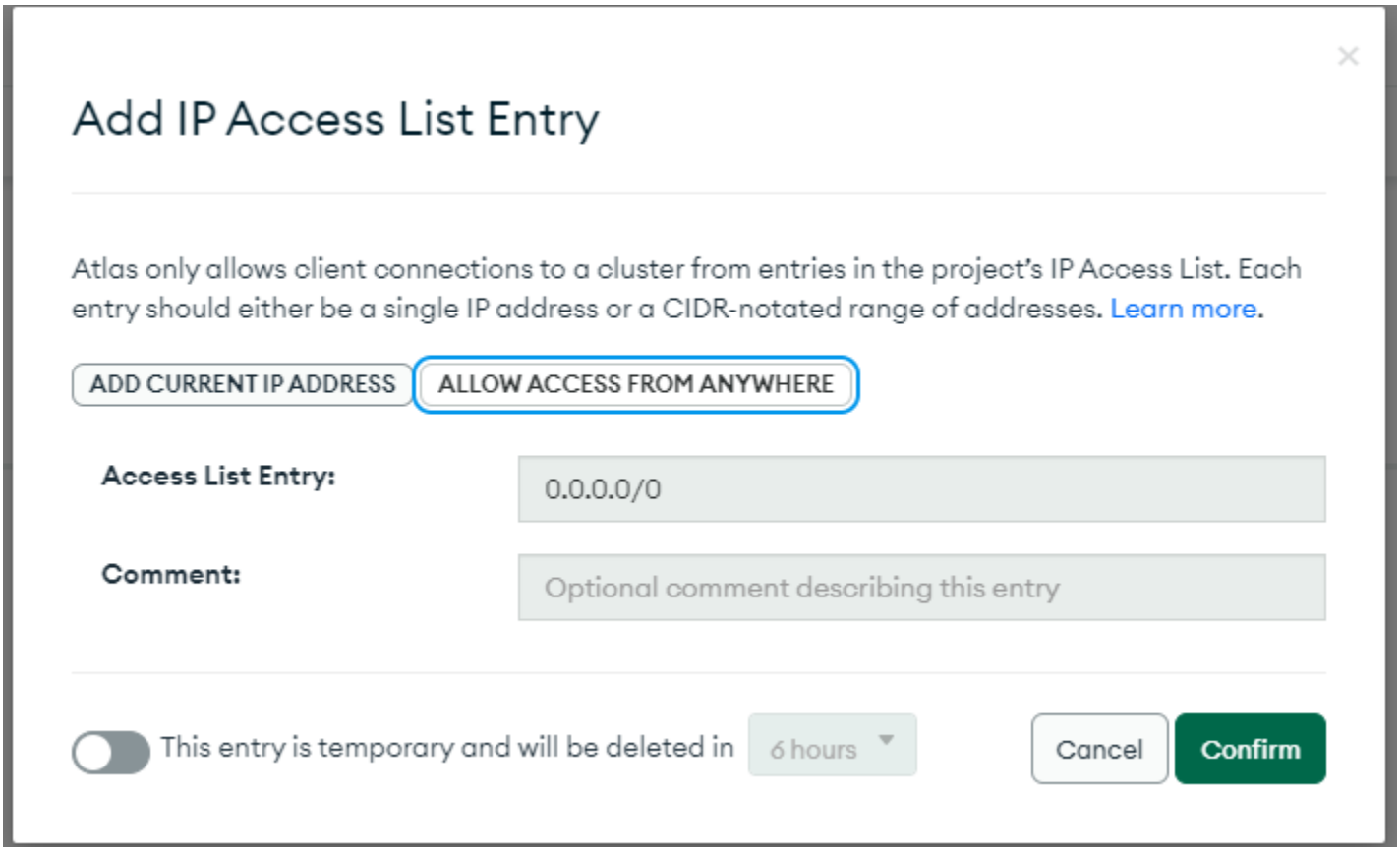
- We selected AWS as the cloud provider and Bahrain region as shown below.



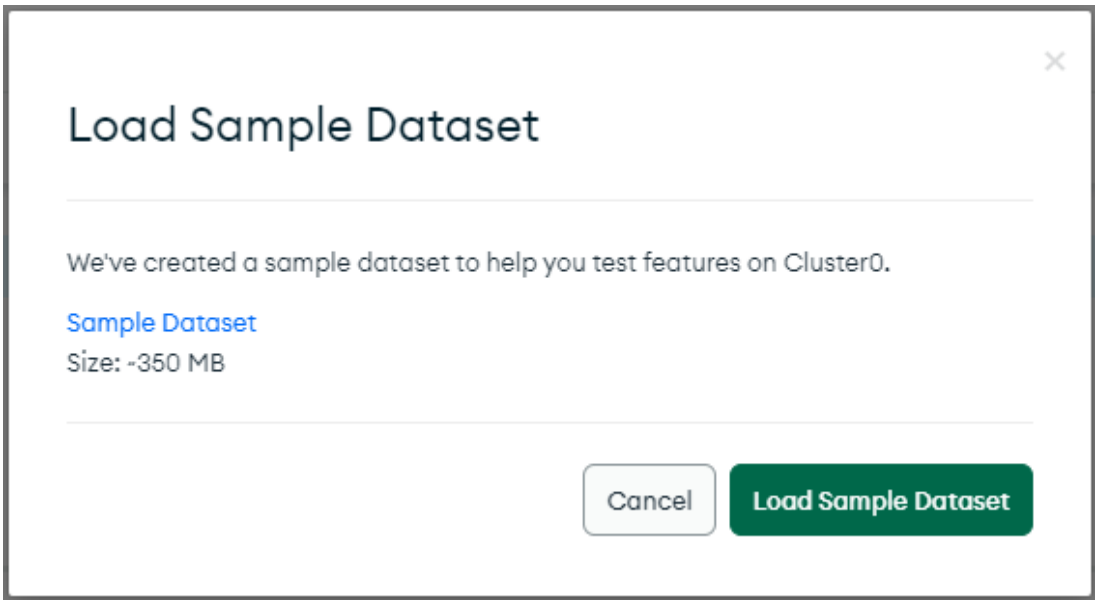
- Create a database username and password.



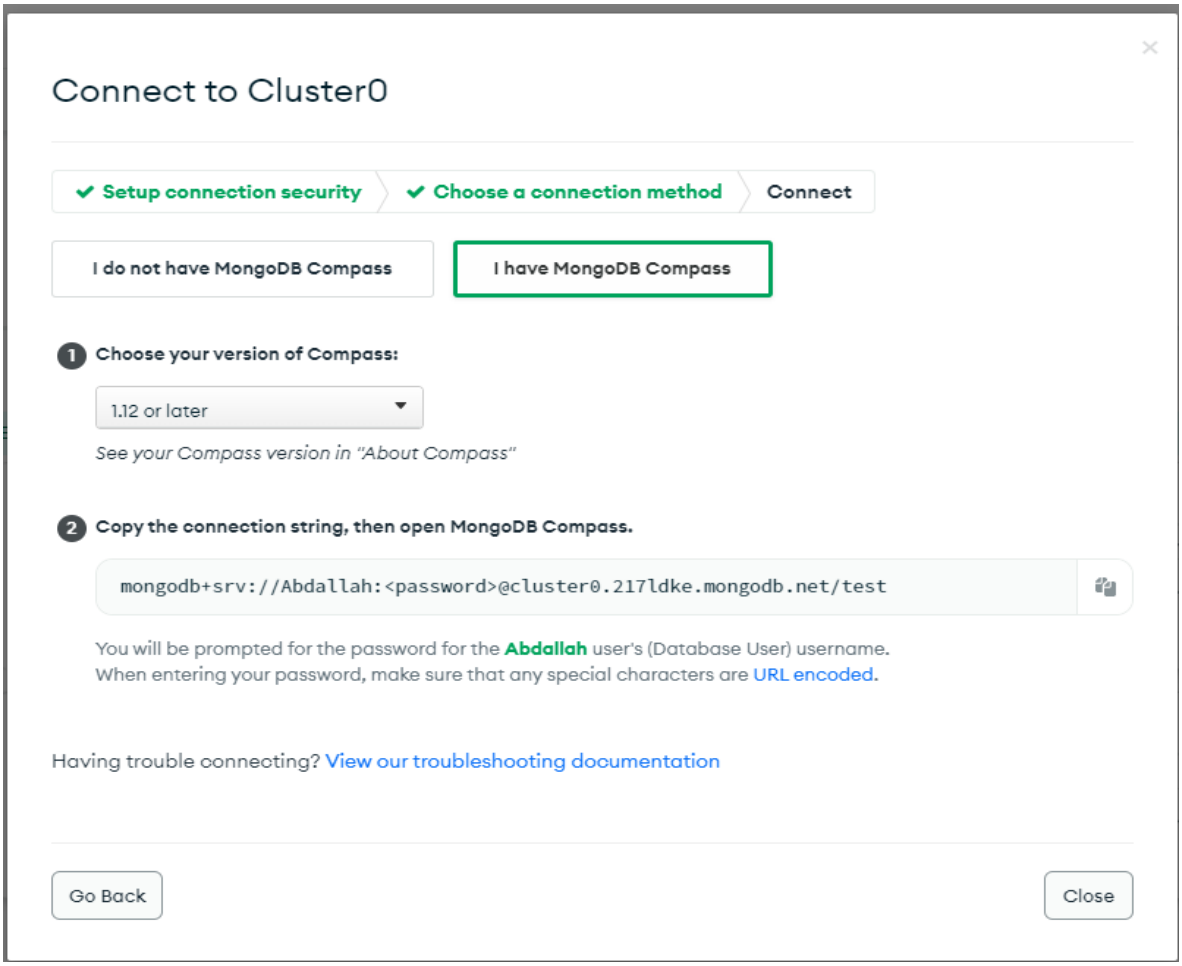
- Edit database network access to allow access from anywhere.



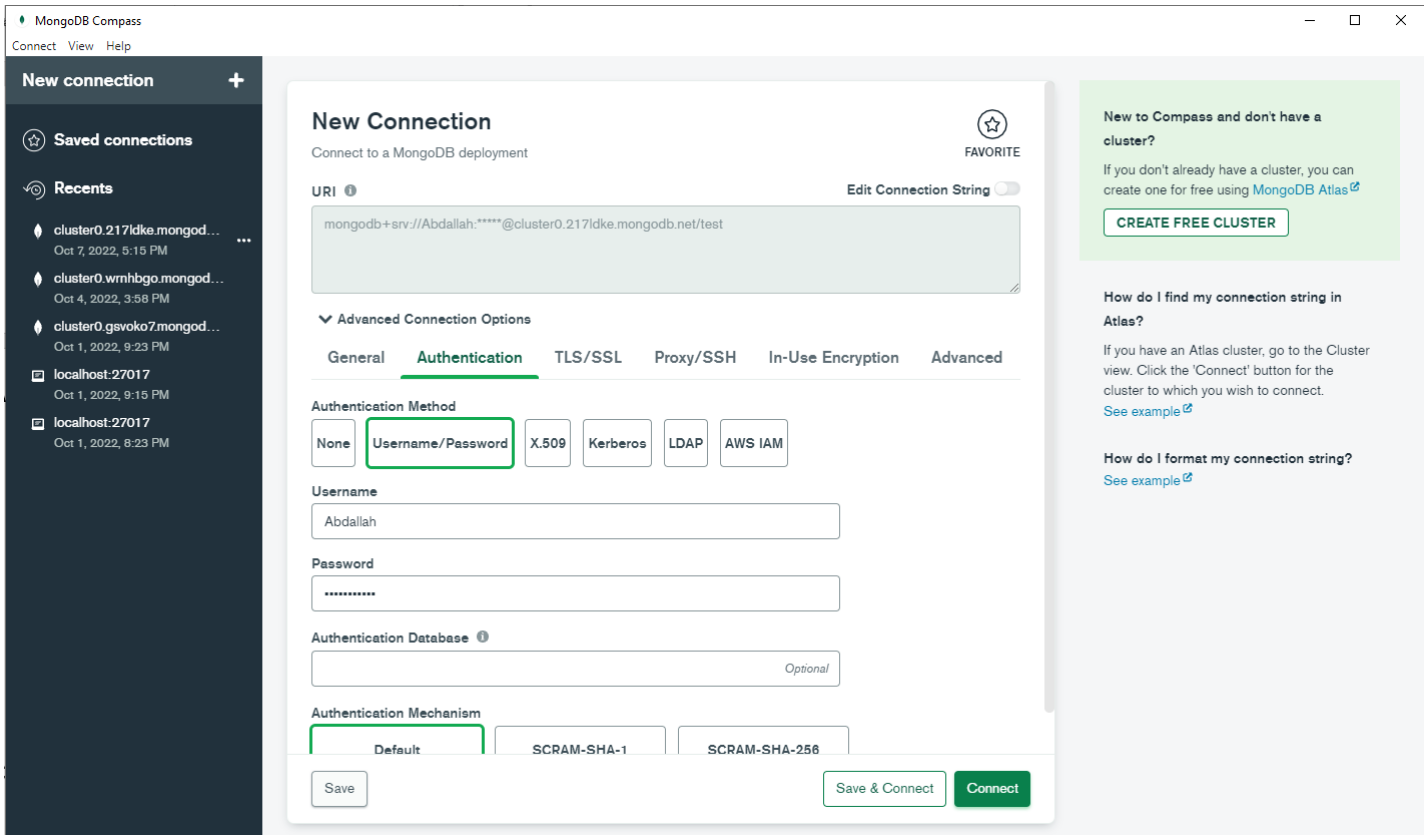
- After creating the cluster successfully, we loaded the sample database Mflix.



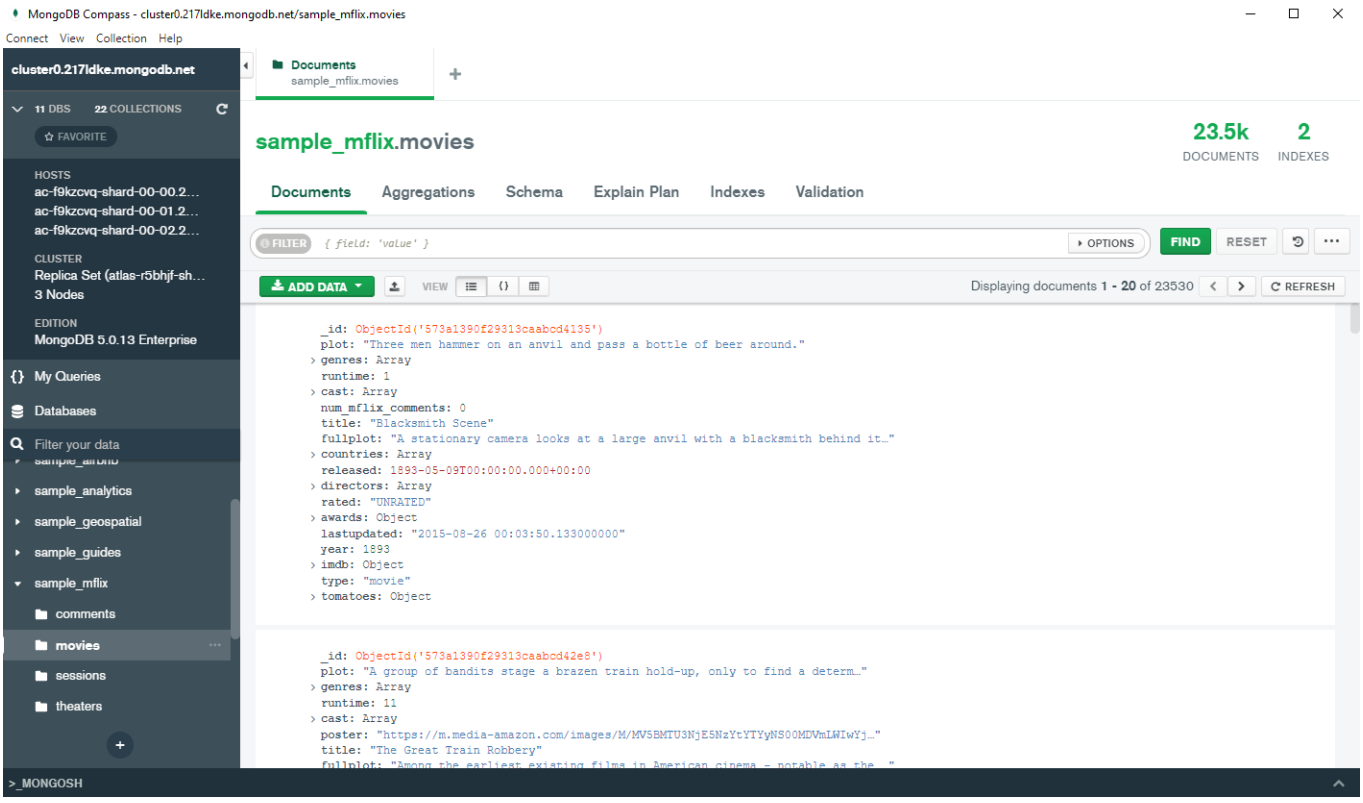
- Copy our database connection string to connect to it from MongoDBCompass.



- After that we downloaded and setup the MongoDBCompass program and then connect it to our database.



- Select the sample_mflix.movie to write our query



II. Briefly describe the movies database document model

Database attributes:

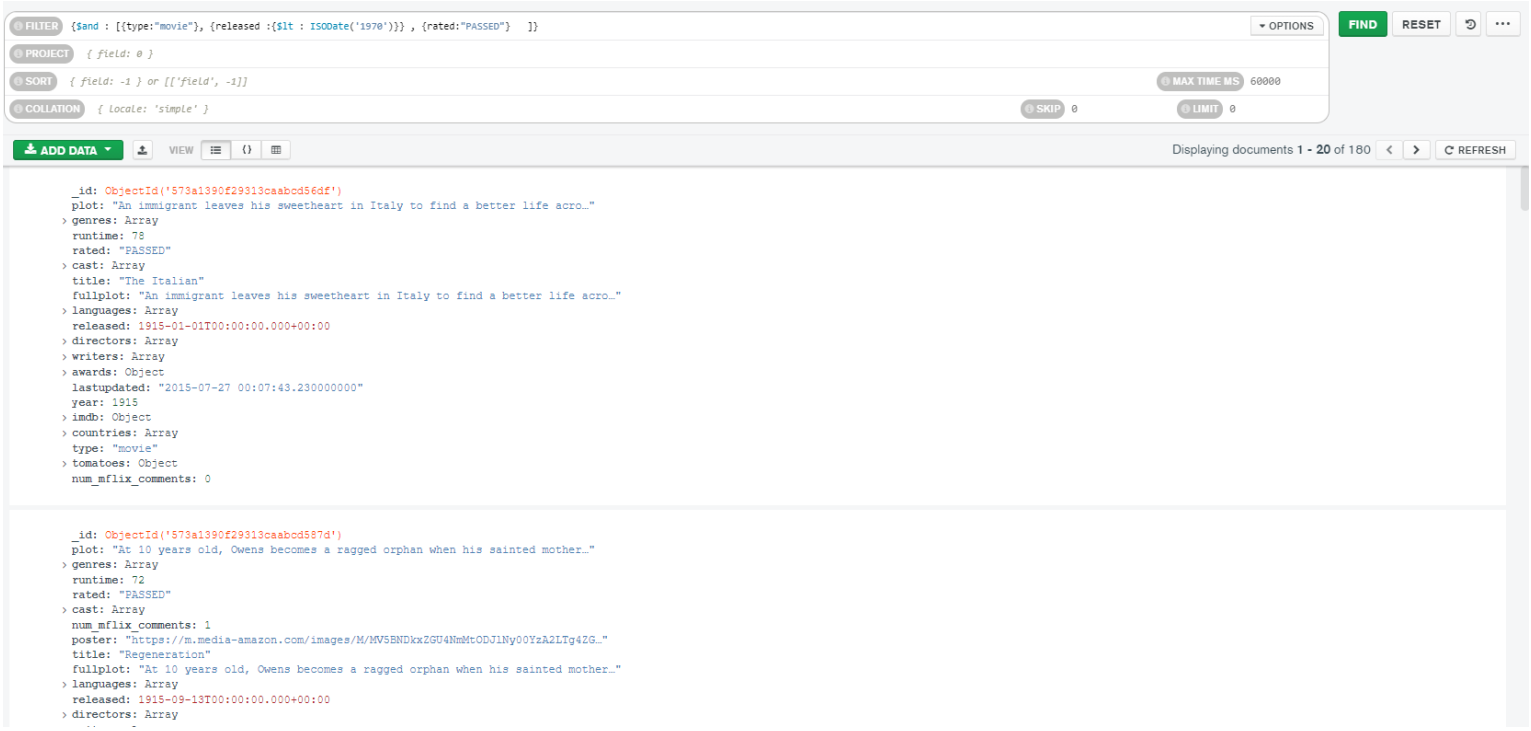
- **_id:** unique field for each file type objectId
- **Awards:** document type consist of 3 fields:
  - **nominations:** type int32
  - **text:** type string
  - **wins:** type int32
- **cast:** type array
- **countries:** type array
- **directors:** type array
- **fullplot:** type string
- **genres:** type array
- **imdb:** document type consist of 3 fields:
  - **id:** type int32
  - **rating:** type double
  - **votes:** type int32
- **languages:** type array
- **metacritic:** type int32
- **num_mflix_comments:** type int32
- **plot:** type string
- **poster:** type string
- **rated:** type string
- **released:** type string
- **runtime:** type string
- **title:** type string
- **tomatoes:** document type consist of 2 fields:
  - **boxOffice:** type string
  - **consensus:** type string
- **type:** type string
- **writers:** type array
- **year:** type int32

III. Filter the documents for type “movies” that are released before 1970 and rated as “PASSED”

- Query:

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

  The number of results: 180



IV. Build an Aggregation Pipeline that shows all entries of type movie that have won at least one award and return the release year aggregate counts.

- First stage
  $match

  ```
  {
   type:"movie"
  }
  ```

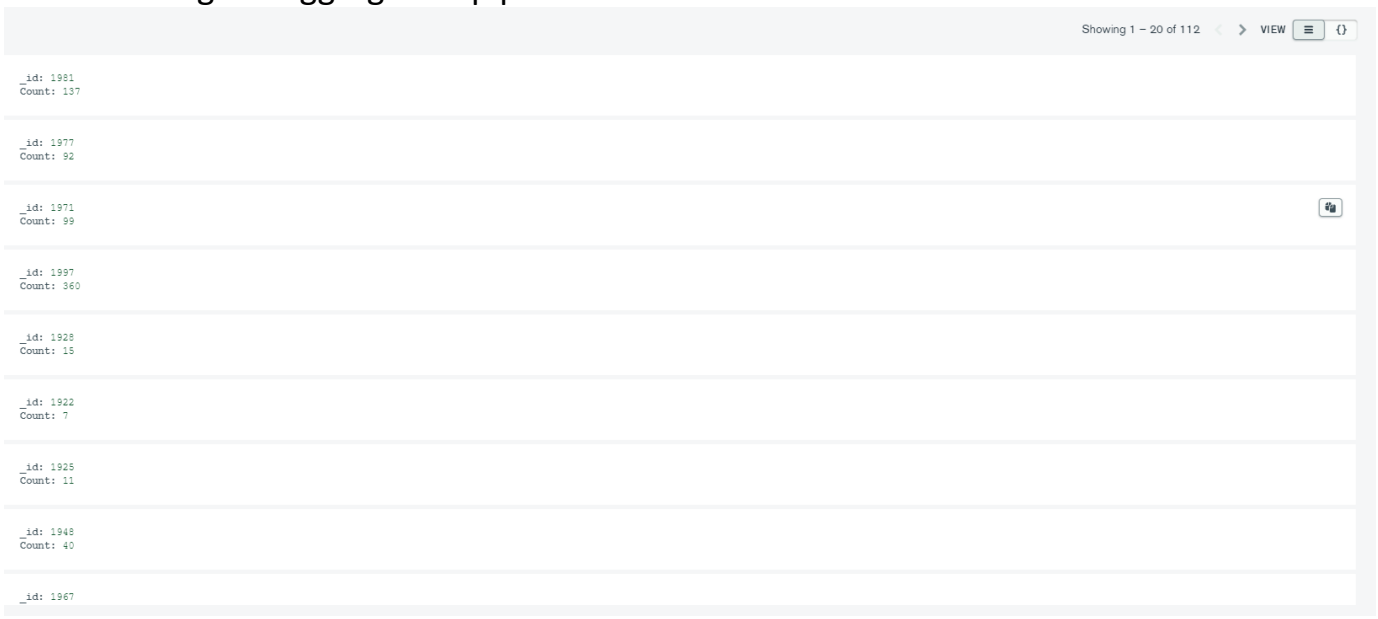
- Second stage
  $match

  ```
  {
    'awards.wins' : {$gte : 1}
  }
  ```

- Third stage
  $group

  ```
  {
    _id: "$year",
    Count: { $count: {} }
  }
  ```

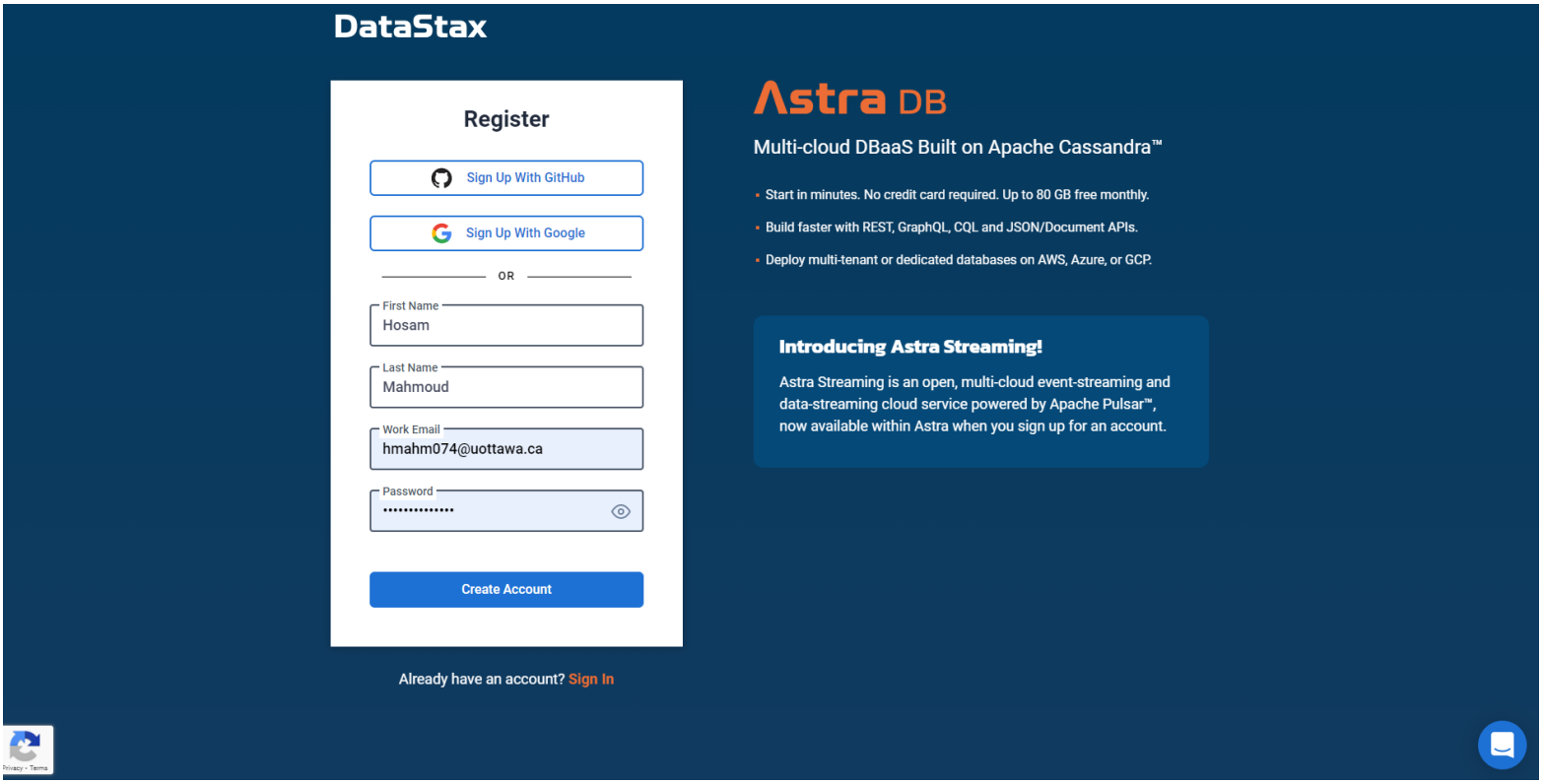
- After Running the aggregation pipeline

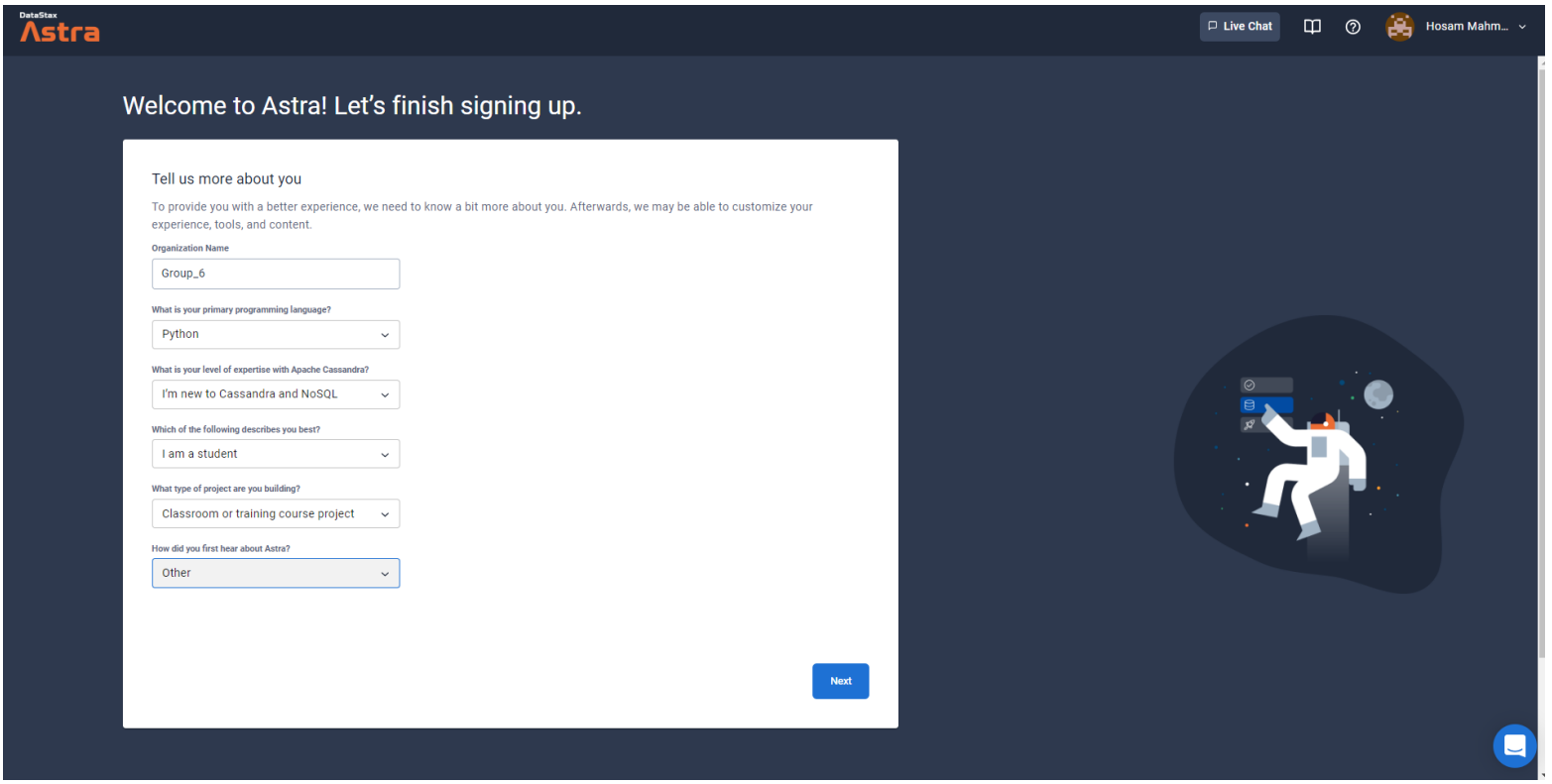


II. Cassandra Lab

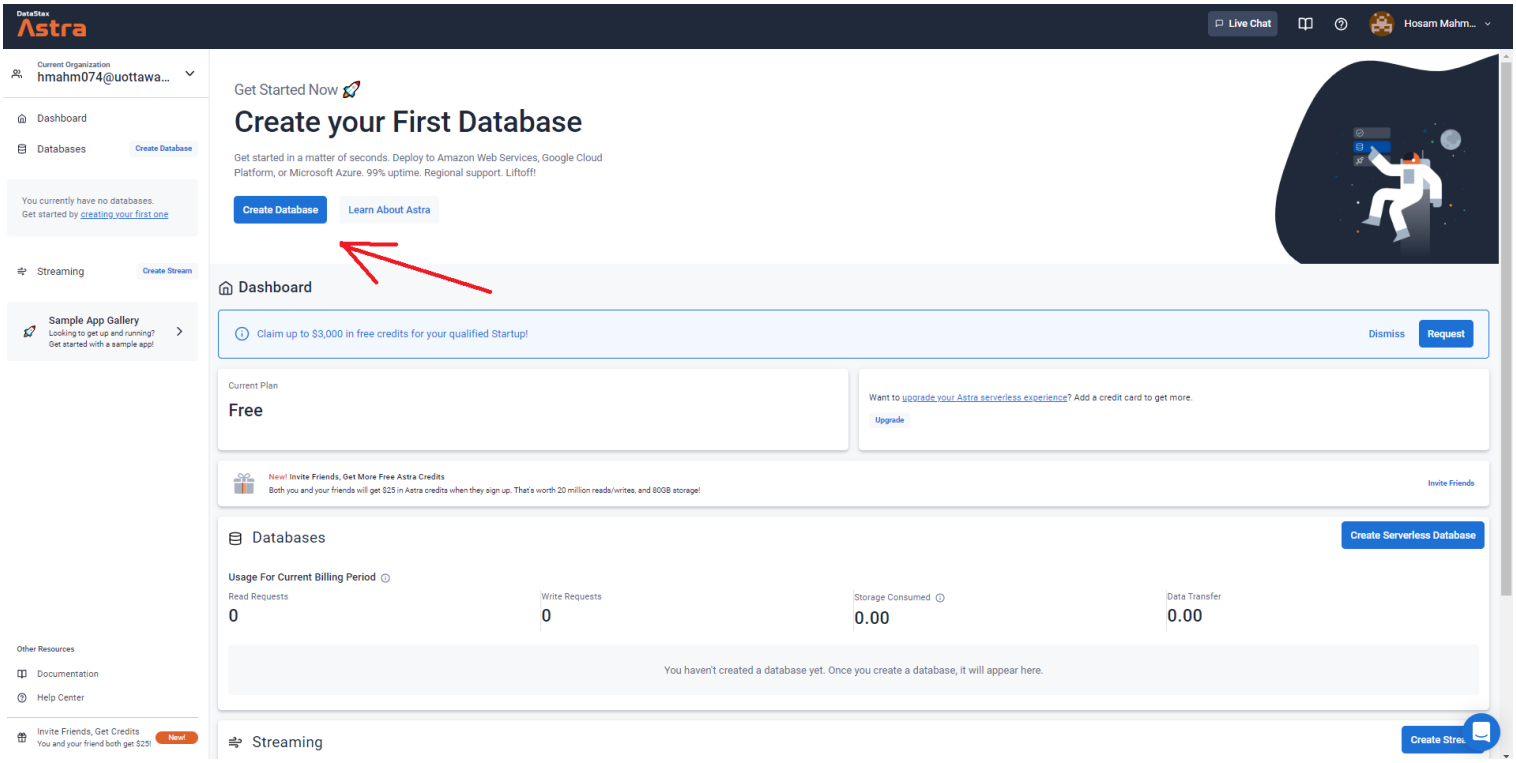
i. Setup

- First, we have created an account on https://astra.datastax.com/.

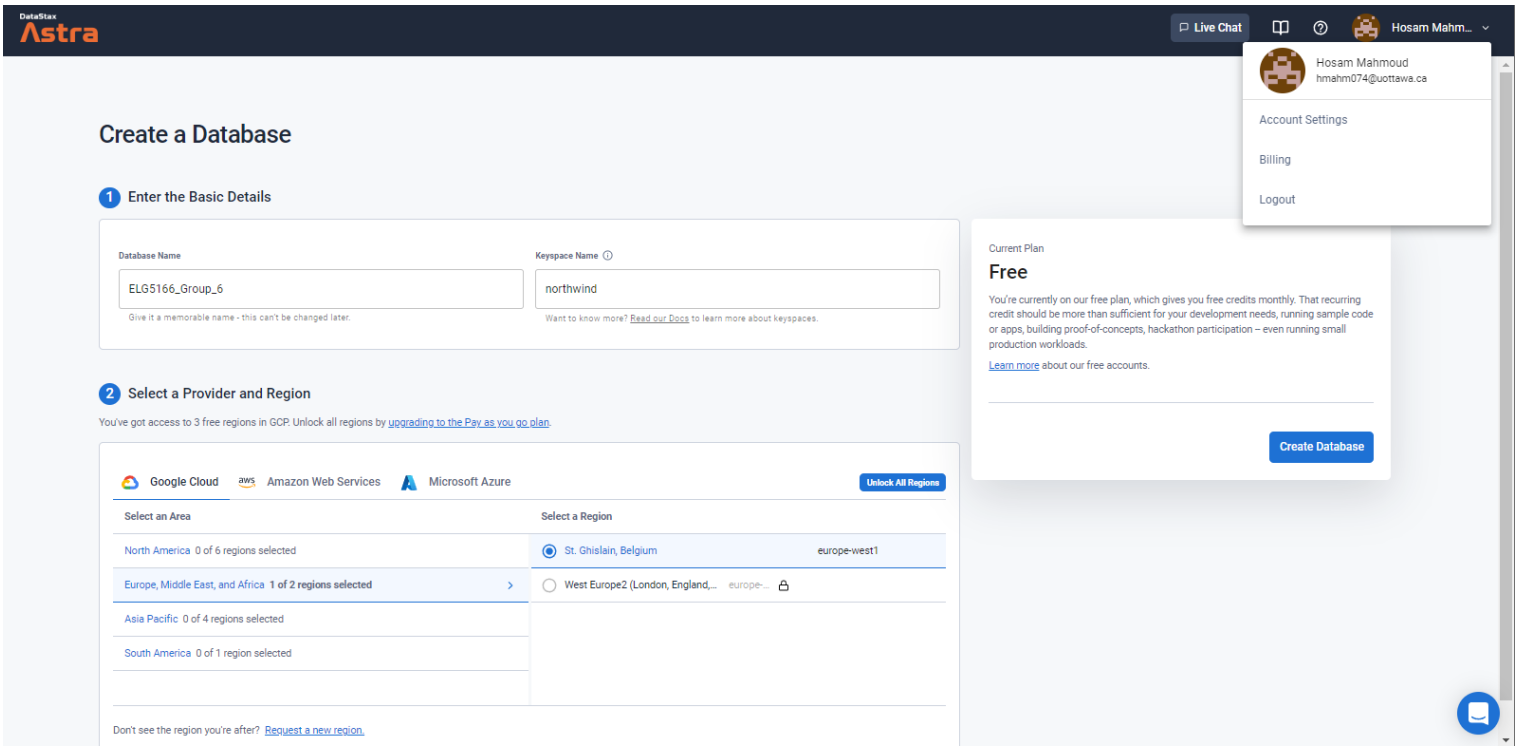




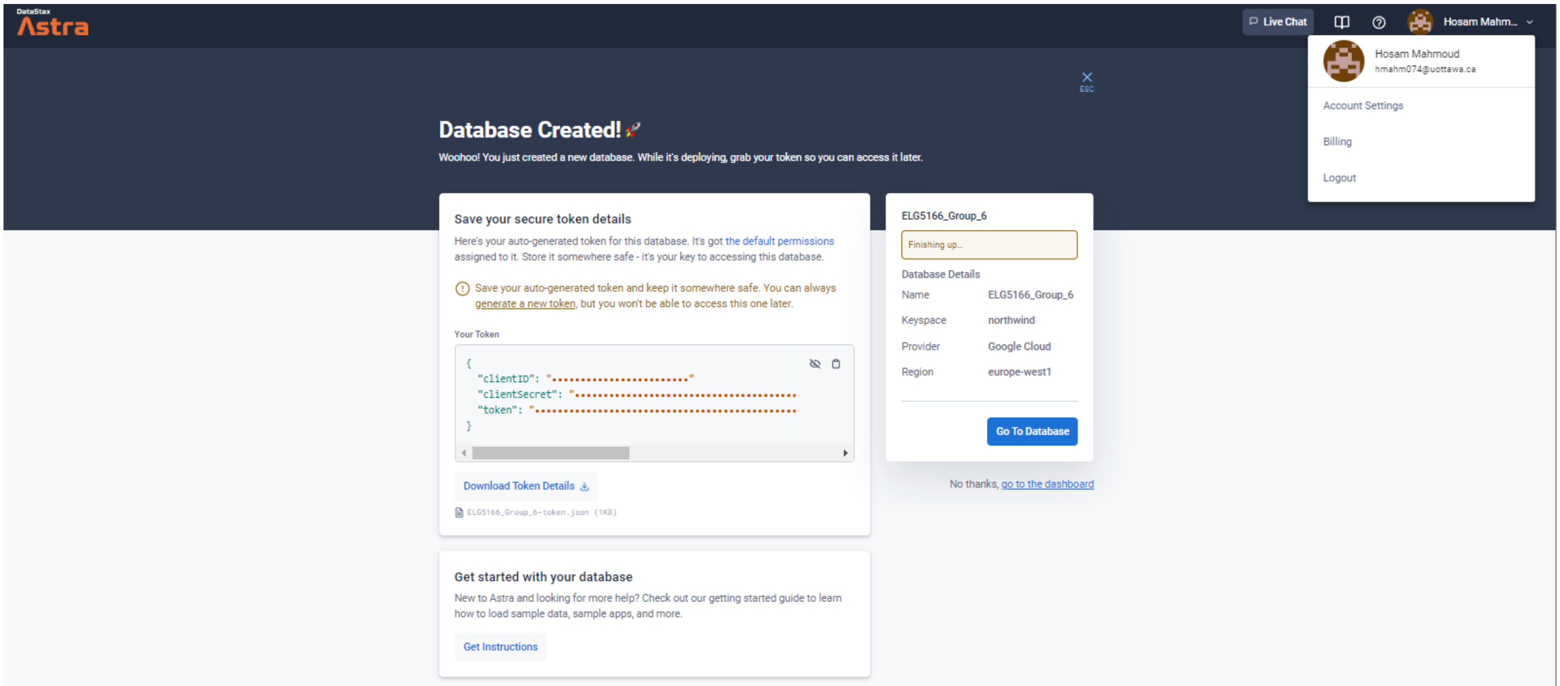
- Then, we created our database by clicking here.



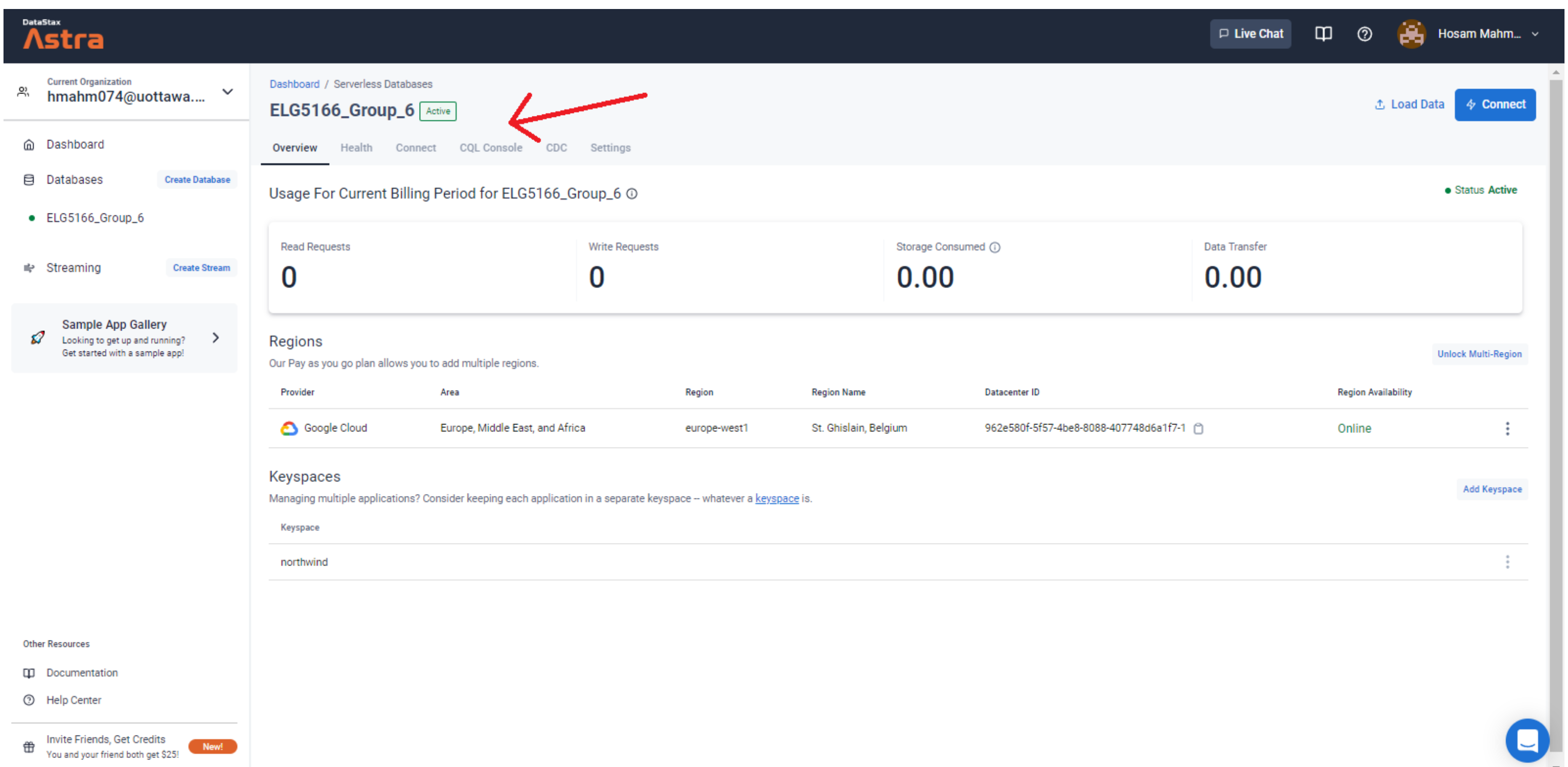
- After that, we have chosen our **database** name, **Keyspace** name, the **cloud provider**, and the **region**.



- Database created successfully.

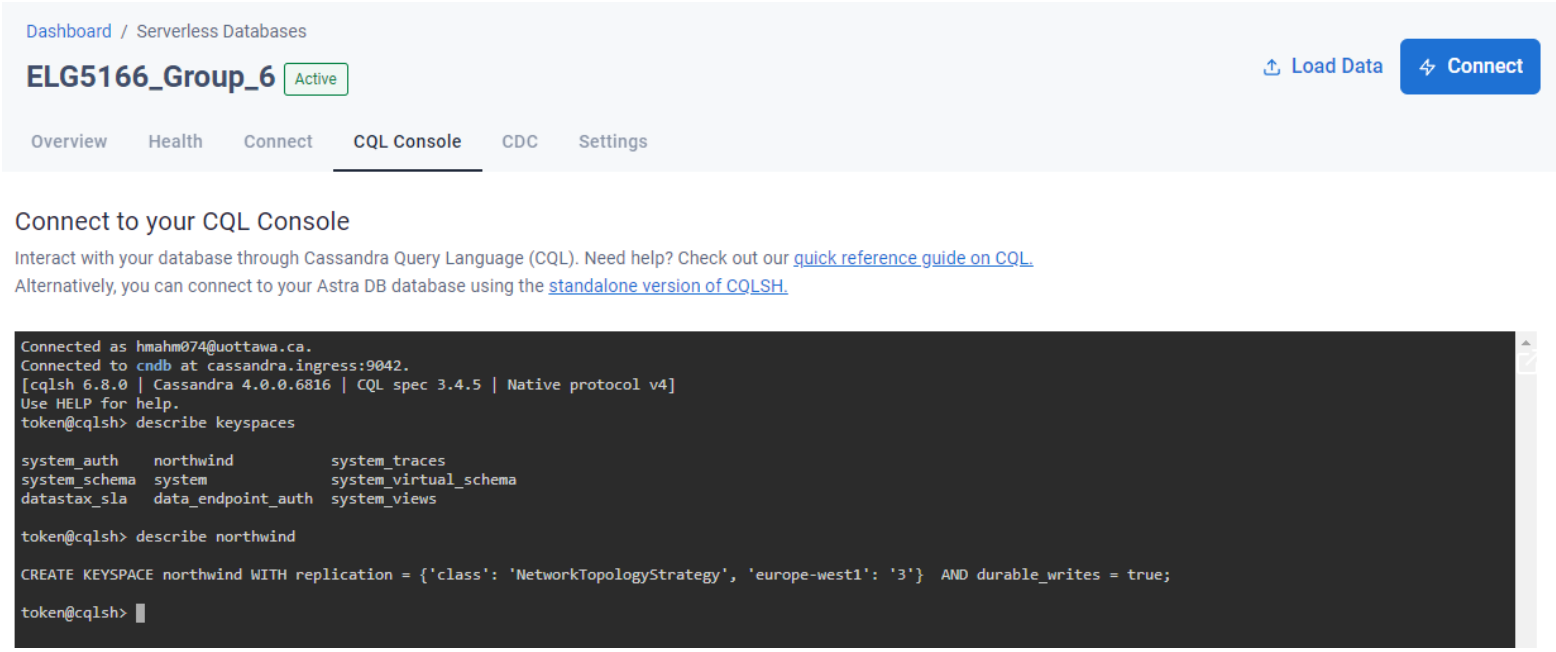


- Now to be able to access Casandra Query Language we clicked here.



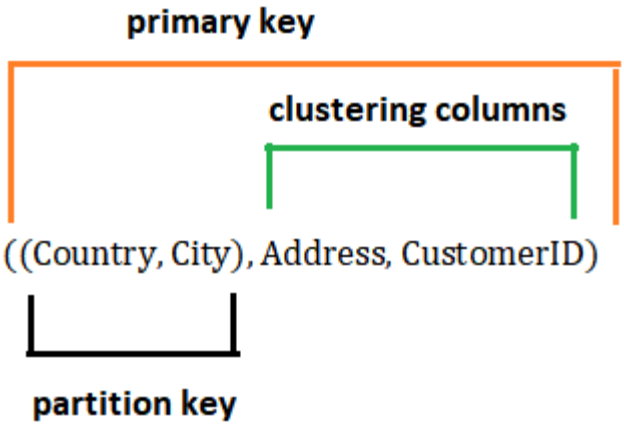
ii. Queries

- To see what is the keyspaces that we have created, we used the following commands:
  - describe keyspaces
  - describe northwind

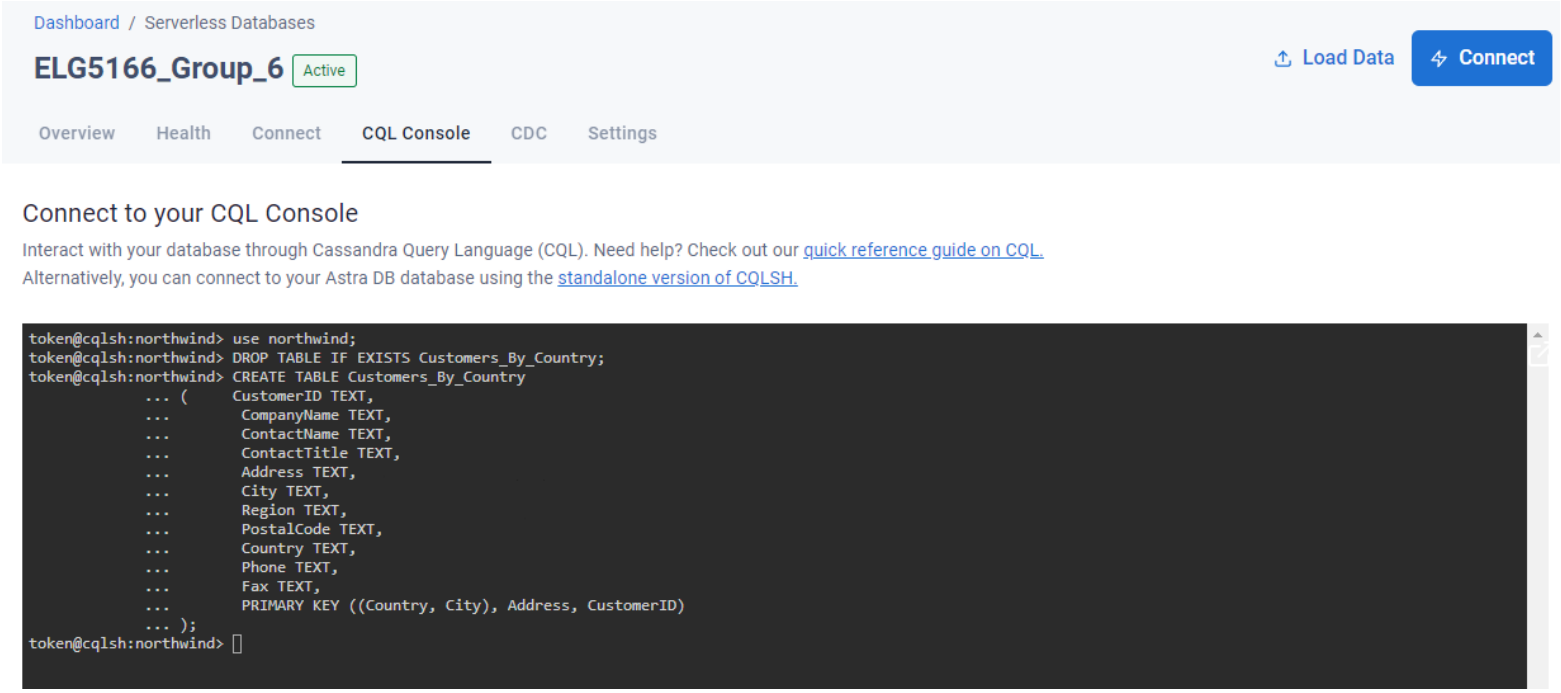


- here we have used the following commands:
  - use northwind;
  - DROP TABLE IF EXISTS Customers_By_Country;
  - CREATE TABLE Customers_By_Country
    (CustomerID TEXT,
      CompanyName TEXT,
      ContactName TEXT,
      ContactTitle TEXT,
      Address TEXT,
      City TEXT,
      Region TEXT,
      PostalCode TEXT,
      Country TEXT,
      Phone TEXT,
      Fax TEXT,
      PRIMARY KEY ((Country, City), Address, CustomerID)
    );

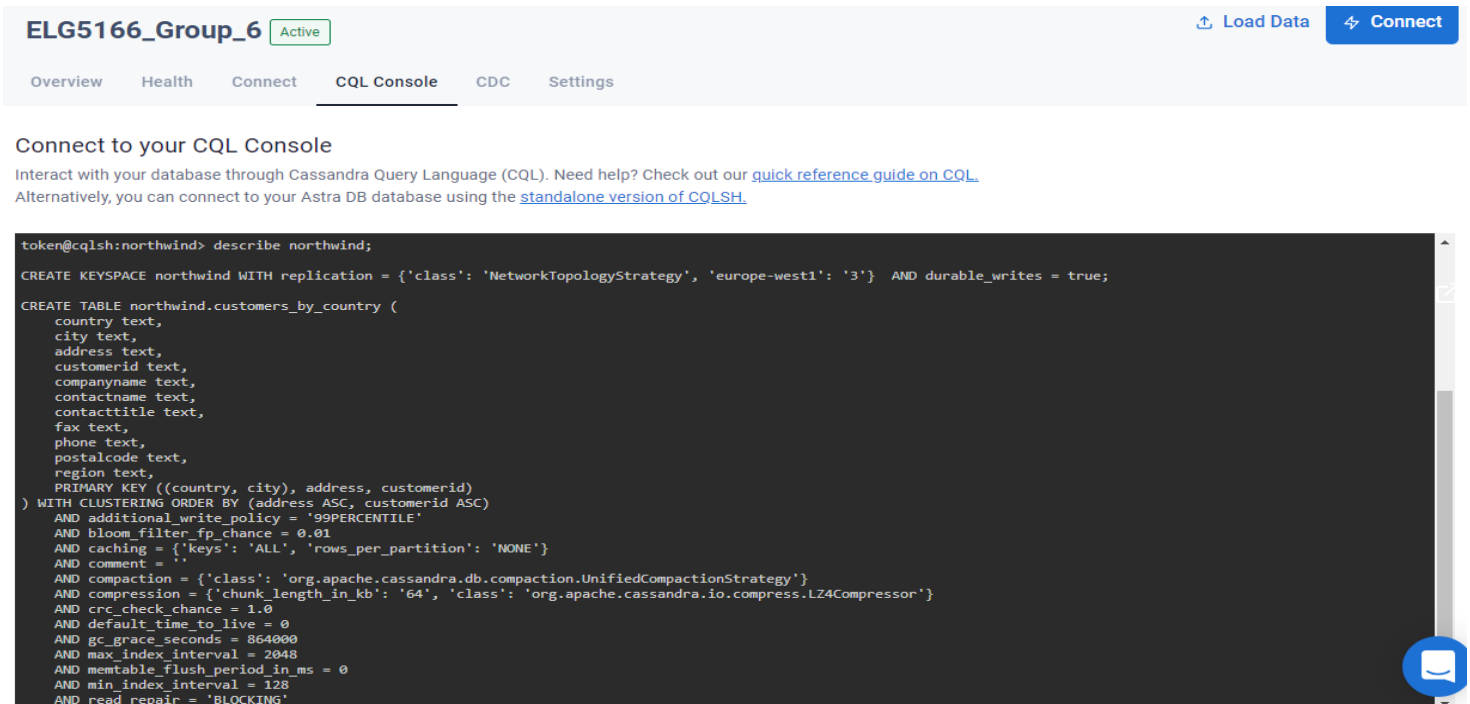
1. We have used the first command so that we don't have to write northwind each time we would like to create a table or drop a table.
2. And for the second command is to ensure that there is no table named “Customers_By_Country” in our keyspace.
3. And for the third command is to create our first table which is called “Customers_By_Country”.



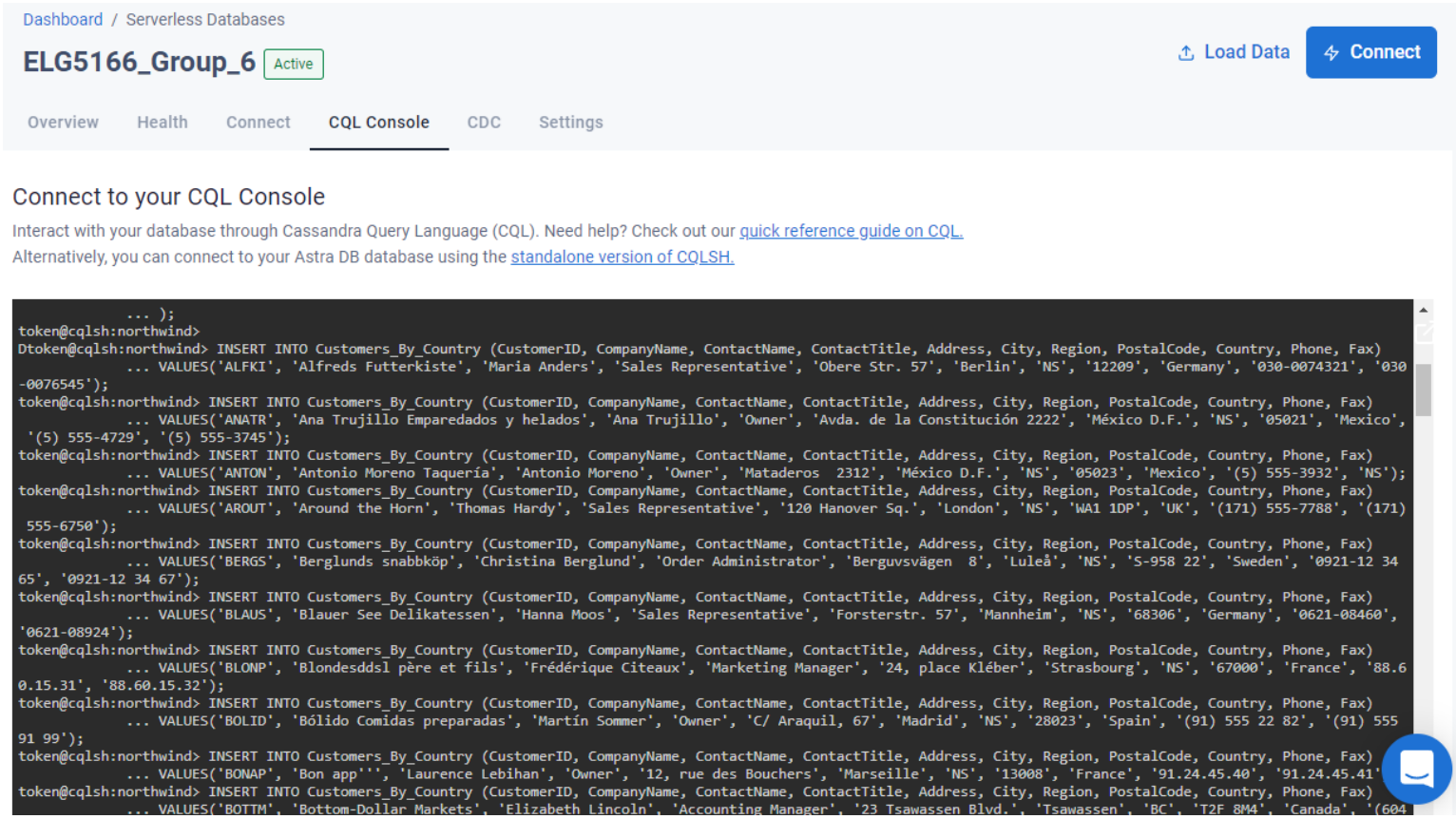
- here we have set “Country” as the main partition and “City” as a sub partition to be able to retrieve the required data fast, without having to use “ALLOW FILTERING”.
- And we have set “Address” as a clustering column to sort the customer’s addresses by their country and city (partition key).
- And also, we have set “CustomerID” as a clustering column to ensure that this table primary key’s is unique.



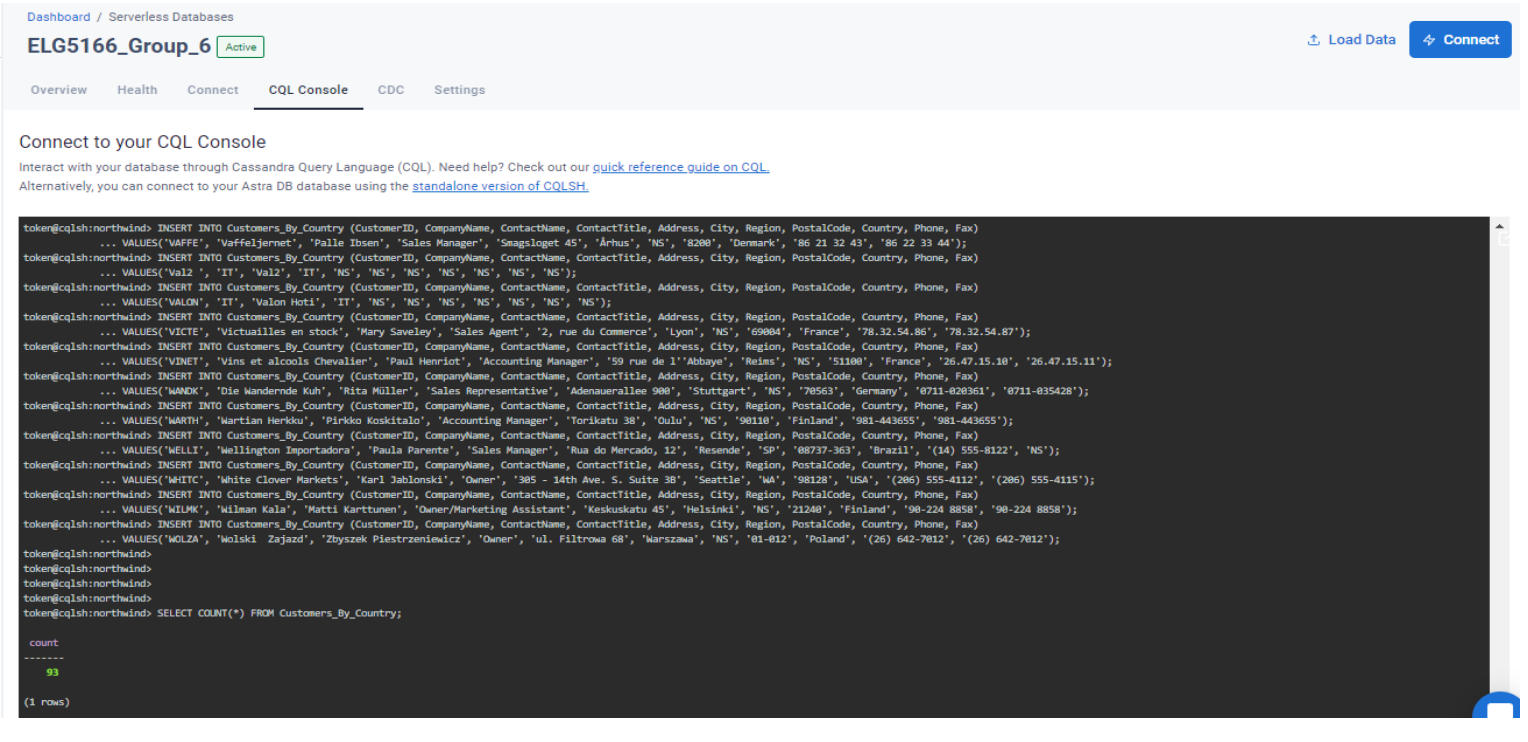
- To ensure that our table is created we used this command:
  - Describe northwind;



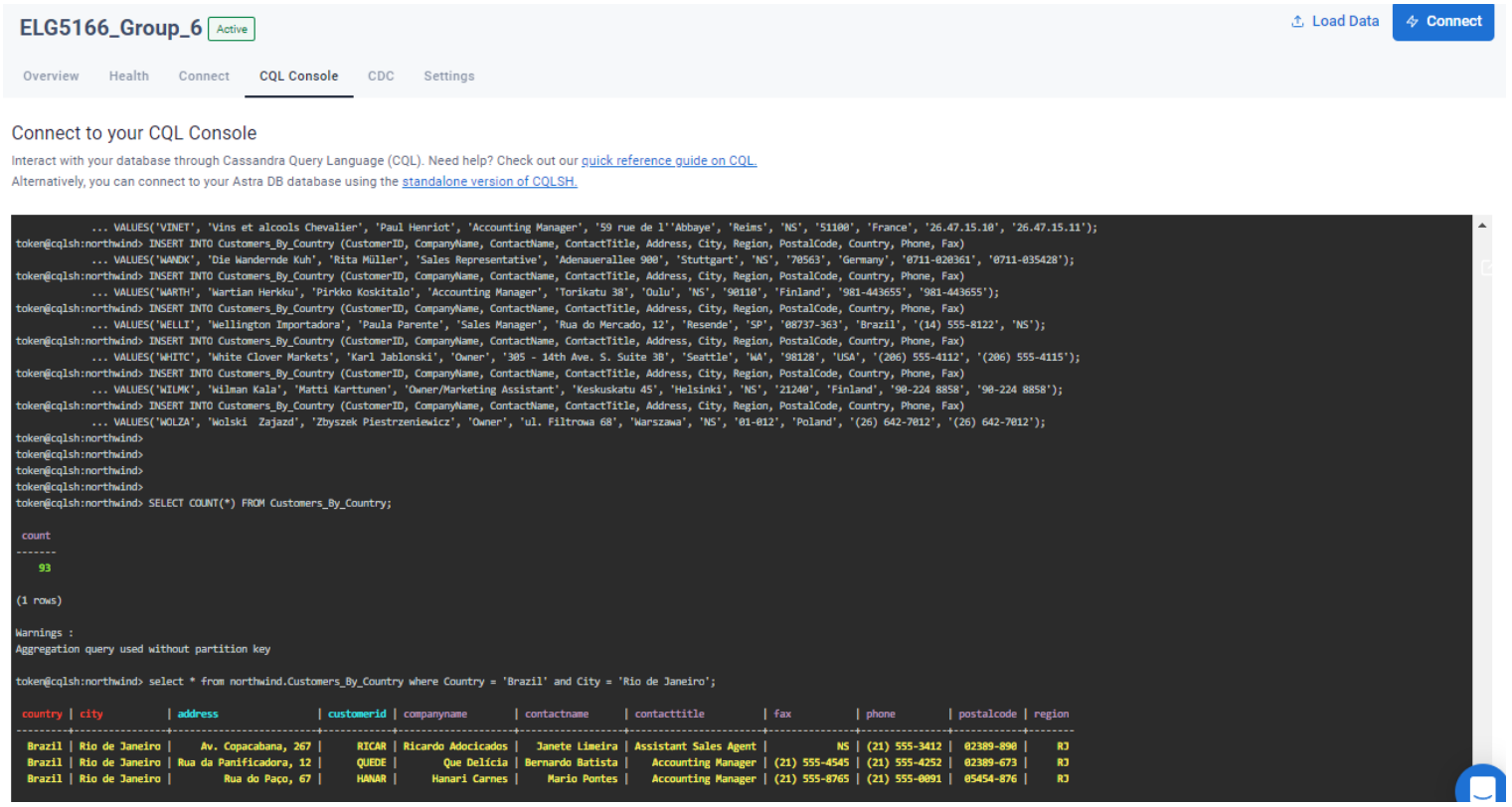
- To insert some data in that table we used commands like this:
    - INSERT INTO Customers_By_Country (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES('ALFKI', 'Alfreds Futterkiste', 'Maria Anders', 'Sales Representative', 'Obere Str. 57', 'Berlin', 'NS', '12209', 'Germany', '030-0074321', '030-0076545');

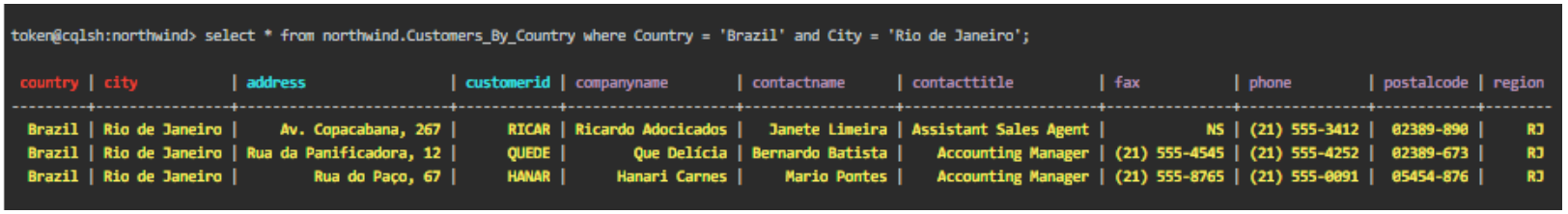


- To get the number of records in this table we have used this command:
    - SELECT COUNT(*) FROM Customers_By_Country;



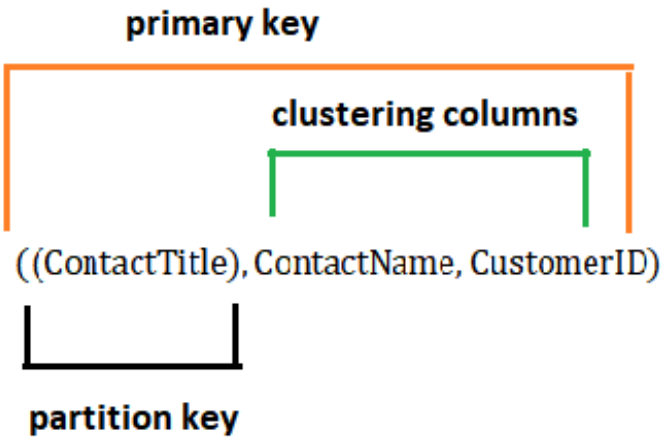
- To get the customers that are from Rio de Janeiro, Brazil, and ordered by their addresses, we have used this command.
    - select * from northwind.Customers_By_Country where Country = 'Brazil' and City = 'Rio de Janeiro';
- we did not need to use the “ALLOW FILTERING” command, because of calling the partition key.





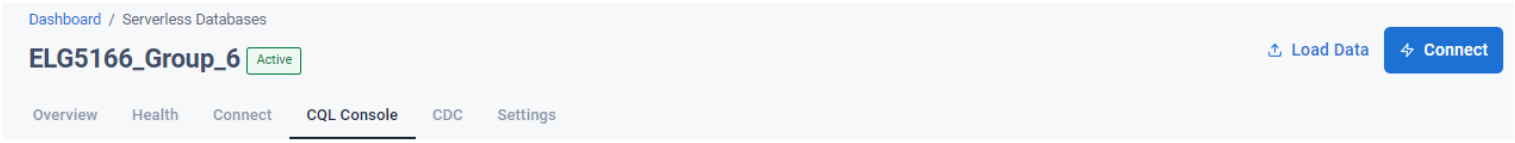
- for the second query we have created another table to partition the customers by their titles, we used these commands:
    - DROP TABLE IF EXISTS Customers_By_title;

- CREATE TABLE Customers_By_title
  (CustomerID TEXT,
     CompanyName TEXT,
     ContactName TEXT,
     ContactTitle TEXT,
     Address TEXT,
     City TEXT,
     Region TEXT,
     PostalCode TEXT,
     Country TEXT,
     Phone TEXT,
     Fax TEXT,
     PRIMARY KEY ((ContactTitle), ContactName, CustomerID)
  );



- here we have set “ContactTitle” as the only main partition to be able to retrieve the required data fast, without having to use “ALLOW FILTERING”.
- And we have set “ContactName” as a clustering column to sort the customer’s names by their title (partition key).

And also, we have set “CustomerID” as a clustering column to ensure that this table primary key’s is unique.

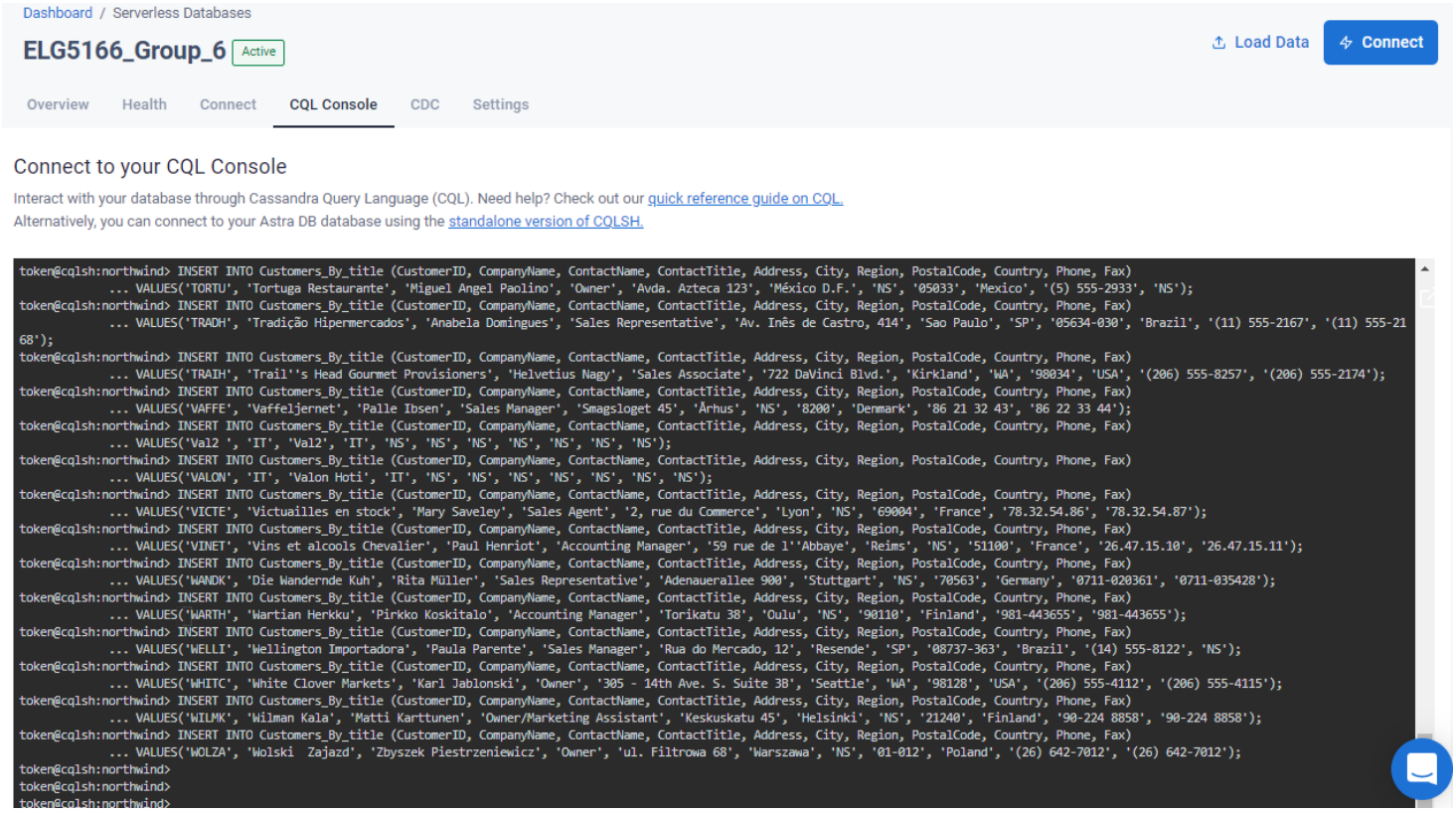


Connect to your CQL Console

Interact with your database through Cassandra Query Language (CQL). Need help? Check out our quick reference guide on CQL.
Alternatively, you can connect to your Astra DB database using the standalone version of CQLSH.

```
Connected as hmahm074@uottawa.ca.
Connected to cndb at cassandra.ingress:9042.
[cqlsh 6.8.0 | Cassandra 4.0.0.6816 | CQL spec 3.4.5 | Native protocol v4]
Use HELP for help.
token@cqlsh> use northwind;
token@cqlsh:northwind> DROP TABLE IF EXISTS Customers_By_title;
token@cqlsh:northwind> CREATE TABLE Customers_By_title
            ... (     CustomerID TEXT,
            ...       CompanyName TEXT,
            ...       ContactName TEXT,
            ...       ContactTitle TEXT,
            ...       Address TEXT,
            ...       City TEXT,
            ...       Region TEXT,
            ...       PostalCode TEXT,
            ...       Country TEXT,
            ...       Phone TEXT,
            ...       Fax TEXT,
            ...       PRIMARY KEY ((ContactTitle), ContactName, CustomerID)
            ... );
token@cqlsh:northwind> S
```

- To insert some data in that table we used commands like this:
    - INSERT INTO Customers_By_title (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES('ALFKI', 'Alfreds Futterkiste', 'Maria Anders', 'Sales Representative', 'Obere Str. 57', 'Berlin', 'NS', '12209', 'Germany', '030-0074321', '030-0076545');



- To get the number of records in this table we have used this command:
    - SELECT COUNT(*) FROM Customers_By_title;

Dashboard / Serverless Databases

ELG5166_Group_6 Active

Load Data | Connect

Overview | Health | Connect | CQL Console | CDC | Settings

Connect to your CQL Console

Interact with your database through Cassandra Query Language (CQL). Need help? Check out our quick reference guide on CQL.
Alternatively, you can connect to your Astra DB database using the standalone version of CQLSH.

```
            ... VALUES('VALON', 'IT', 'Valon Hoti', 'IT', 'NS', 'NS', 'NS', 'NS', 'NS', 'NS', 'NS');
token@cqlsh:northwind> INSERT INTO Customers_By_title (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax)
            ... VALUES('VICTE', 'Victuailles en stock', 'Mary Saveley', 'Sales Agent', '2, rue du Commerce', 'Lyon', 'NS', '69004', 'France', '78.32.54.86', '78.32.54.87');
token@cqlsh:northwind> INSERT INTO Customers_By_title (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax)
            ... VALUES('VINET', 'Vins et alcools Chevalier', 'Paul Henriot', 'Accounting Manager', '59 rue de l''Abbaye', 'Reims', 'NS', '51100', 'France', '26.47.15.10', '26.47.15.11');
token@cqlsh:northwind> INSERT INTO Customers_By_title (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax)
            ... VALUES('WANDK', 'Die Wandernde Kuh', 'Rita Müller', 'Sales Representative', 'Adenauerallee 900', 'Stuttgart', 'NS', '70563', 'Germany', '0711-020361', '0711-035428');
token@cqlsh:northwind> INSERT INTO Customers_By_title (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax)
            ... VALUES('WARTH', 'Wartian Herkku', 'Pirkko Koskitalo', 'Accounting Manager', 'Torikatu 38', 'Oulu', 'NS', '90110', 'Finland', '981-443655', '981-443655');
token@cqlsh:northwind> INSERT INTO Customers_By_title (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax)
            ... VALUES('WELLI', 'Wellington Importadora', 'Paula Parente', 'Sales Manager', 'Rua do Mercado, 12', 'Resende', 'SP', '08737-363', 'Brazil', '(14) 555-8122', 'NS');
token@cqlsh:northwind> INSERT INTO Customers_By_title (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax)
            ... VALUES('WHITC', 'White Clover Markets', 'Karl Jablonski', 'Owner', '305 - 14th Ave. S. Suite 3B', 'Seattle', 'WA', '98128', 'USA', '(206) 555-4112', '(206) 555-4115');
token@cqlsh:northwind> INSERT INTO Customers_By_title (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax)
            ... VALUES('WILMK', 'Wilman Kala', 'Matti Karttunen', 'Owner/Marketing Assistant', 'Keskuskatu 45', 'Helsinki', 'NS', '21240', 'Finland', '90-224 8858', '90-224 8858');
token@cqlsh:northwind> INSERT INTO Customers_By_title (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax)
            ... VALUES('WOLZA', 'Wolski  Zajazd', 'Zbyszek Piestrzeniewicz', 'Owner', 'ul. Filtrowa 68', 'Warszawa', 'NS', '01-012', 'Poland', '(26) 642-7012', '(26) 642-7012');
token@cqlsh:northwind>
token@cqlsh:northwind>
token@cqlsh:northwind>
token@cqlsh:northwind>
token@cqlsh:northwind> SELECT COUNT(*) FROM Customers_By_title;

 count
-------
    93

(1 rows)
```

- To get the customers that are in the Sales Manager role and ordered by their names, we have used this command.
    - select * from northwind.Customers_By_title where contacttitle = 'Sales Manager';
- we did not need to use the “ALLOW FILTERING” command, because of calling the partition key.

Dashboard / Serverless Databases

ELG5166_Group_6 Active

Load Data | Connect

Overview | Health | Connect | CQL Console | CDC | Settings

Connect to your CQL Console

Interact with your database through Cassandra Query Language (CQL). Need help? Check out our quick reference guide on CQL.
Alternatively, you can connect to your Astra DB database using the standalone version of CQLSH.

```
token@cqlsh:northwind> select * from northwind.Customers_By_title where contacttitle = 'Sales Manager';

 contacttitle  | contactname       | customerid | address                | city      | companyname                    | country     | fax            | phone          | postalcode | region
---------------+-------------------+------------+------------------------+-----------+--------------------------------+-------------+----------------+----------------+------------+--------
 Sales Manager |     Annette Roulet |      LAMAI |  1 rue Alsace-Lorraine | Toulouse |              La maison d'Asie |      France |    61.77.61.11 |    61.77.61.10 |      31000 |     NS
 Sales Manager | Art Braunschweiger |      SPLIR |           P.O. Box 555 |   Lander |         Split Rail Beer & Ale |         USA | (307) 555-6525 | (307) 555-4680 |      82520 |     WY
 Sales Manager |       Fran Wilson |      LONEP |     89 Chiaroscuro Rd. | Portland |      Lonesome Pine Restaurant |         USA | (503) 555-9646 | (503) 555-9573 |      97219 |     OR
 Sales Manager |       Georg Pipps |      PICCO |            Geislweg 14 | Salzburg |              Piccolo und mehr |     Austria |     6562-9723 |     6562-9722 |       5020 |     NS
 Sales Manager |        Hari Kumar |      SEVES |        90 Wadhurst Rd. |   London |             Seven Seas Imports |          UK | (171) 555-5646 | (171) 555-1717 |   OX15 4NB |     NS
 Sales Manager |  José Pedro Freyre |      GODOS |           C/ Romero, 33 |   Sevilla |            Godos Cocina Típica |       Spain |             NS | (95) 555 82 82 |      41101 |     NS
 Sales Manager |     Lino Rodriguez |      FURIB | Jardim das rosas n. 32 |   Lisboa | Furia Bacalhau e Frutos do Mar |    Portugal |  (1) 354-2535 |  (1) 354-2534 |       1675 |     NS
 Sales Manager |      Michael Holz |      RICSU |      Grenzacherweg 237 |   Genève |              Richter Supermarkt | Switzerland |             NS |     0897-034214 |       1203 |     NS
 Sales Manager |       Palle Ibsen |      VAFFE |          Smagsloget 45 |    Århus |                    Vaffeljernet |     Denmark |   86 22 33 44 |  86 21 32 43 |       8200 |     NS
 Sales Manager |      Paula Parente |      WELLI |     Rua do Mercado, 12 |  Resende |          Wellington Importadora |      Brazil |             NS | (14) 555-8122 |  08737-363 |     SP
 Sales Manager |      Roland Mendel |      ERNSH |           Kirchgasse 6 |     Graz |                    Ernst Handel |     Austria |     7675-3426 |     7675-3425 |       8010 |     NS

(11 rows)
token@cqlsh:northwind>
```