Cloud systems

Student Name

institution

Table of Contents

[**1.** **INTRODUCTION** 3](#_Toc41387321)

[**System Architecture** 3](#_Toc41387322)

[**System Hardware Requirements** 3](#_Toc41387323)

[**Random Access Memory (RAM) and CPU** 3](#_Toc41387324)

[**Storage** 4](#_Toc41387325)

[**Azure PowerShell** 4](#_Toc41387326)

[**Testing** 4](#_Toc41387327)

[**2.** **DATASET IN THE CLOUD** 5](#_Toc41387328)

[**Type** 5](#_Toc41387329)

[**Exporting Data** 5](#_Toc41387330)

[**3.** **DATA PROCESSING/MANIPULATION** 8](#_Toc41387331)

[**World Countries Population in 2018** 8](#_Toc41387332)

[**Population Growth Between 1960 And 2018** 8](#_Toc41387333)

[**Correctness** 9](#_Toc41387334)

[**Localhost** 9](#_Toc41387335)

[**Azure** 9](#_Toc41387336)

[**Scalability** 10](#_Toc41387337)

[**4.** **SCALABILITY** 11](#_Toc41387338)

[**Software and Query Optimization** 11](#_Toc41387339)

[**CONCLUSION** 13](#_Toc41387340)

[REFERENCES 14](#_Toc41387341)

# **INTRODUCTION**

Cloud computing is fast changing the landscape of data science by providing numerous solutions and tools for machine learning. Through available open source datasets in the internet, training algorithms especially in artificial intelligence has been made much easier. We are now able to acquire massive datasets and affordable computing power through cloud computing platforms to test, develop and deploy solutions with security, reliability and scalability.

In this paper, I’ll be providing proof of concept (PoC) in big data processing by uploading a dataset, manipulating/processing it and documenting results of important insights from the dataset using Microsoft Azure computing platform.

## **System Architecture**

The Azure cloud servers will be accessed remotely thus we’ll be working with the Client Server system architecture. The Azure cloud servers are demonstrated in the diagram:

Online Transaction Processing(OLTP)

Analysis

Online Analytical Processing(OLAP)

Reports

SQL

SQL DATABASE

ETL – (EXTRACT, TRANSFORM, LOAD)

Using relational database management system (RDBMS) hosted on Azure, the dataset workload will be processed using the online transaction processing (OLTP) and analyzed by the online analytical processing (OLAP) components on the relational database management system architecture on Azure [6].

Relational data is defined using a predefined schema or rather structure and a set of unique keys o constraints to maintain data referential integrity. Structured Query Language (SQL)

## **System Hardware Requirements**

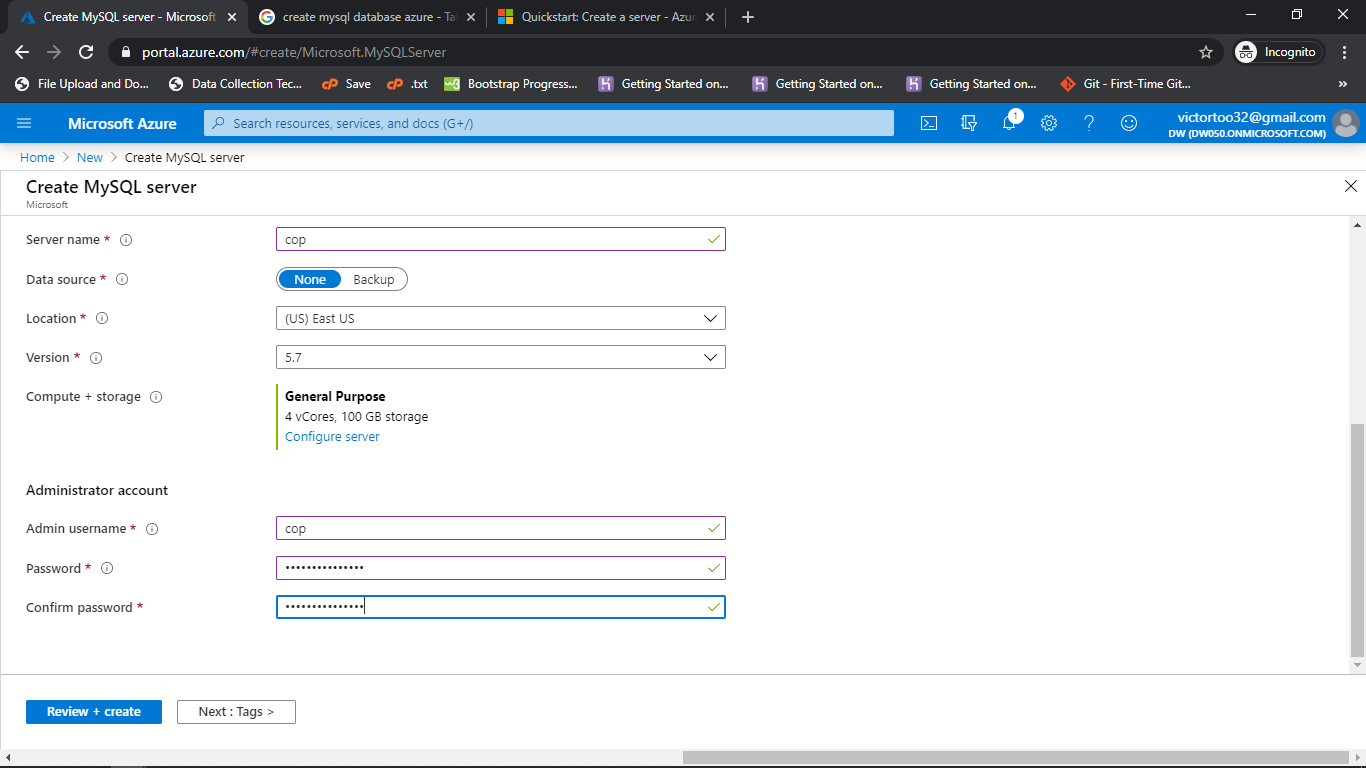
The Proof of Concept (PoC) in this case will be done under basic Azure hardware specifications as specified below:

### **Random Access Memory (RAM) and CPU**

* Minimum of 4GB (Gigabytes) to handle computing.
* 4 virtual cores(vCPUs) will adequately support our concept testing optimally by handling the computing [1].

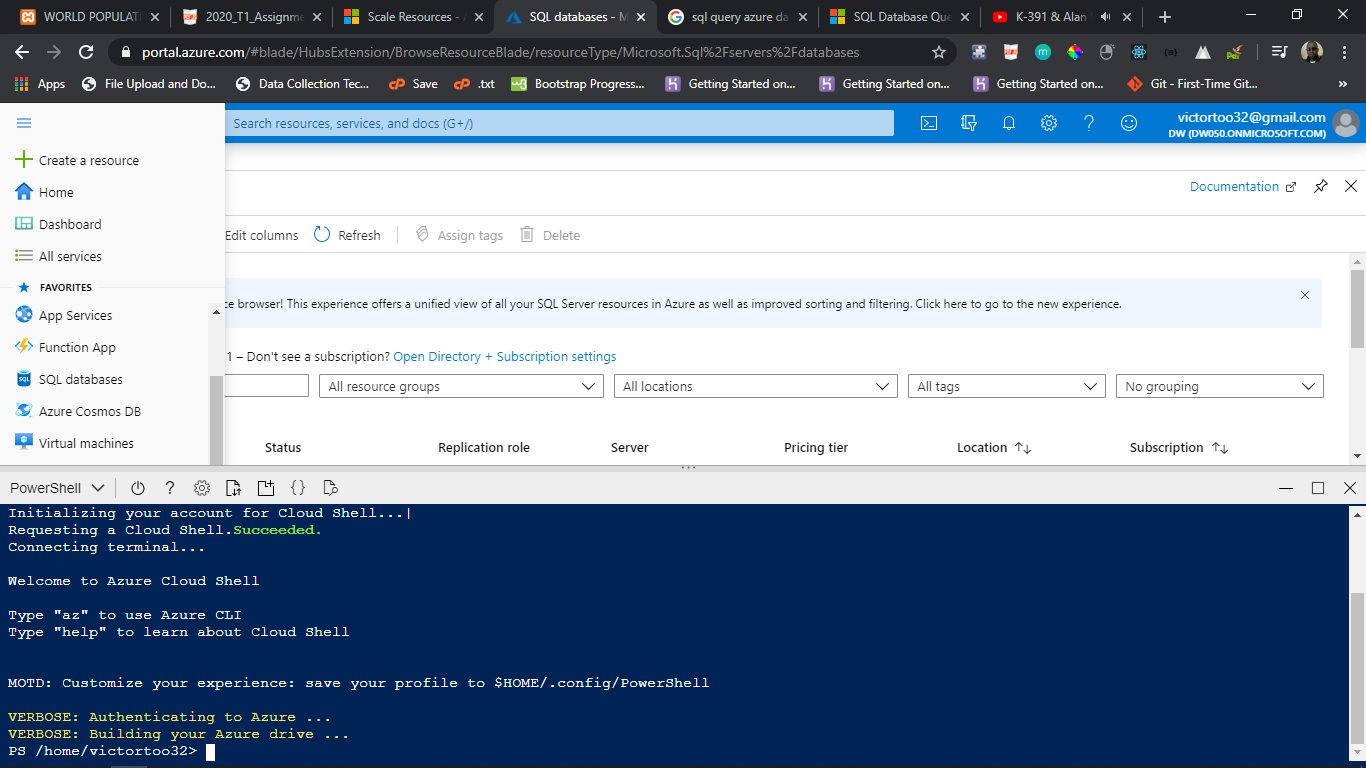
### **Storage**

* Minimum of 100GB (Gigabytes) of storage space. Adequate storage is necessary for handling big data processing and storage in cloud computing. Its worth noting our dataset will not be big to this scale but the idea of storage is to proof the concept under testing [1].



### **Azure PowerShell**

* The azure PowerShell comes a pool of azure stack tools with command support. The tools can help in data manipulation and other tasks through commands run in the PowerShell [5].



## **Testing**

This proof of concept will be tested using word bank total population public dataset which provides statistics on total population in every country in the world since 1960 to 2018. We will use csv data format of the population and further process the data to draw results such as the countries with highest population, the population of each gender and other metrics to support the concept of data processing or manipulation [4].

The csv data will be exported to Azure cloud hosted SQL database. The data will be queried and manipulated on the cloud to provide results in demonstration of how big data can be processed in the cloud thus proofing cloud computing techniques concept required in this paper [4].

The link to the data used is referenced in the reference section at the end of this paper.

# **DATASET IN THE CLOUD**

This section focuses on getting the downloaded data to a database in Microsoft Azure Cloud computing platform.

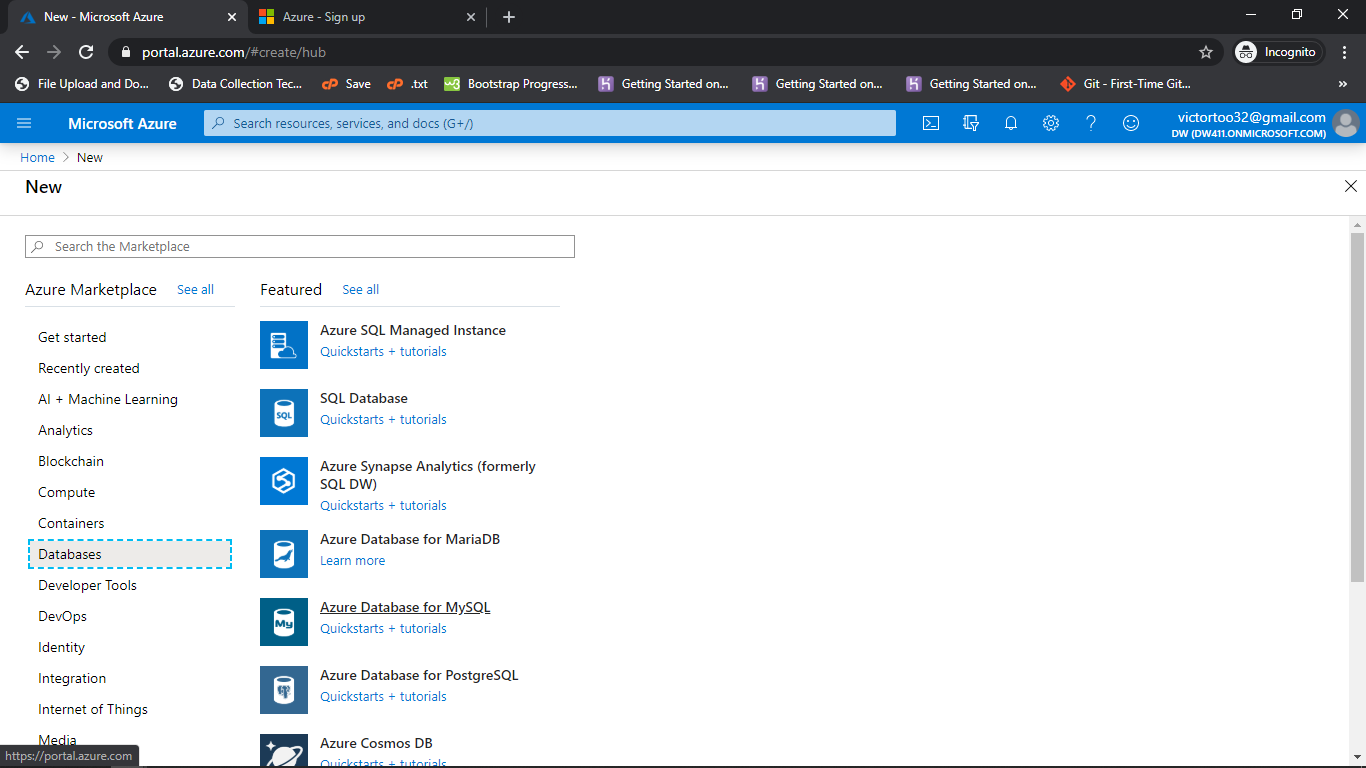
## **Type**

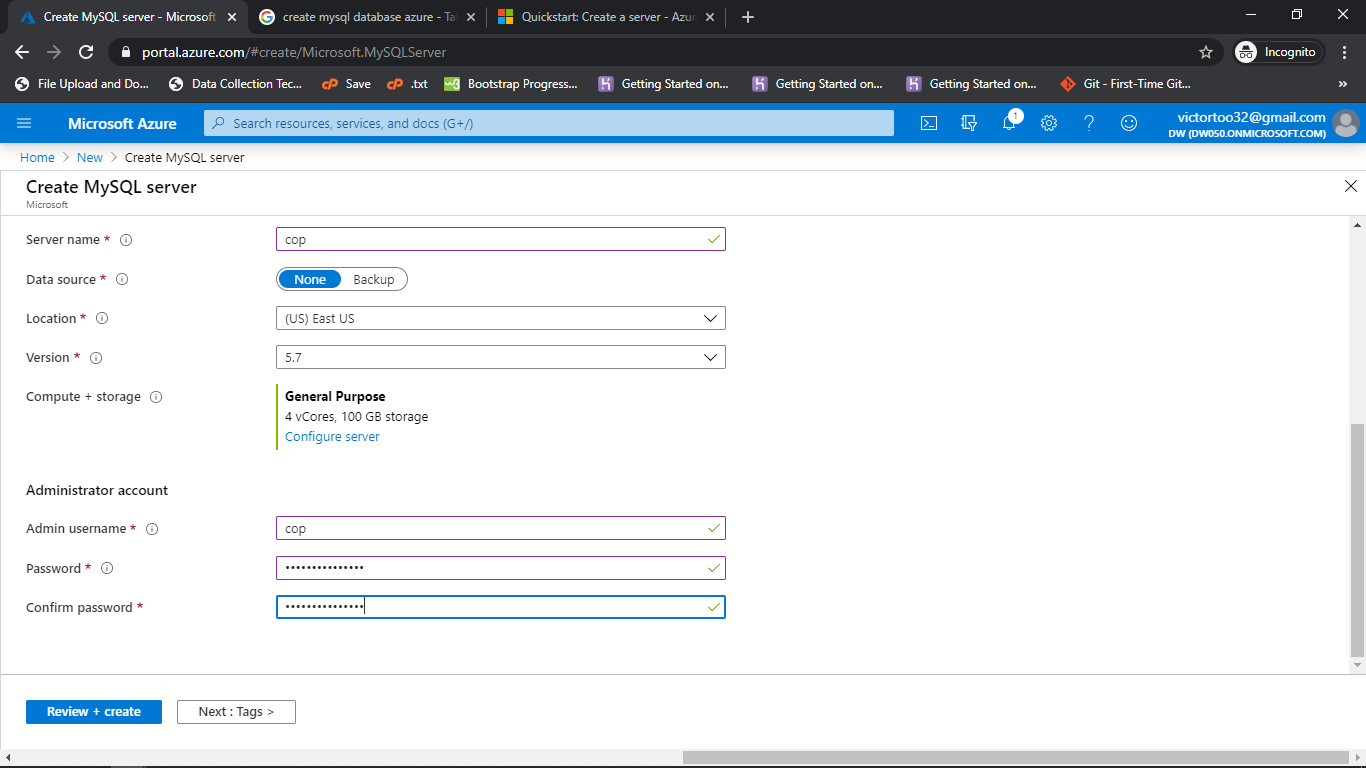
The downloaded data is in CSV format, this is easy to export to an Azure database as it retains rows and columns ideal for a relational database management system (RDBMS) [2].

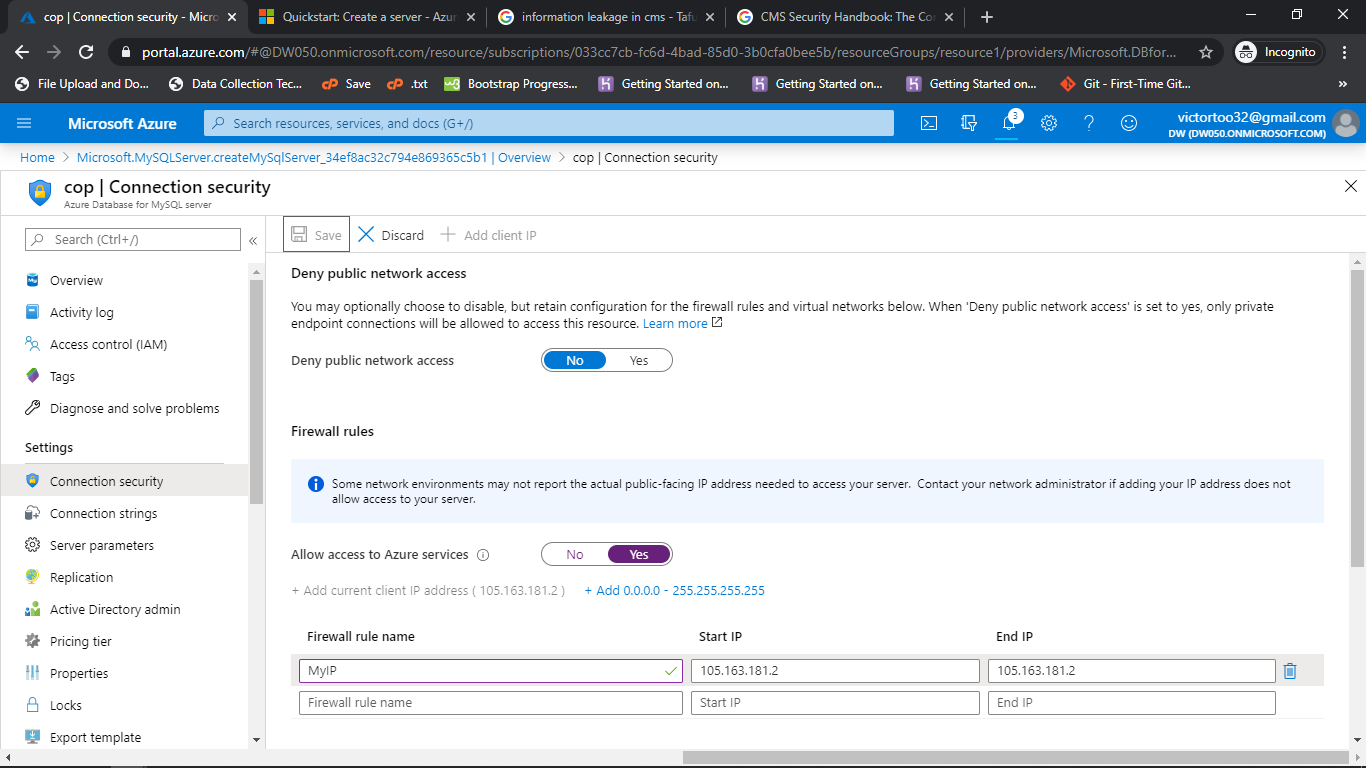
## **Exporting Data**

1. Once logged in to Microsoft Azure account portal, create an Azure database for MySQL server. The database will contain the respective table for the population data. This is illustrated in the screenshots below:

* Click on create resource, this takes you to a new page with different resources, click on databases and select Azure database for MySQL, proceed and provide details of the server and username as well as password. After Successful creation, configure firewall to allow external connection [2].

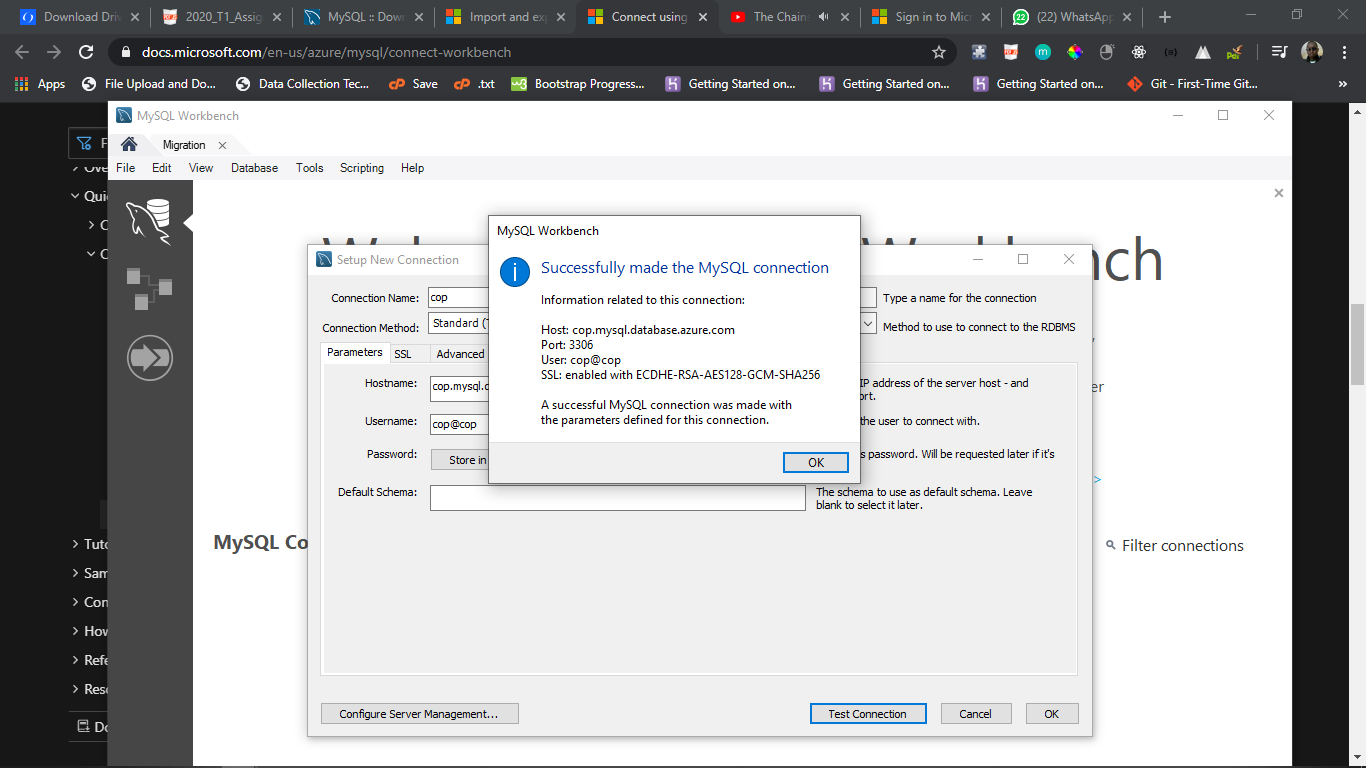




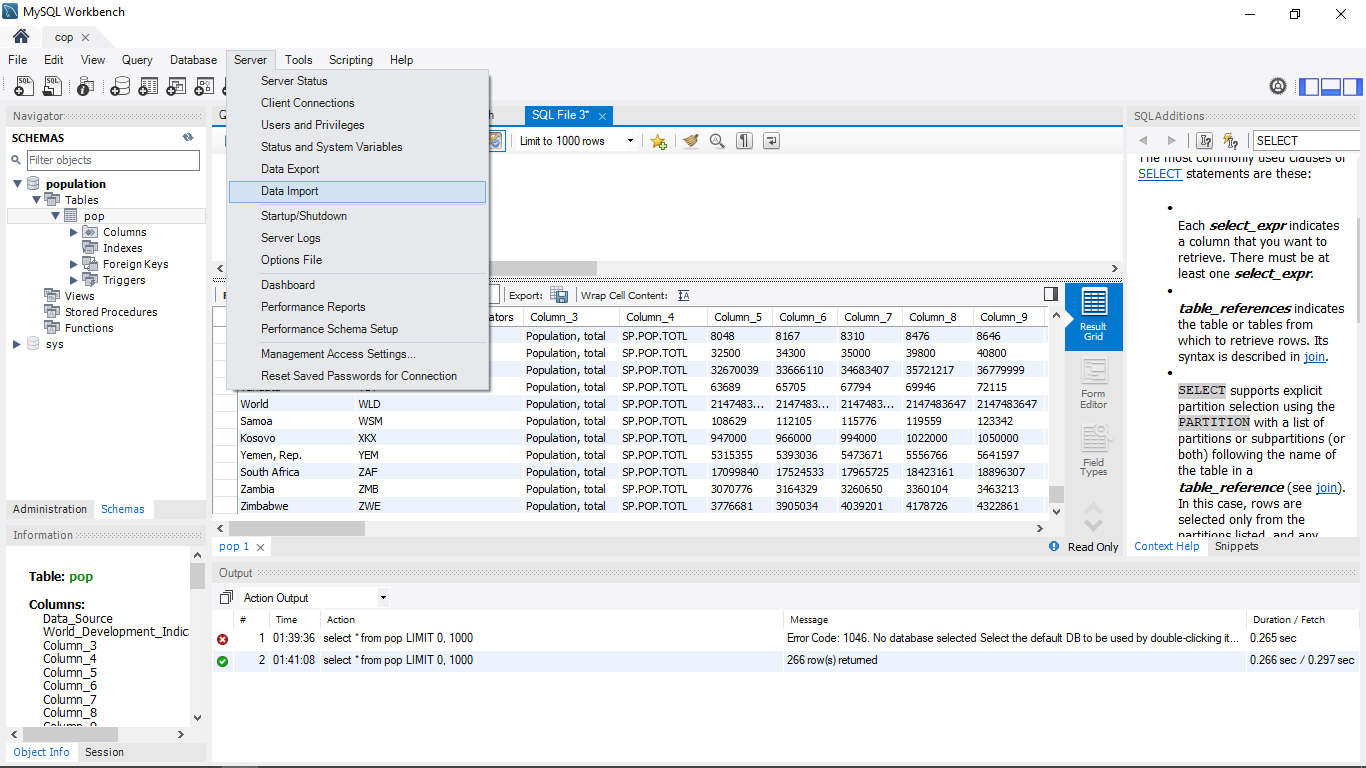


1. Using MySQL work bench, a third-party application to connect to Azure MySQL database, the data will be exported to the cloud. See the following screenshots [3].

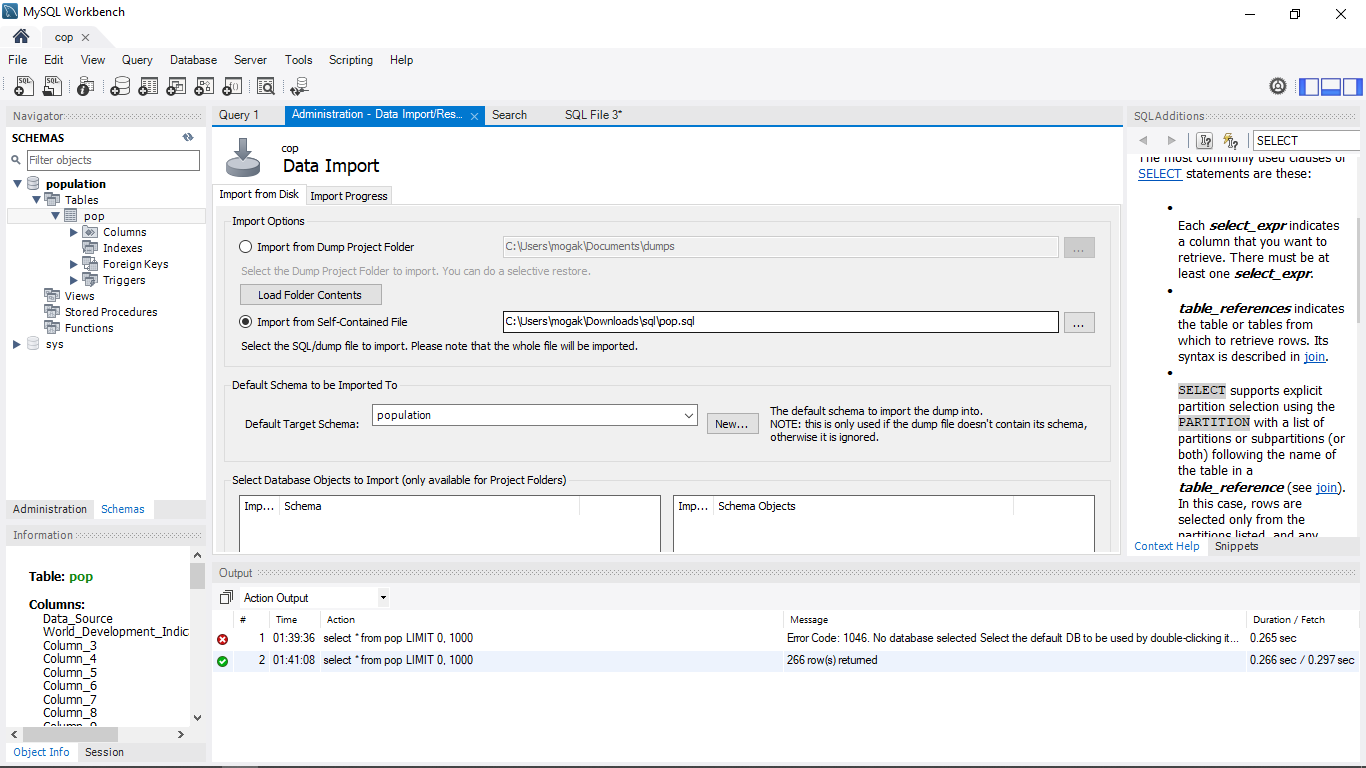
* Provide database credentials to connect to database



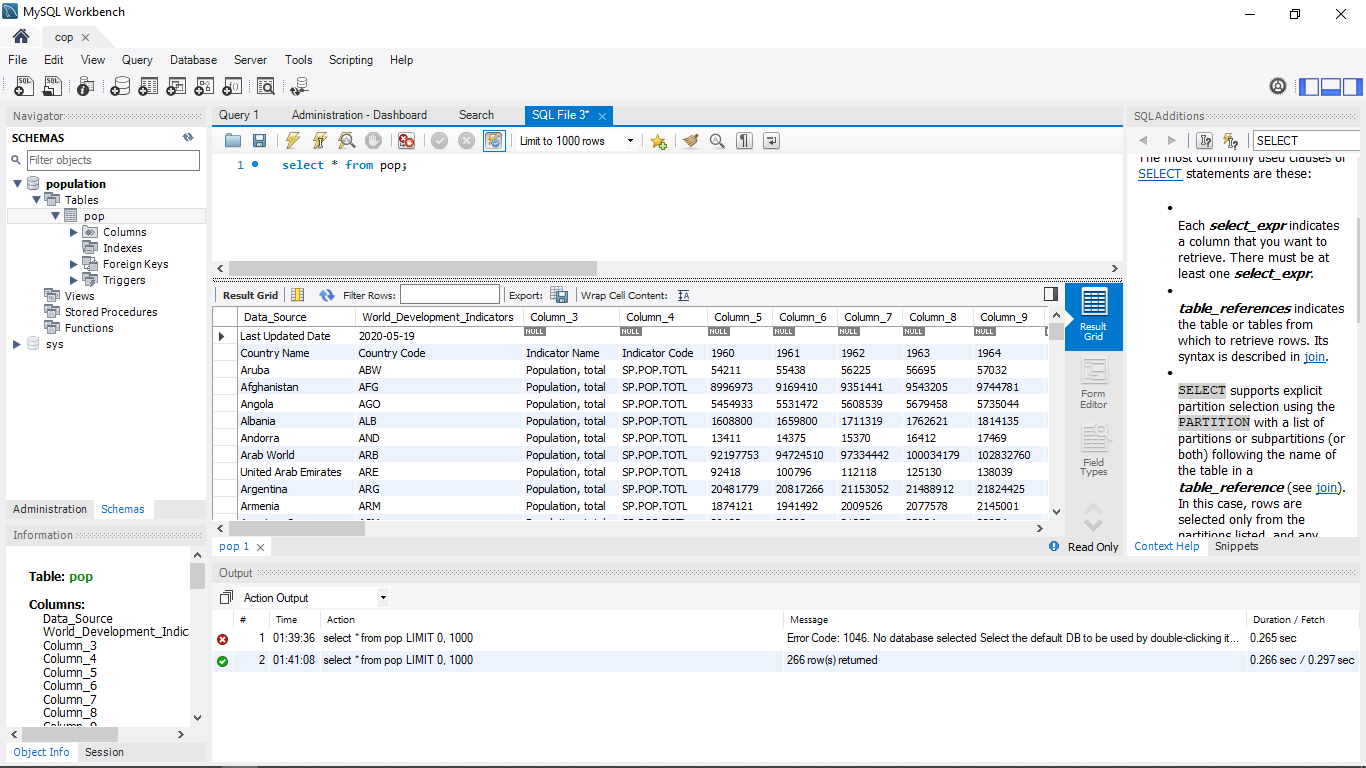
* Once connected, select schemas from the left sidebar and double click on the table to import to. On the top navigation under server, select data import to import the data into the database [3]. See the screenshots below:



Select your dataset from your local directory and click on start Import



The data is imported successfully. Incase of any errors, they are thrown at the botom console.



# **DATA PROCESSING/MANIPULATION**

The population dataset was successfully exported to Microsoft Azure database for MySQL in the previous section. In this section, we will manipulate the data and rely information using a simple web-based application.

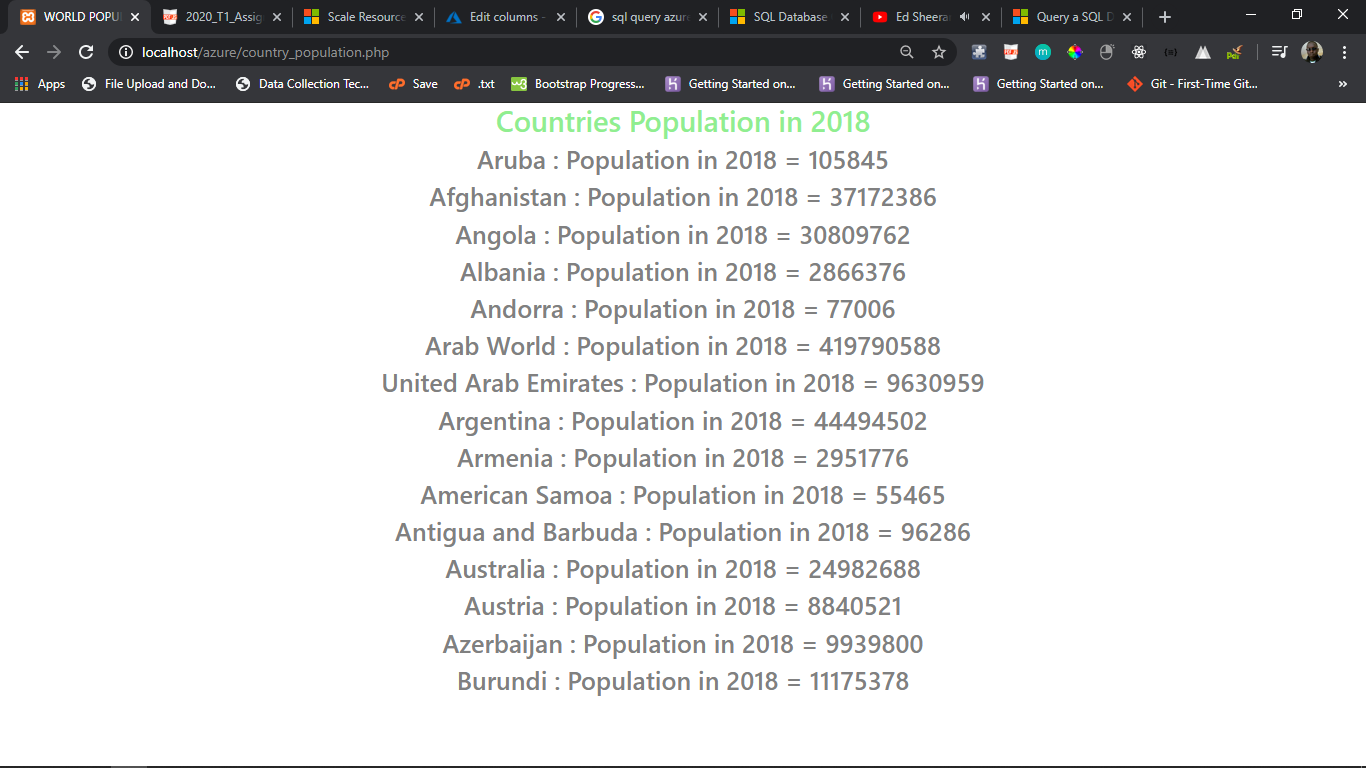
**Querying**

Using PHP programming language, we’ll establish a connection from our application code and query the database on Azure as well as do manipulation of the data [5].

For this Proof of Concept, we will query two aspects, the total population of the first 15 countries according to names/alphabets in the year 2018.

### **World Countries Population in 2018**

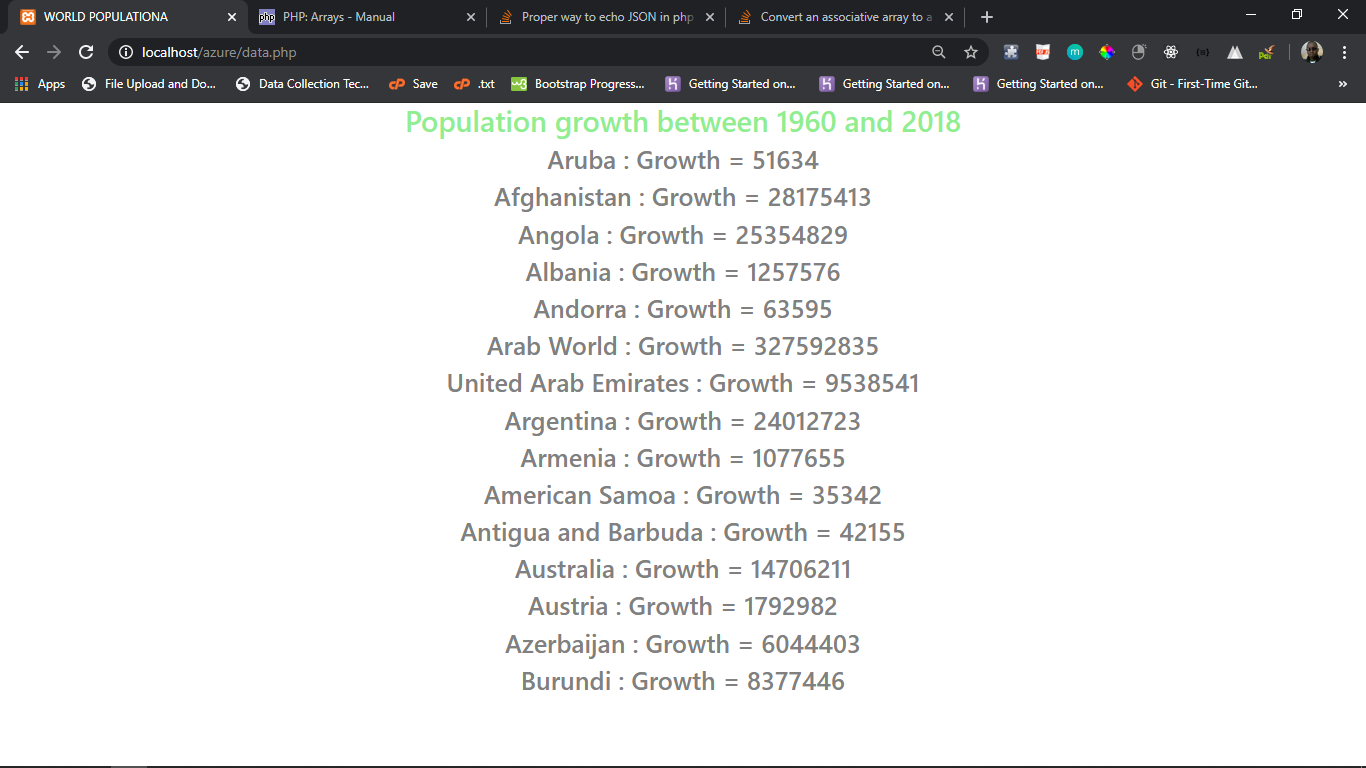
1. Establishing connection to PHP
2. provide database credentials for the database
3. Create connection using mysqli\_real\_connect. The connection will be used to query the database
4. Query results



### **Population Growth Between 1960 And 2018**

Here, the population growth is determined by querying the population of the first 15 countries in terms of alphabets and subtracting the population of the respective countries in 1960. The code returns the name of the country and population growth

Query results

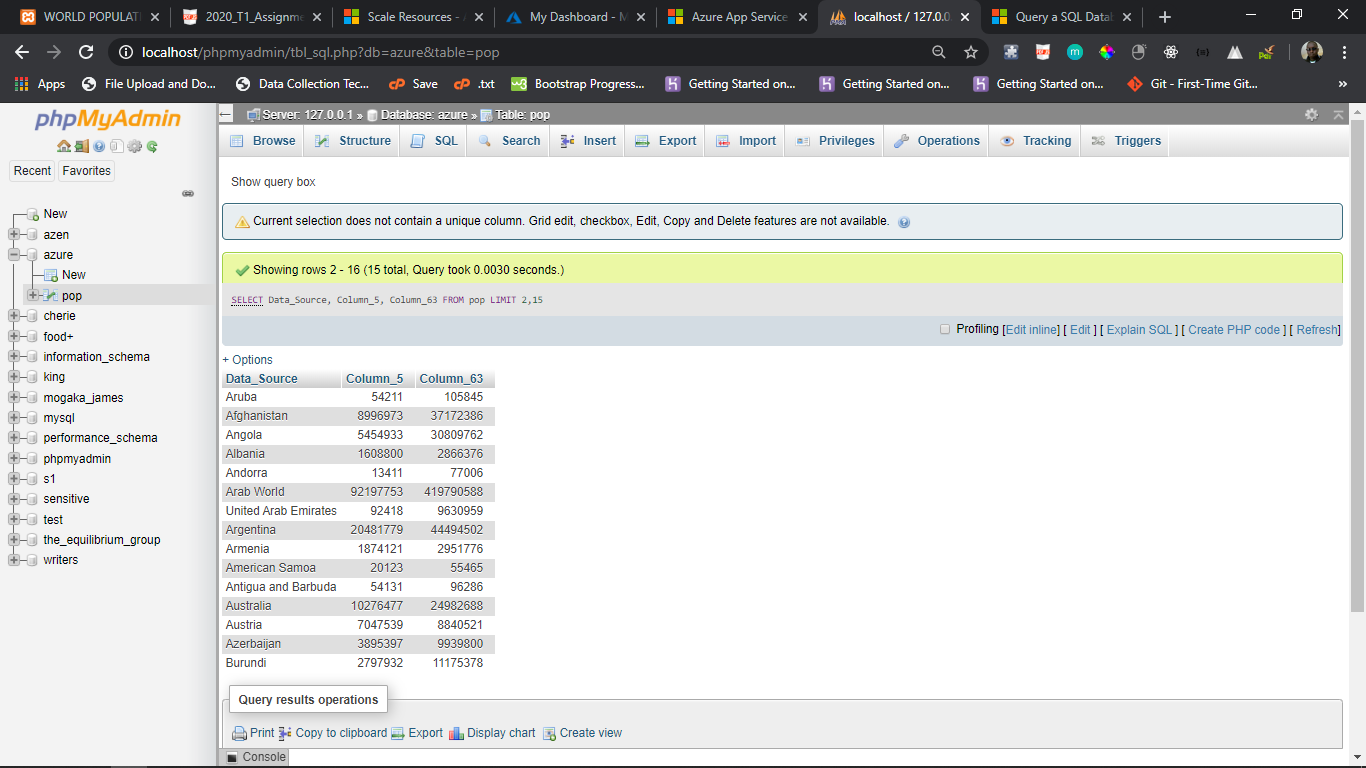


## **Correctness**

The population data is hereby queried on the local machine i.e. localhost and the results compared to the same query execution in the cloud on Azure. Taking the the population query in 3.1.2 above, well measure execution time of the query on local machine and on the cloud to retrieve the execution time or returning the country name, population in 1960 and population in 2018

### **Localhost**

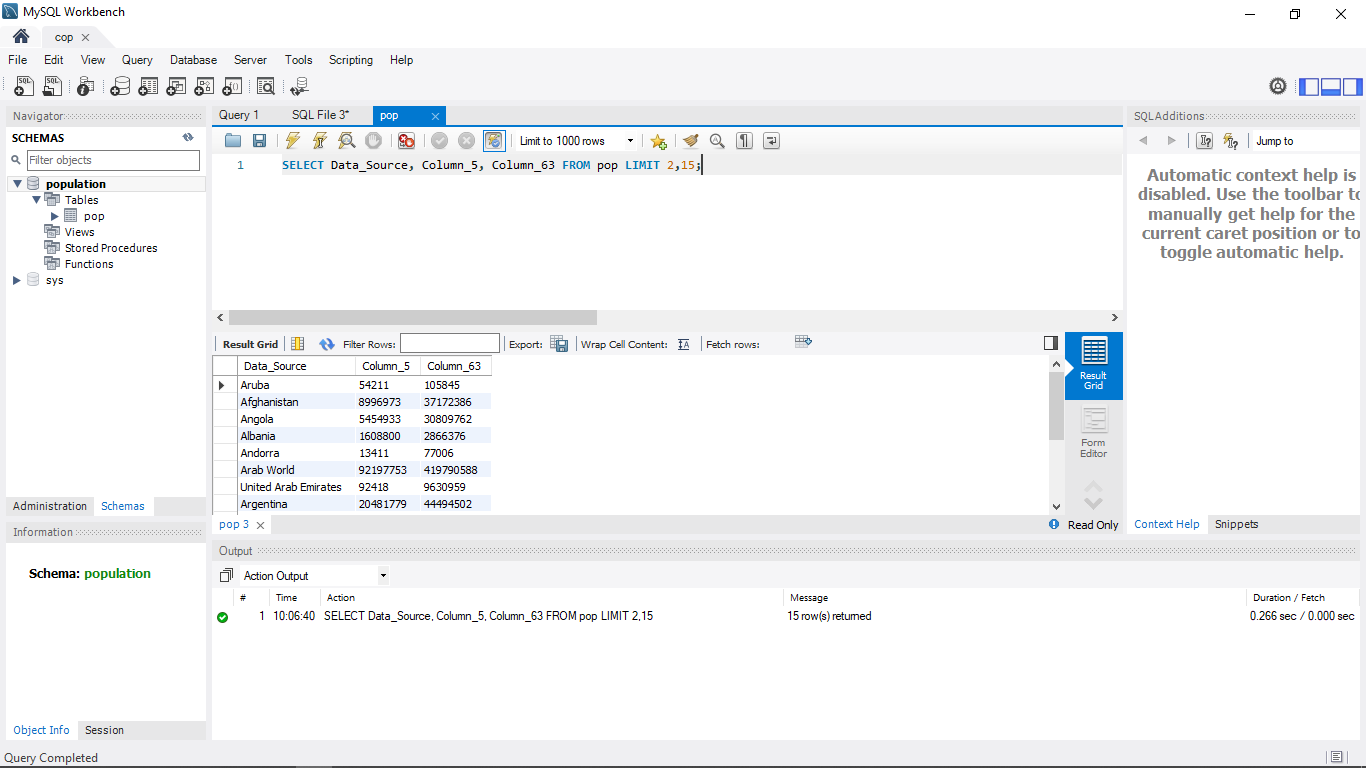
*Execution time = 0.0030 seconds (Illustrated in screenshot below)*



### **Azure**

*Execution time =0.000s (Illustrated in screenshot below, query through sql workbench)*

*The query fetched the data in 0.000s, 30x faster than localhost*



## **Scalability**

For an application hosted in Azure, scaling can be done on the fly rather automatically as the demand for resources increase which could be as a result of connection of millions of devices.

Single Azure SQL database only allows for manual scalability [6]. Scaling on Azure is based on a combination of CPUs, reads and writes.

# **SCALABILITY**

In a fixed resource system, the resources i.e. Storage, Computing or CPUs, Virtual machines cannot be increased. A cloud platform such as Azure provides for autoscaling where by the system scales Up and Down according to resource requirements from traffic by connected devices. When resources can not be increased to cater for traffic, the software developer must resolve to software and query optimization [6].

## **Software and Query Optimization**

Writing better code to avoid repeated calls for a function for instance helps lighten the code and enable its execution thus helping optimize system resource consumption during execution. Working with Multithreaded environments and languages such as python that have libraries to help in multithreading can help the system to support scaling traffic

In an SQL database like the one we have on Microsoft Azure, query optimization i.e. SQL query optimization can help response time and reduce system computing. Query optimization include:

1. Selecting required fields for each computing rather than selecting all (\*)
2. Avoid select distinct. Distinct selection works by grouping all data together before removing duplicated thus requires more computing
3. Create Joins using Inner Join

Using the population data on Microsoft Azure SQL data base:

*Total number of countries = 266*

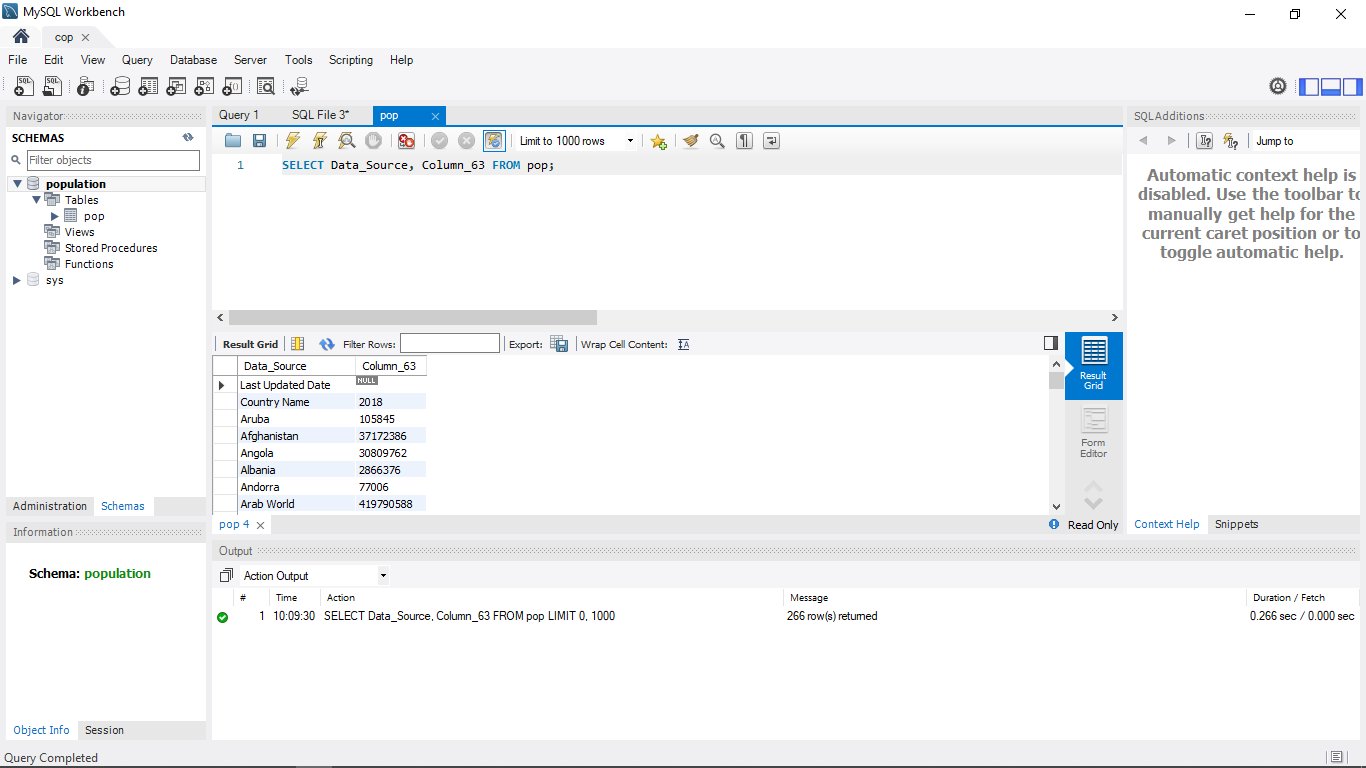
*Total number of columns = 64(with data in each year since 1960)*

*To compute a cartesian or cross join countries, using where, all possible variable of*

*(266 \* 64) are created before narrowing down the ones matching the where clause*

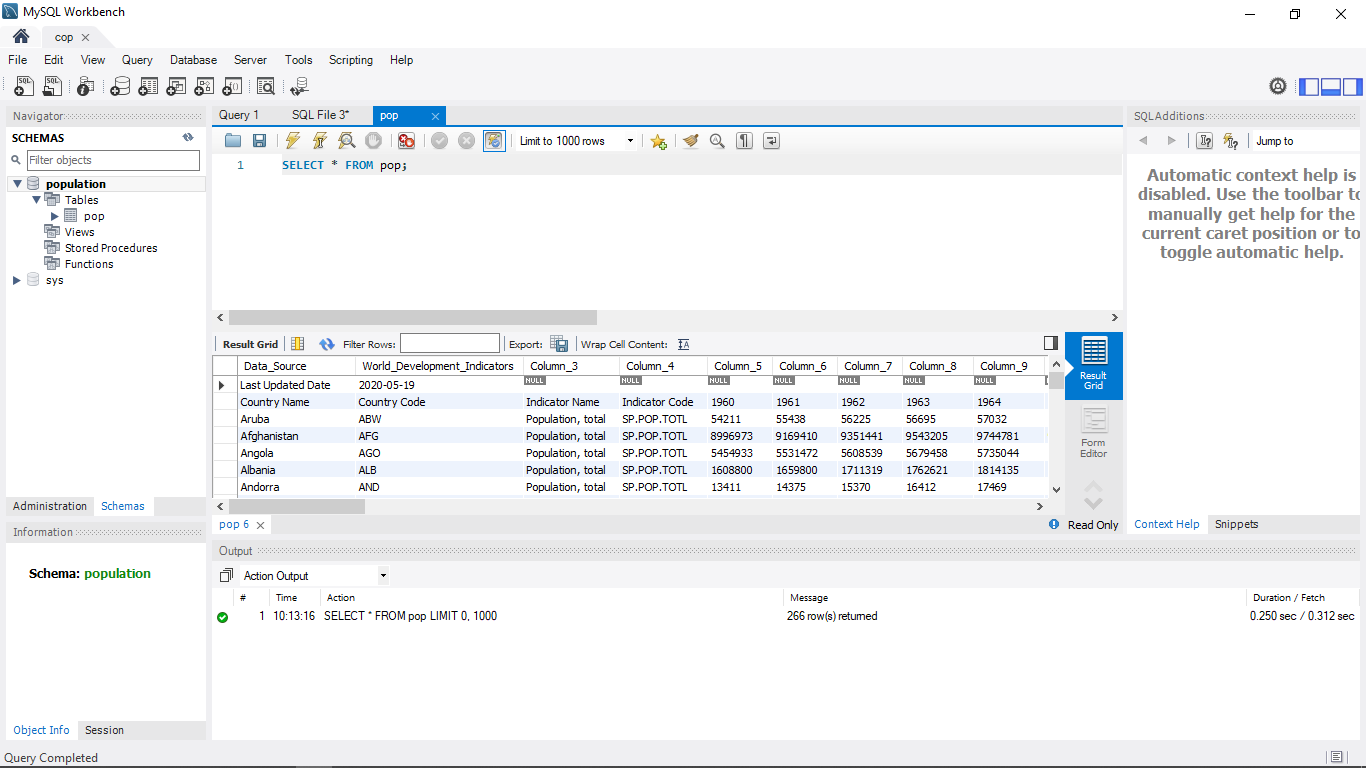
*This results in up to 100x extra and unnecessary work for a single query!*

*In a system with a million devices for instance, the query will be grossly suboptimal with probability of freezing and slow response.*



Querying all the data takes 0.266 and 0.000 seconds to return the whole information as shown above. Cumulatively it takes 0.266 seconds.

Selecting all on the other hand takes 0.25 and 0.312 seconds to return the data. Cumulatively it takes 0.562 to return the data. This is 2x slower and would have significant bottlenecks on fetching data when scaled to millions of devices.



# **CONCLUSION**

This paper was to provide Proof of Concept (PoC) of cloud computing. Using Microsoft Azure, a cloud computing platform, world population data provided by world bank was downloaded and exported to an Azure SQL database in the cloud and using Python, the data was queried successfully returning population of the first 14 countries in terms of alphabets and the population growth of the first 15 countries since 1960 by computing the difference between the two years. We created an account on Microsoft Azure platform and using basic system resources, created a MySQL database. Using SQL Workbench and the connection credentials provided by Azure, the population data was exported to the cloud. The same connection credentials were used to connect python and PHP languages for querying. The results were the relayed using a simple web application.

# REFERENCES

[1] D. Mrozek, T. Dąbek and B. Małysiak-Mrozek, "Scalable Extraction of Big Macromolecular Data in Azure Data Lake Environment", *Molecules*, vol. 24, no. 1, p. 179, 2019. Available: 10.3390/molecules24010179.

[2] "Population, total | Data", *Data.worldbank.org*, 2020. [Online]. Available: https://data.worldbank.org/indicator/SP.POP.TOTL?end=2018&start=2018&view=map. [Accessed: 26- May- 2020].

[3] D. Inan and R. Juita, "Analysis and Design Complex and Large Data Base using MySQL Workbench", *International Journal of Computer Science and Information Technology*, vol. 3, no. 5, pp. 173-183, 2011. Available: 10.5121/ijcsit.2011.3515.

[4] "Import and export - Azure Database for MySQL", *Docs.microsoft.com*, 2020. [Online]. Available: https://docs.microsoft.com/en-us/azure/mysql/concepts-migrate-import-export. [Accessed: 26- May- 2020].

[5] "Microsoft Windows Azure: Developing Applications for Highly Available Storage of Cloud Service", *International Journal of Science and Research (IJSR)*, vol. 4, no. 12, pp. 662-665, 2015. Available: 10.21275/v4i12.nov151864.

[6] K. Milan, "MICROSOFT AZURE - ONE OF THE POSSIBLE WAYS TO VIRTUALIZATION", *Trends in Education*, vol. 9, no. 1, pp. 139-147, 2016. Available: 10.5507/tvv.2016.019.