Jessica Strait
Professor Marc Rigas
DS 220
Project Final Report

Introduction

The first World Happiness Report was released in 2012 by the United Nations Sustainable Development Solutions Network (Cramer). The project utilizes a universal collection of Gallup World Poll questions as well as region-specific questions to quantify an overall happiness score for each country, divided into several contributing factors and a residual; the universal question collection spans a wide range of topics, from personal well-being to financial welfare to the environment ("World Happiness Report"). With the recent outbreak of COVID-19 shaping and shaking the worldview of every nation, analyzing unique aspects of the past few reports can help us better understand what confounding factors might impact the 2020 report. The inconsistency of variables across different years brings up different opportunities for analysis; in one project, the guiding question of happiness across regions also prompted an interest in which regions saw the highest residual value proportion within the overall happiness score (Strait). Results showed that the residual made up almost half of the entire score in African nations- could it be that Western-led data gathering insufficiently inquired about the sources of happiness in these countries? These types of cultural discrepancies may result in disproportionately low category-based scoring for African nations, a concept that would require a much larger scope of data gathering by the United Nations to determine.

Additionally, in the wake of COVID-19, should different universal questions be implemented within the 2020 data? While these questions could not have been compared across years, it may have painted a better picture of citizens' emotional reactions to their countries COVID-19 measures. Despite the United States' prevalence in the socio-scientific community, happiness is rarely measured within the nation to the comprehensive extent at which the UN conducted this study. Professor Catherine Sanderson of Amherst University suggested that the World Happiness Report could serve as an American call to action, stating, "...we should care about happiness, that this is something we should measure" (Cramer). In academia and in politics, it is clear that the happiness of the world's inhabitants is grossly understudied and deserves extensive research, both for the politicians who will shape their nation's future and for the people who deserve complete information to advocate for their country's development. For the task of comparing SQL and NoSQL tools, the true goal was to examine aggregations and querying techniques on specific contributing factors and annual data. Guiding questions included comparing family values across countries, contrasting government corruption and gross domestic product (GDP) per capita, and summarizing regional happiness scores.

Methods

Dataset Content

The dataset includes the 2016 and 2017 editions of the World Happiness Report- because the reported variables change in several editions, choosing sequential editions with similar variables was essential for analysis without extensive data wrangling ("World Happiness Report"). The reports include 13 attributes: Country, Region, Happiness Rank, Happiness Score, Lower Confidence Interval, Upper Confidence Interval, Economy (GDP per Capita), Family, Health (Life Expectancy), Freedom, Trust (Government Corruption), Generosity, and Dystopia Residual. The last seven attributes indicate how much that particular factor contributed to the Happiness Score (reported on a scale of 1-10); however, the Dystopia Residual attribute reflects the difference between the Happiness Score and the other six contributing attributes. This value can be viewed as a cultural source of individual happiness that could not be represented quantitatively in comparison to the scores of other countries. The Sustainable Development Solutions Network uses an imaginary country called "Dystopia" as a reference for the lowest scores within each contributing attribute for a total Happiness Score of 1.85 (with a Dystopia Residual score of 0). This score normalizes the Dystopia Residual value for all other countries, so that it must be positive. There are 157 unique cases (rows) in the data each reflecting a different country.

Retrieving Data

These reports and accompanying qualitative data were available directly from the World Happiness Report website; however, quantitative CSV files are also publicly available and encouraged for download on data-sharing website Kaggle (Sustainable Development Solutions Network). The 2016 and 2017 CSV files were downloaded without difficulty. The straightforward nature of the data made preliminary exploration easy; sorting by Happiness Rank and gaining an understanding of which regions frequently appear at either extreme of the scale could be done within the CSV files prior to importing data. In order to conduct and compare join techniques, some data cleaning was useful to rename data columns such that a simple full join could be conducted as a method of comparison. In addition to this data cleaning method being convenient for conceptualizing a join, it was also mandatory for data importation in the MongoDB Compass interface (Figure 1B - more details below). Finally, all variables were imported as strings, including numerical data such as Happiness Score and each contributing factor. In MongoDB, the documents had to be manipulated to store the values as decimals; in SQL, I was able to use the "CAST" method without completing this wrangling step.

SQL Tool

The Microsoft SQL Server Management Studio was selected as a commonly used and appropriate SQL-based tool with which to conduct the study. Through Penn State's own secure

Winlabs environment, I created a schema for each year's data to be imported as a flat file and enrolled as the owner through the account provided to me by the College of IST (Figure 1A). I then identified "Country" as an appropriate primary key, since each case is represented by one and only one country's data (Figure 2A). My greatest challenge with importing the data in the Microsoft SQL Server Management Studio was running queries on a brand new table; after some online troubleshooting, I realized that the security features of the program were preventing querying. I accessed the "Security" tab and gave my account permission to manipulate the schema (Figure 3A). Creating and assigning appropriate permissions for the schema in addition to actually loading the data required a handful of steps and left room for error, but the data was ready for querying after the security issues were remedied (Figure 4A).

NoSQL Tool

MongoDB was selected as an appropriate NoSQL tool for the analysis of my dataset. A graphical database like Neo4J wouldn't make sense here, because there is no relationship between the countries. A tool like Redis would be incredibly cumbersome to load every country and all of its variables as key-value pairs. MongoDB was an effective tool to include every country as its own document and run queries across the documents. I also selected MongoDB due to its schema-less property, which would make it easy to scale-out and include unstructured data in an expansion of this project. With assistance from lecture materials and course-related resources, I set up an account on MongoDB Atlas and imported the data through the MongoDB Compass interface. I completed the data cleaning referenced above to prevent data importation errors pertaining to the default variable punctuation within the 2017 CSV file (Figure 1B). Because of the schemaless nature of MongoDB, importing the data was significantly less setup-intensive than the Microsoft SQL Server. Automatic sharding negated the need for modifying security features. With the data cleaning done, both datasets were visible and ready for querying in just a few clicks (Figure 2B).

Results

Joins

The first query I chose to use as a comparison method for each tool was a simple "join" operation. Because I had data from two years (and because a larger project would include data from even more years), a join would be the most efficient way to easily compare rankings from each year; as seen in other projects with the same dataset, identifying which countries saw a sharp increase or decrease in happiness within a year is one of the most interesting questions that can be answered in an expanded version of this project (Strait). Microsoft SQL Server Management Studio supported a straightforward join statement, joining the data on the "Country" primary key as intended (Figure 5A). However, MongoDB Atlas produced a very different result when using the $lookup function, which is somewhat equivalent to the left outer join operation in SQL ("$Lookup (Aggregation)"). MongoDB conducts a join by creating a new

variable (in my aggregation, titled “2017Stats”) and saves the joined information from the foreign field into this new variable within the local document; however, because I was joining two complete tables and not just one variable, “2017Stats” became an array within a 2016 case (Figure 3B). While the information is still present and can be queried, navigating an array within a variable rather than having easy access to all original and joined variables as in the SQL program would make large-scale joins across many years of data much less manageable.

Exploring One Contributing Factor

The next metric I compared between both programs was the examination of one single contributing factor; in this case, I sought to select the country, its happiness rank, and order the data by the “Family” contributing factor, descending. In Microsoft SQL Server Management Studio, I used simple “SELECT” and “ORDER BY” queries within my schema (Figure 6A). In MongoDB Compass, I used the “FILTER” interface to project my variables of interest while sorting by “Family” (Figure 4B). The only significant difference between the outputs was MongoDB presenting the object ID by default; I could have chosen to remove this feature in the projection statement. With this query, I learned that Uzbekistan feels very strong family connections, even though other aspects of their Happiness Score result in a lower rank. This gives some insight to the larger question of what contributing factors are most essential in different regions (and how family values differ in global cultures, regardless of non-social contributing factors), and this question was easily answerable with either tool.

Contrasting Two Contributing Factors

Arranging data in accordance to one contributing factor was straightforward with the SQL and NoSQL tool; the next step was to identify a possible correlation between two contributing factors and return affected nations. I chose to contrast a country’s GDP per capita versus trust in government (presented as perception of corruption). In Microsoft SQL Server Management Studio, this query selected the variables of interest but used “CAST” to represent GDP and government trust as floats so that I could compare them to quantitative metrics and order the data by government trust (Figure 7A). In MongoDB Compass, I used the “Filter,” “Project,” and “Sort” tools to complete the same task. The greatest challenge in MongoDB Compass was the lack of a “CAST” equivalent; prior data wrangling was required to yield appropriate results that I was able to query directly from the user interface with my SQL tool (Figure 5B). Overall results of the query showed that Bulgaria and Romania experience positively influential GDP per capita scores, but also present the lowest confidence in government trustworthiness while still performing well economically.

Aggregation

For a final comparison metric between the tools, I chose to conduct a simple aggregation to compute the average happiness score for each region. In Microsoft SQL Server Management

Studio, I was able to use the "AVG" aggregation alongside the "GROUP BY" method. The greatest challenge with the SQL tool was identifying the necessity of the "ROUND" method, since the tool originally produced long decimals that would not be easy to visualize if this data were graphed or otherwise presented directly from the query. Results were produced in descending order by average regional happiness score (Figure 8A). In MongoDB, I returned to the aggregation pipeline builder used in the "join" operation. I used three types of aggregations within the pipeline to complete the task ("Aggregation Pipeline Builder"). First, a "$group" aggregation to compute the averages, in combination with a "$toDecimal" method; secondly, a "$sort" aggregation to present the data in descending order; and finally, a "$project" aggregation to round the averages to two decimal places (Figure 6B). Although the query results were identical between the two tools, MongoDB Compass required more steps and more syntaxical manipulation to produce a full pipeline with three separate aggregations.

Discussion

MongoDB Atlas

For this particular project, MongoDB Atlas was an appropriate NoSQL tool and had some advantages in comparison to the Microsoft SQL Server Management Studio. The horizontal scaling ability of MongoDB would make comparisons across years much quicker and without requiring as much memory (Jayaram). If this dataset were to get extremely large, like if data were collected in every country for the next hundred years or every month instead of every year, MongoDB may be more effective than an SQL program for that volume of information. The automatic sharding (or "horizontal partitioning") feature of MongoDB Atlas also provided an extra layer of security that I did not have to manipulate manually as in the Microsoft SQL Server, which would require extensive permission management if many users were accessing the same schema at different times (Drake). Since the sharding itself required no action on my part, this also prevents potential risk of table corruption that individual sharding errors could produce (Drake). However, MongoDB might be somewhat excessive for the project at this point. It would undoubtedly be useful in the event of scaling out the data or adding unstructured information (perhaps a brief description of a catastrophe, election, or significant contributing factor for certain countries), but for the structured data as of now, the schemaless property of document stores isn't really being used to its full potential. Additionally, the MongoDB Compass interface came with some limitations that will be discussed along with the metrics of the study.

Microsoft SQL Server Management Studio

Using Microsoft SQL Server Management Studio was effective for the type of data I was analyzing; however, it might be less effective if more cases were added. If the data were to be scaled out (across years, I would probably use spread-gather methods to create a "year" variable, such that data could be grouped by year or even by country), an SQL tool would require

significant resources to complete the vertical scaling process in comparison to a tool like MongoDB that could scale horizontally (Jayaram). The dataset I used was very tidy, largely floats or short strings, and without an unreasonable number of variables. Additionally, this database cannot grow indefinitely: data only exists for the last several years, and can only continue to grow by the fixed number of countries on Earth for future research. Scalability is currently not a great concern for this type of limited data, especially when the data structure won't change significantly. If scalability became a priority for the dataset, an SQL tool would not allow for as much flexibility with data types. Additionally, if the UN Sustainable Development Solutions Network wanted to add unstructured data, this would not be easily queried with an SQL tool in comparison to a document store. Nevertheless, the consistency and durability properties are important for this type of data, because ensuring the same data types and permanent transactions are critical for a fairly concrete set of information; a NoSQL should be chosen with caution in regards to eventual consistency when it is clear that the data itself changes infrequently but analysis may be regular.

Comparing Metrics

In addition to comparing these basic principles of SQL and NoSQL tools for the project, discussion of the metrics when executing the guiding questions also demonstrates which tools are most appropriate for answering the questions outlined in the initial project proposal. The join operation was the first and perhaps most telling metric when it came to making this comparison. In an SQL tool, joins for tables are normal and even encouraged due to the relational nature of the data being analyzed. Although I was able to conduct a join-like aggregation in MongoDB Atlas, document stores are not designed to build relational models, and the "$lookup" method created a new array variable within the origin document with the content of the matching foreign document case ("$Lookup (Aggregation)"). Comparing data across years would be much more challenging with this format in comparison to the straightforward outer join method in Microsoft SQL Server Management Studio.

The next two metrics used for comparing the efficiency of the tools involved selecting and manipulating certain contributing factors within the dataset ("World Happiness Report"). Selecting, presenting, and sorting data based on the "Family" variable was straightforward in both tools, implying that single-variable queries can be conducted easily and quickly on either tool; the comparable execution time is likely in part due to the automatic sharding within the MongoDB Atlas program (Drake). However, filtering and contrasting two numeric variables proved a greater challenge. In the SQL tool, these variables could be cast as floats without significant data wrangling; the NoSQL tool would have either required an extensive aggregation pipeline or external data cleaning (which I opted for) to convert each necessary variable across every document. Three separate document methods had to be implemented in addition to two comparison statements, all within the document manipulation section rather than the aggregation

pipeline. Both systems were able to complete this query, but the syntax of the NoSQL method in combination with the necessary data cleaning step became cumbersome.

In regards to the aggregation pipeline, this method became an essential step in computing the average score of each region. The pipeline itself within the free, cloud-based MongoDB Atlas does take time and a lot of syntaxical research to be useful. In comparison to an SQL tool, which can conduct an aggregation and appropriate querying statements all within a few lines, the addition of stages and multiple stage previews significantly slows down the process of actually completing a fairly simple aggregation. The "$avg" method itself was not difficult when combined with the "$todecimal" method, but the two additional stages needed just to present the data in the format that the SQL tool could accomplish in just a few lines of code filled the screen and took several seconds to produce appropriate previews. Overall, for this type of project that mostly focused on basic to intermediate queries and basic aggregations, the execution time and syntaxical requirements of MongoDB was excessive given the relatively small size of the document store at this point. The structured data is well-suited for a relational database, but the addition of unstructured data might make a document store more appropriate. The desire to join data and make timewise comparisons could be completed in Microsoft SQL Server Management Studio with much less difficulty. However, should the project be scaled to include many years or perhaps base information on populous cities rather than country data, the automatic sharding of MongoDB might make the processing time worth it for the eventual consistency and security.

Next Steps

If a data science team were to pursue deeper analysis of this dataset, perhaps the most interesting time in which to analyze happiness would be in the wake of the COVID-19 pandemic. In order to make predictions about how COVID-19 affected a country's happiness, I would need to seek new attributes to supplement the data that already exists in the 2020 dataset. Firstly, I would gather data about the country's government: the name of the nation's leader and the general type of government as either a monarchy or a democracy. This information might also show interesting trends about government oversight and general happiness, not only in 2020 but at any given time. Next, I would use a factor with levels of "full shutdown, partial shutdown, no change" as a variable representing the country's COVID-19 response. This factor would give an idea of how much control the government has and what type of response they were able to implement. In addition to data regarding the federal actions taken, I would then need to gather information about COVID-19 deaths in that country. However, in order to normalize the data, I would also gather information about the country's total population and generate a proportion of deaths to citizens. In addition to information about deaths, measuring lost GDP during the pandemic would also be worthwhile. Then, the overall questionnaire could include "COVID-19 Response" questions alongside the existing contributing factors to quantify to what extent the

pandemic impacted the country's happiness score. Note: each contributing factor is not listed on the entity relationship diagram to reduce visual clutter, but all contributing factors remain the same from the original dataset in addition to the COVID-19 Response factor.

In addition to this primary COVID-19 guiding question, I would like to test my hypothesis regarding the regional aspect of the residual. Firstly, I would alter the Dystopian Residual variable within the country entities to instead represent a proportion of the overall happiness score. My hypothesis is that while the current residual variable doesn't show that much of a score difference across countries; however, when comparing how much of the happiness score is composed of the residual, there is a regional disparity in which a greater proportion of the happiness score of African is unable to be quantified with the existing contributing factors (Strait). Therefore, I would create a regional entity and compute a regional residual proportion to portray this information alongside country proportional data. As a data science team, these findings could be reported to the United Nations Sustainable Development Solutions Network to encourage diverse and responsible data gathering in majority world countries to truly understand the factors influencing their happiness, and to recognize that these factors may differ from the Western world. The entity relationship diagram reflecting my changes is in Appendix C.

Conclusion

The World Happiness Index analyzes the well-being of citizens all around the world in the form of relevant contributing factors that could be manipulated with a variety of tools, including Microsoft SQL Server Management Studio and NoSQL program MongoDB Atlas. While each tool came with advantages and disadvantages, for the current structured state of the data, an SQL program allows for easy querying and analysis of the dataset. Developing a schema was somewhat challenging in the Microsoft SQL Server Management Studio, but worthwhile to ensure consistency across data types. However, this tool did not offer the same automatic security features via sharding that MongoDB Atlas boasts. When running queries, Microsoft SQL Server Management Studio allowed for easy and time-efficient aggregation. Join operations to analyze multiple years of data could be done without difficulty, managing data types is doable with minimal data cleaning, and querying across multiple variables was welcomed by the interface. MongoDB Atlas does not allow for easy joins to relate different tables, has a complex aggregation pipeline that slows data manipulation, and can only compare in efficiency with straightforward projection queries. However, if the dataset were to be expanded to include unstructured data or information from many years, a NoSQL tool might be more appropriate to manage an extensive and complex dataset. If next steps were taken, including government and COVID-19 information might make a document store better-suited for national details. Computing proportions could be done with an SQL tool sans aggregation pipeline. In general,

both tools could complete the benchmarks outlined in this study, but Microsoft SQL Server Management Studio allowed structured data manipulation in a simple and user-friendly form.

References

"Aggregation Pipeline Builder" *Aggregation Pipeline Builder - MongoDB Manual,* docs.mongodb.com/compass/master/aggregation-pipeline-builder/

Cramer, Maria. "Smile? The Results From the 2020 World Happiness Report Are In." *The New York Times*, The New York Times, 20 Mar. 2020, www.nytimes.com/2020/03/20/world/europe/world-happiness-report.html.

Drake, Mark. "Understanding Database Sharding." *DigitalOcean*, DigitalOcean, LLC, 7 Feb. 2019, www.digitalocean.com/community/tutorials/understanding-database-sharding.

Jayaram, Prashanth. "When to Use (and Not to Use) MongoDB - DZone Database." *Database Zone*, Devada Media, 30 Nov. 2016, dzone.com/articles/why-mongodb.

"$Lookup (Aggregation)" *$Lookup (Aggregation) - MongoDB Manual,* docs.mongodb.com/manual/reference/operator/aggregation/lookup/.

Strait, Jessica. "Final Project: World Happiness Index." (2020) GitHub Repository. https://github.com/jesslynne73/Intro-to-R/blob/master/Final-Project/FinalProject.Rmd

Sustainable Development Solutions Network. "World Happiness Report." *Kaggle*, 27 Nov. 2019, www.kaggle.com/unsdsn/world-happiness.

"World Happiness Report." *United Nations Sustainable Development Solutions Network*, 20 Mar. 2020, worldhappiness.report/.

Appendix A: Microsoft SQL Server Management Studio

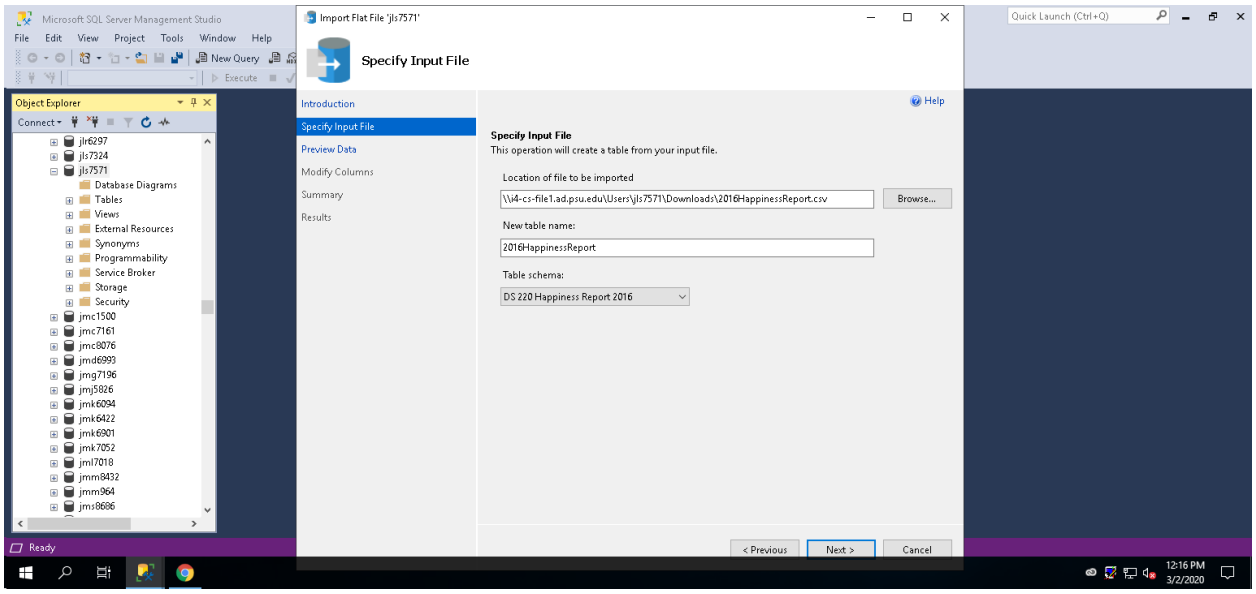

Figure 1A: Creating a schema and importing data as a flat file

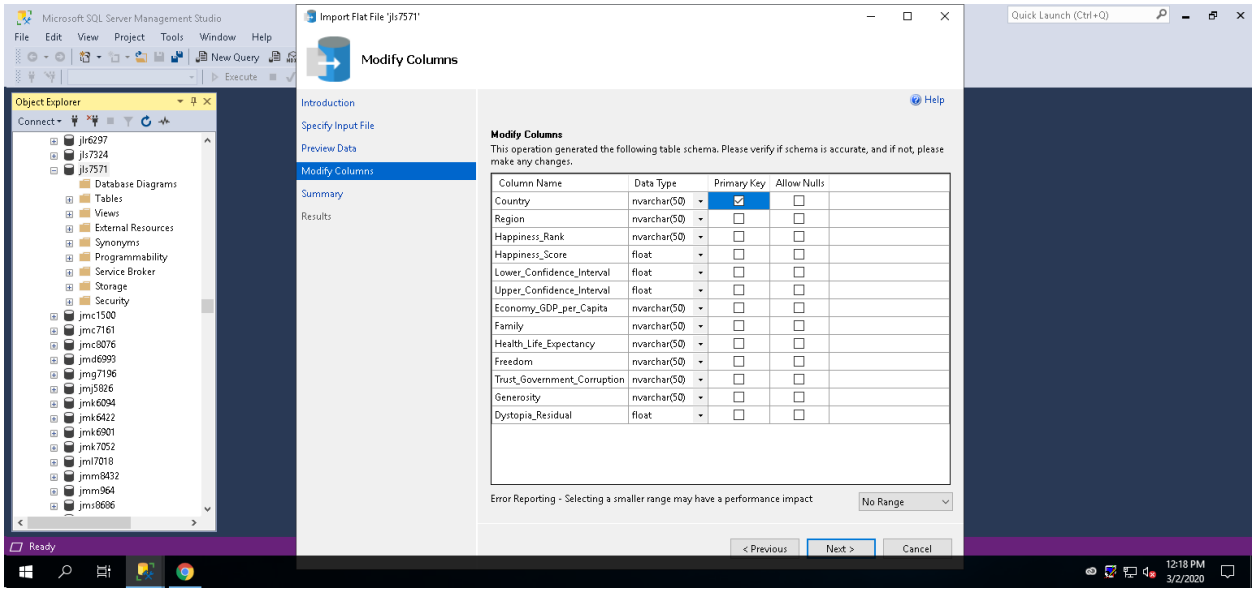

Figure 2A: Assigning a primary key

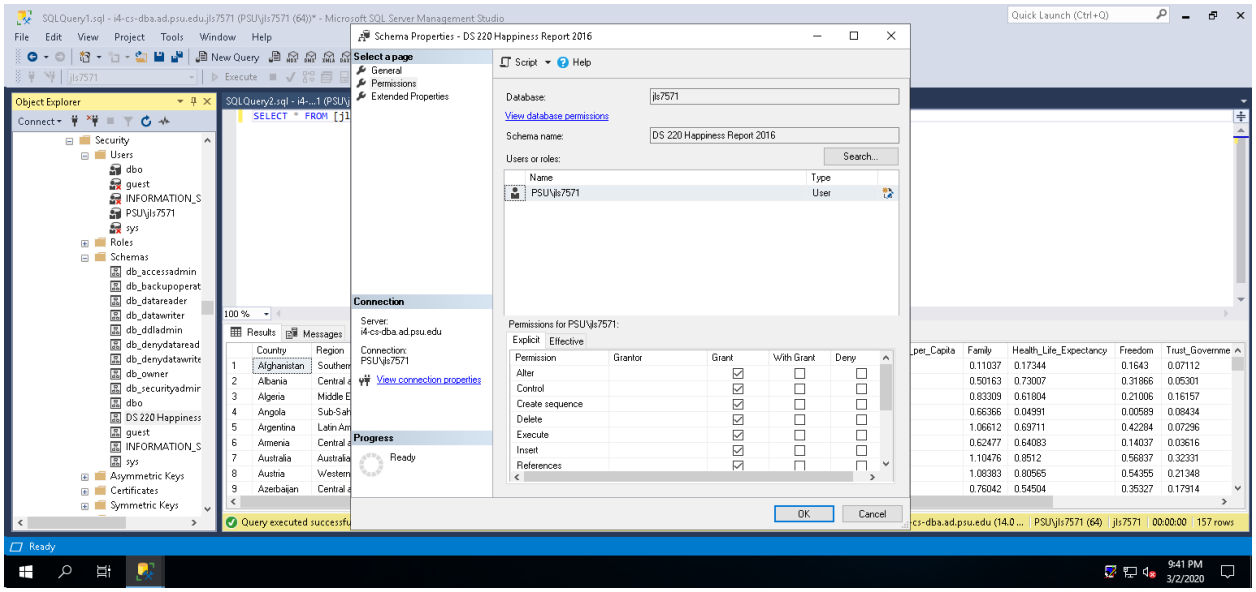

Figure 3A: Granting account permissions within schema security

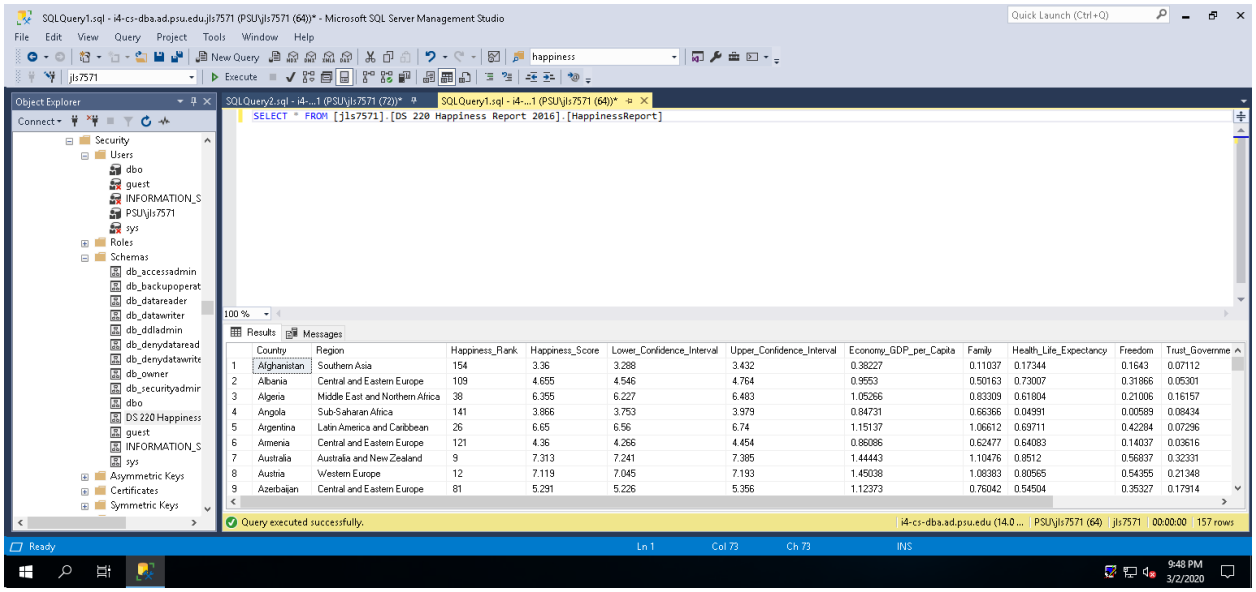

Figure 4A: Sample of imported datasets in the Microsoft SQL Server Management Studio

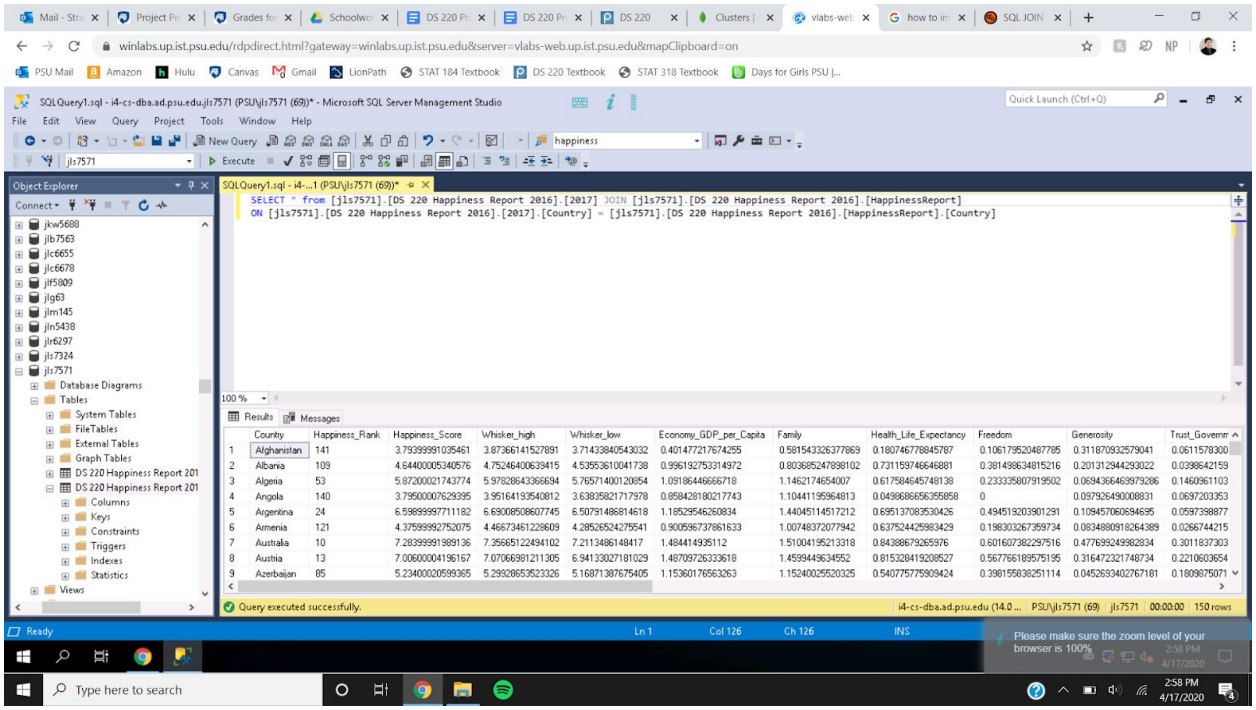

Figure 5A: Joining 2016 and 2017 datasets in Microsoft SQL Server Management Studio

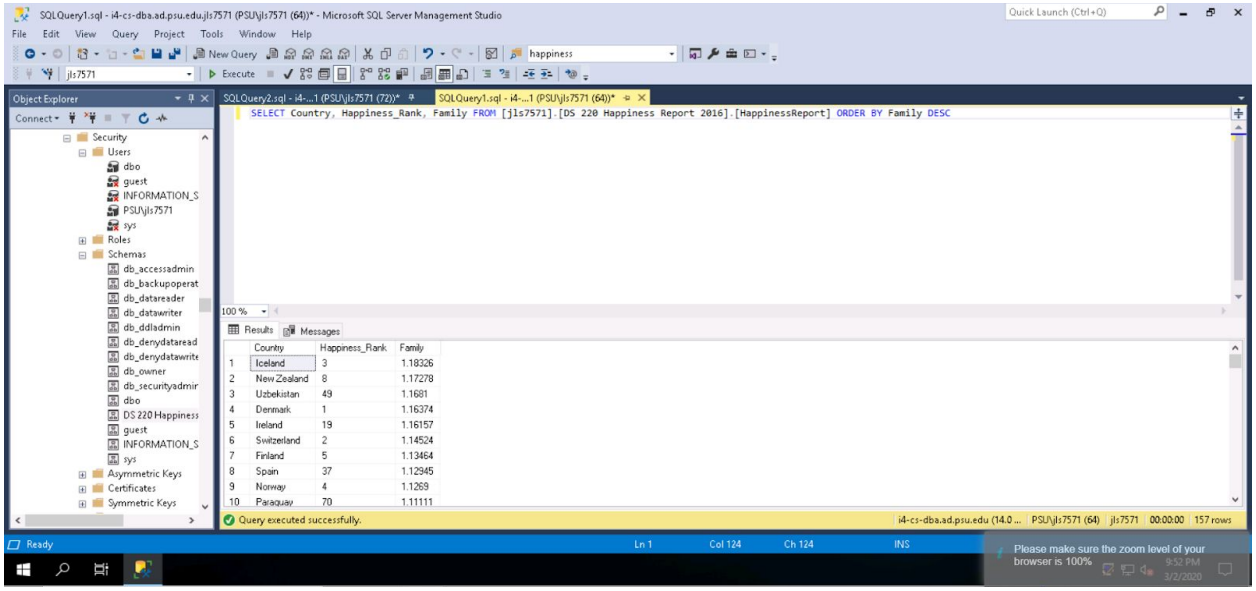

Figure 6A: Presenting data according to the “Family” contributing factor

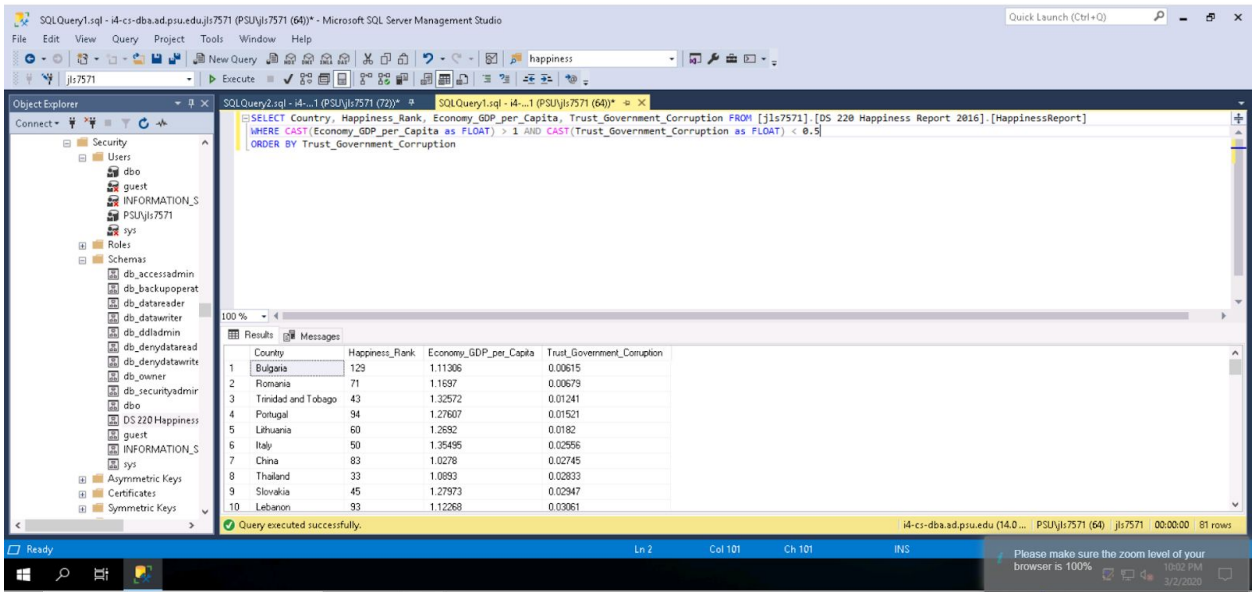

Figure 7A: Identifying countries with a high GDP per capita and low trust in government

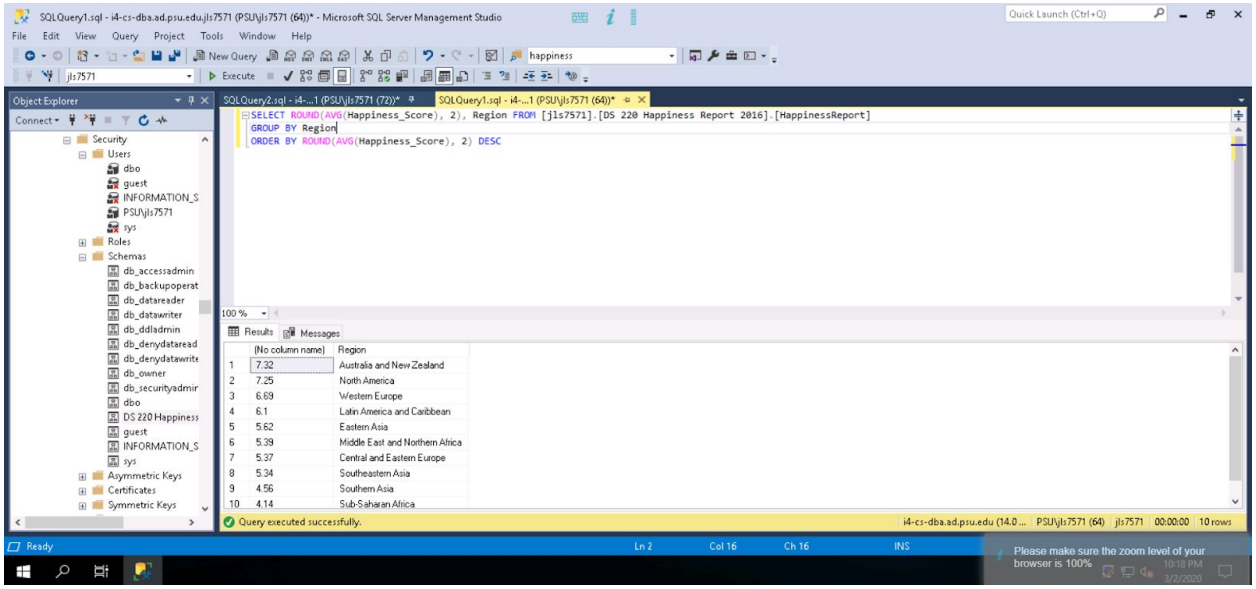

Figure 8A: Aggregating average happiness score by region

Appendix B: MongoDB Atlas

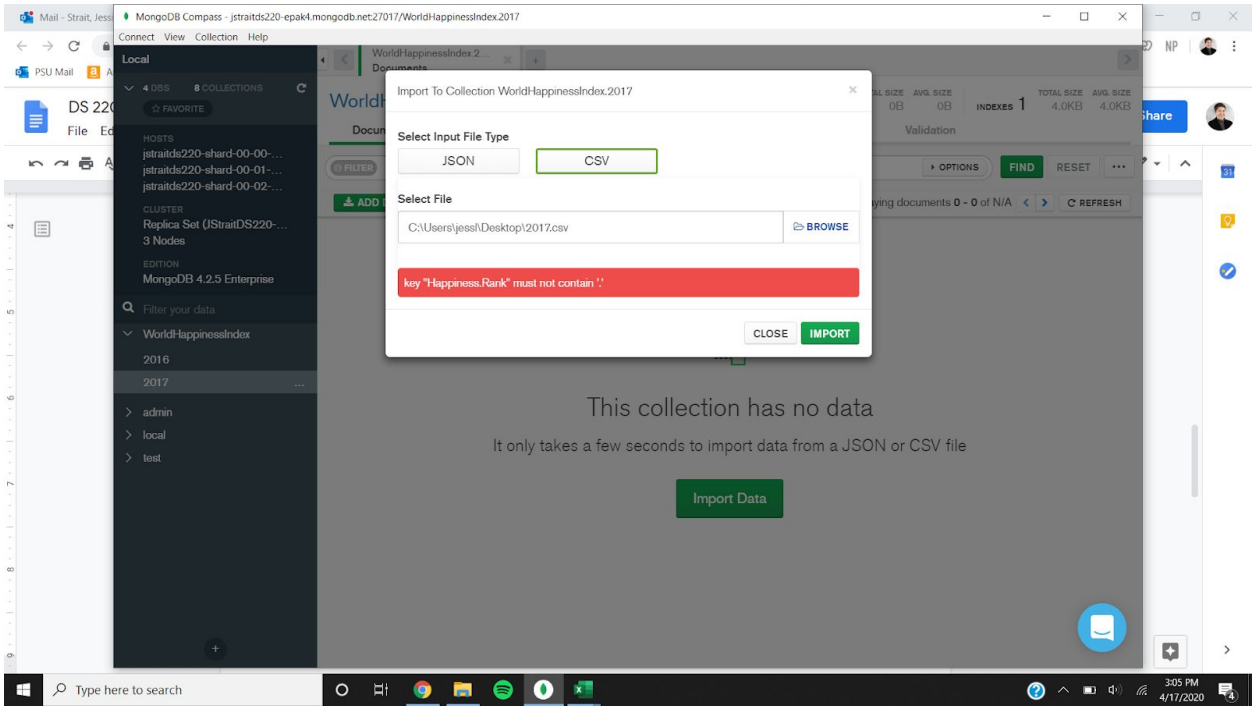

Figure 1B: Error message mandating data cleaning on the 2017 dataset variable punctuation

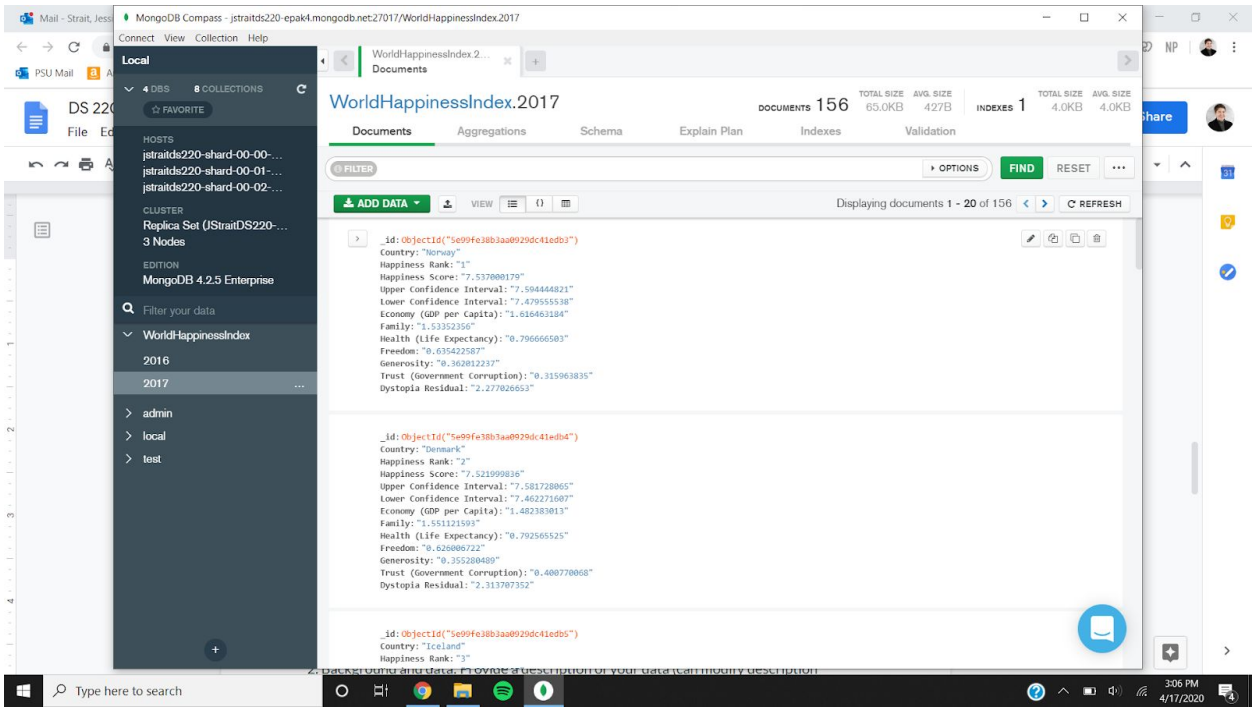

Figure 2B: Sample of imported datasets in the MongoDB Compass interface

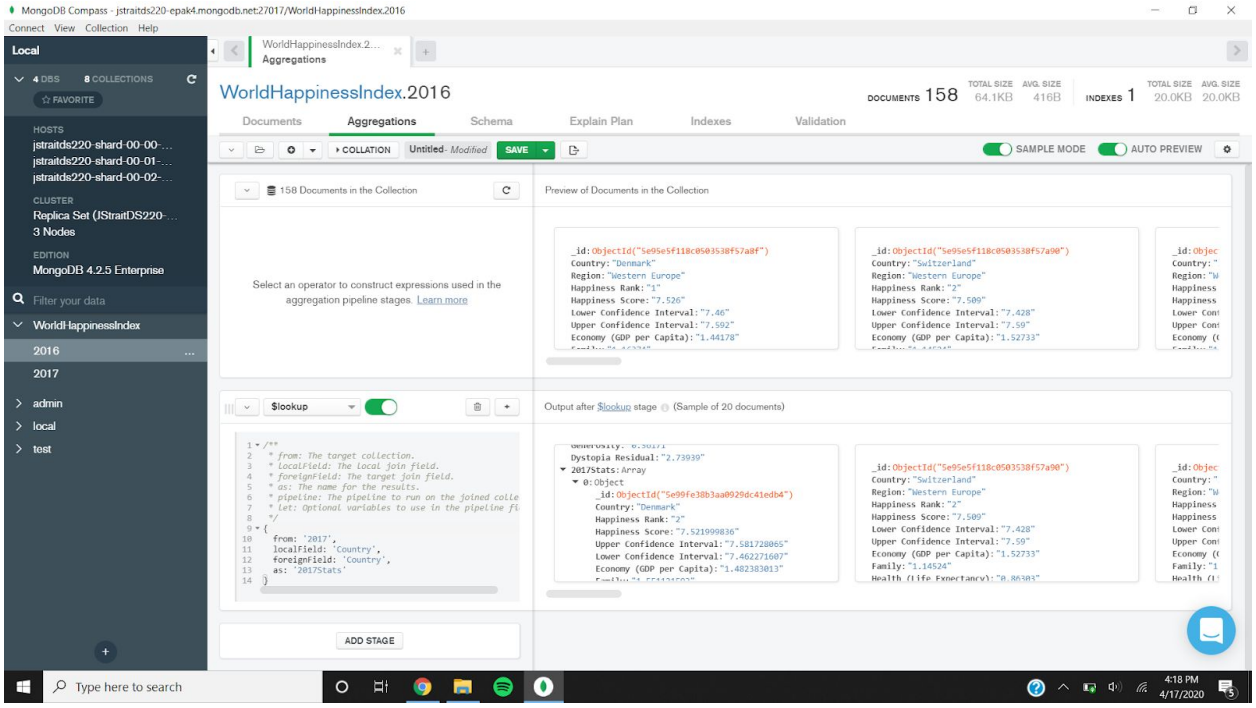

Figure 3B: Joining 2016 and 2017 datasets in MongoDB Compass

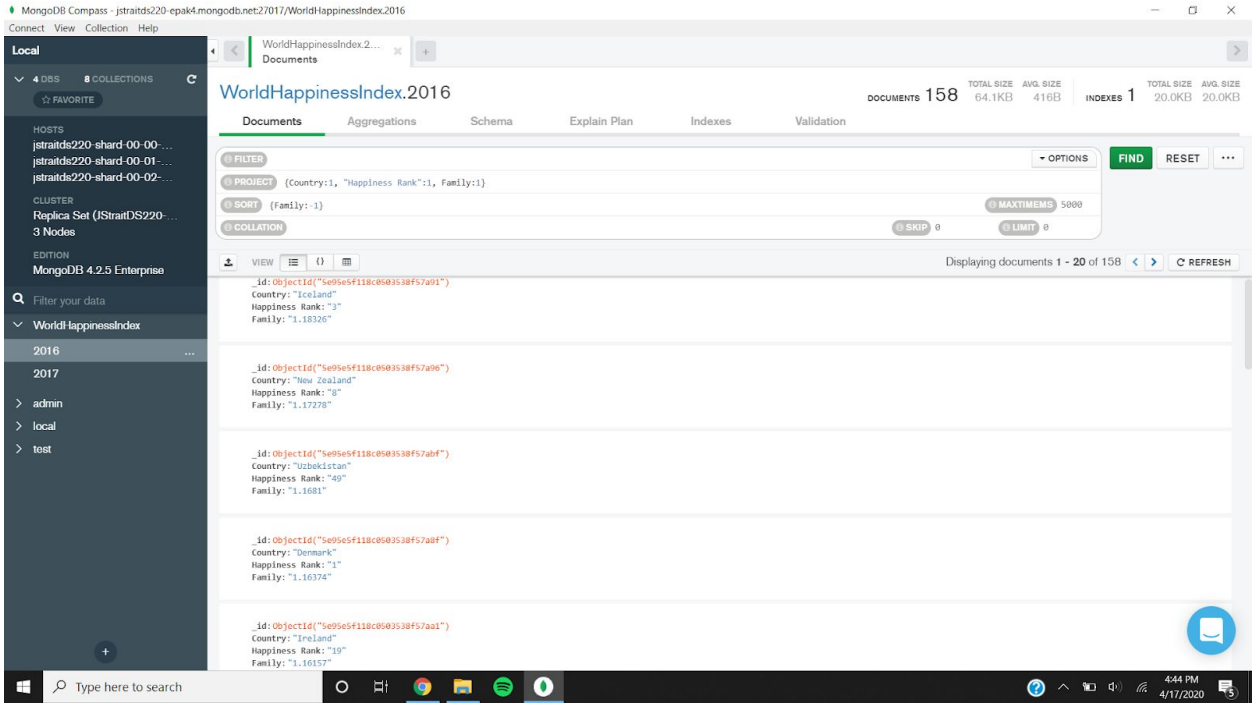

Figure 4B: Presenting data according to the “Family” contributing factor

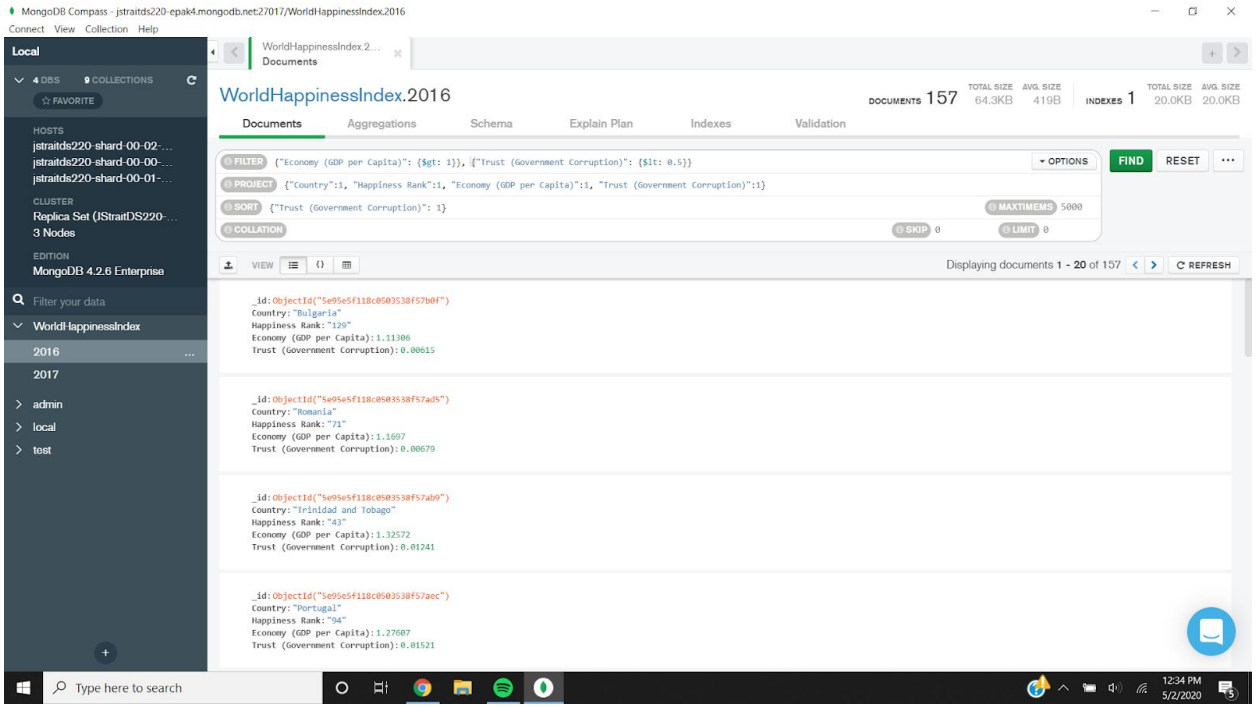

Figure 5B: Identifying countries with a high GDP per capita and low trust in government

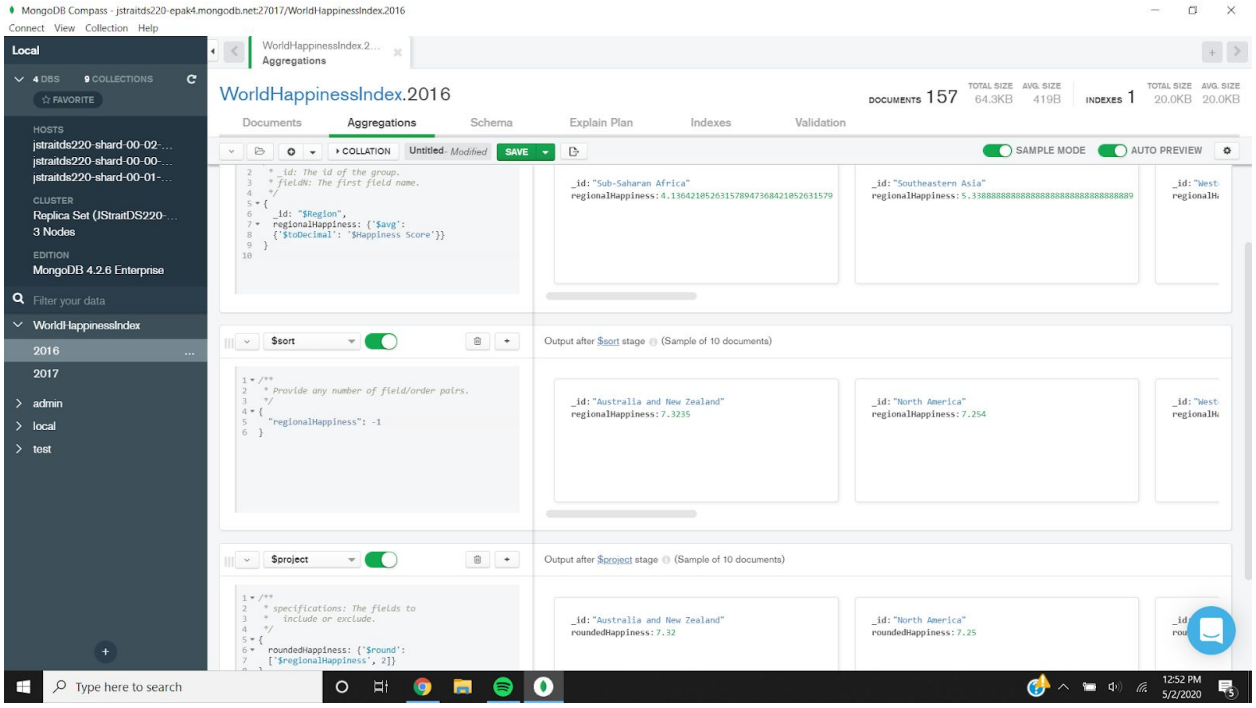

Figure 6B: Aggregating average happiness score by region

Appendix C: Entity-Relationship Diagram