Project BCM
WORLD HAPPINESS REPORT

Data - Practical Test Paper
Presented by Kavi Persand.

Assumptions:

- Files "HR_2016.csv"and "happiNess_report_2017.csv" has "Family" and "Trust" fields as one of the main requirements for calculation of "Happiness Score" but files "2018.csv" and "report_2019.csv" have "Social Support" and "Perceptions of corruption".

- Since we are not sure what variables have been used to calculate "Family", "Trust", "Social Support" and "Perceptions of Corruption";
    - "Family" has been used as a proxy for "Social Support" and vice versa.
    - "Trust" " has been used as a proxy for "Perceptions of Corruption" and vice versa.

- In file "2018.csv", Column 8, Row 21, "United Arab Emirates" has a value of "N/A", since the average value or exact value is unavailable; it has been left as N/A.

- When taking GDP per Capita into account as a measure of happiness, it is important to note its limitations as a measure of well-being. It is agreed to further develop and strengthen indicators complementing GDP that integrate economic, social and environmental dimensions in a balanced manner.

- Happiness is something intangible, which cannot be given a number, and while the remaining variables are subjective to the country where the survey is being carried out, it is advisable to use the World Happiness Report as a guide.

- Happiness is often associated as the opposite of depression; this does not always appear to be the case. People living with mental health problems can simultaneously report feeling happy. This point can be argued by noticing a recent increase in rate of suicide in Nordic countries.

- Happiness index is also subjective to the person taking the survey, as less happy people, for example are seen as being more politically active than happy ones.

Summary

An analysis of World Happiness Report has been carried out using an open source dataset. The objective of the project includes studying the various factors which leads to the calculation of a "Happiness Index" for each country and understanding its distribution throughout the world.

The process involves ingesting the dataset in SQL, cleaning the dataset and then using it in R through RODBC server. Further, a regression model and clustering analysis has been done in R. Finally, the data has been visualised through Tableau in which the data has been imported from Microsoft SQL Server using the SQL server connection.

Data Overview

Source: The dataset is an open source dataset from a report published on world happiness, first in 2012 and thereafter every year. The happiness rankings are based on world opinion polls and are designed and developed by Gallup. There is a question called "Cantril Ladder" in the questionnaire, which requires respondents to imagine a ladder.

For these respondents, the best life is 10 points and the worst life is 0 points, and they are requested to evaluate their current life according to this standard.

The scores are from representative samples around the world, and Gallup's weight is used to make the estimation results representative. The happiness score comes from the following six factors: economic, social support, life expectancy, freedom, integrity and generosity (donation). The dataset is present here for 2016, 2017, 2018 and 2019: https://www.kaggle.com/unsdsn/world-happiness/data

Data Description:

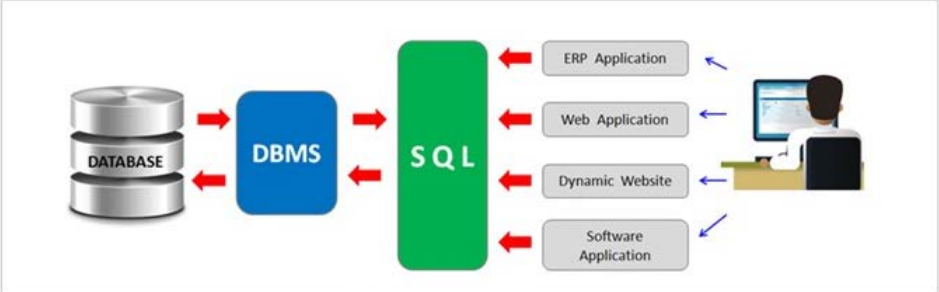
The datasets are identical except for the year they contain information of and have the following columns:

- **Country**: Name of the country
- **Region**: Region of the world, the country belongs to
- **Happiness Rank**: Rank of the country according to happiness score
- **Happiness Score**: Metric measured as a combination of various factors
- **Economy (GDP per capita):** The extent to which GDP contributes to happiness
- **Family**: The extent to which Family contributes to happiness
- **Health (Life Expectancy):** The extent to which Life Expectancy contributes to happiness
- **Freedom**: The extent to which Freedom contributes to happiness
- **Trust (Government Corruption):** The extent to which trust in government contributes to happiness
- **Generosity**: Generosity of the general public and its contribution to happiness
- **Dystopia Residual**: Contribution to Dystopia residual to happiness. Dystopia is an imaginary country that has the world's least happy people. The purpose of having this is to have a lower benchmark so that all countries do positively against it. This variable has no physical significance.

Purpose of this project

The findings in this report is going to be used by BCM Ltd; a financial institution recently set up, who wants to use the happiness indicators in its policy-making decisions with regards to its cross border activities.

The datasets consist of files "HR_2016.csv", "happiNess_report_2017.csv", "2018.csv" and "report_2019.csv". There is also a "json" file, "countries_continents_codes_flags_url.json" which contains data regarding countries which will be used for this report.
For convenience purposes the ".json" file is converted into ".csv" for import into the Database Management Sytem (DBMS).



A Database Management System (DBMS) is a software tool that enables users to manage a database easily. It allows users to access and interact with the underlying data in the database. These actions can range from simply querying data to defining database schemas that fundamentally affect the database structure.

A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

Data Loading:
Data is present in the form of 4 csv files, one for each year – 2016, 2017, 2018 and 2019. The files are imported in a Database called "**Project_BCM**" through the import task (using SQL Server 2019).

A general schema of the imported file is here (using table "dbo.Country" as example):

Table Name: dbo.Country		
Column Name	**Data Type**	**Description**
Country	VARCHAR(100)	Stores Country name. Data Type is VARCHAR(100) - stores alphanumeric data to a maximum of "100" characters.
Image_File	VARCHAR(MAX)	Stores Country Flag data. Data Type is VARCHAR(MAX) - stores alphanumeric data to a maximum of "8000" characters (for SQL Server).
Image_URL	VARCHAR(MAX)	Stores link to Country data. Data Type is VARCHAR(MAX) - stores alphanumeric data to a maximum of "8000" characters (for SQL Server).
Alpha_2	VARCHAR(100)	Stores two letter Country code defined in ISO 3166-1. Data Type is VARCHAR(100) - stores alphanumeric data to a maximum of "100" characters.
Alpha_3	VARCHAR(100)	Stores three letter Country code defined in ISO 3166-1. Data Type is VARCHAR(100) - stores alphanumeric data to a maximum of "100" characters.
Country_Code	VARCHAR(MAX)	Stores link to Country data. Data Type is VARCHAR(MAX) - stores alphanumeric data to a maximum of "8000" characters (for SQL Server).
iso_3166_2	VARCHAR(100)	Codes for identifying the principal subdivisions (e.g., provinces or states) of all countries coded in ISO 3166-1. Data Type is VARCHAR(100) - stores alphanumeric data to a maximum of "100" characters.
Region	VARCHAR(100)	Region of the globe to which country belong to. Data Type is VARCHAR(100) - stores alphanumeric data to a maximum of "100" characters.
Sub_Region	VARCHAR(100)	Sub-Region of the globe to which country belong to. Data Type is VARCHAR(100) - stores alphanumeric data to a maximum of "100" characters.
Intermediate_Region	VARCHAR(100)	Intermediate-Region of the globe to which country belong to. Data Type is VARCHAR(100) - stores alphanumeric data to a maximum of "100" characters.
Region_Code	VARCHAR(MAX)	Code belonging to that region of the globe. Data Type is VARCHAR(MAX) - stores alphanumeric data to a maximum of "8000" characters (for SQL Server).
Sub_Regiob_Code	VARCHAR(MAX)	Code belonging to that sub-region of the globe. Data Type is VARCHAR(MAX) - stores alphanumeric data to a maximum of "8000" characters (for SQL Server).
Intermediate_Region_Code	VARCHAR(MAX)	Code belonging to that intermediate-region of the globe. Data Type is VARCHAR(MAX) - stores alphanumeric data to a maximum of "8000" characters (for SQL Server).

- Schema used for table “dbo.Country”:

```
CREATE TABLE dbo.Country(
                            Country VARCHAR(100) NULL,
                            Image_File VARCHAR(MAX) NULL,
                            Image_URL VARCHAR(MAX) NULL,
                            Alpha_2 VARCHAR(100) NULL,
                            Alpha_3 VARCHAR(100) NULL,
                            Country_Code VARCHAR(MAX) NULL,
                            iso_3166_2 VARCHAR(100) NULL,
                            Region VARCHAR(100) DEFAULT 'Nan',
                            Sub_Region VARCHAR(100) DEFAULT NULL,
                            Intermediate_Region VARCHAR(100) DEFAULT NULL,
                            Region_Code VARCHAR(MAX) NULL,
                            Sub_Region_Code VARCHAR(MAX) NULL,
                            Intermediate_Region_Code VARCHAR(MAX) NULL
                            )
```

- Schema used for table “dbo.Raw_WHR_2016”:

```
CREATE TABLE dbo.Raw_WHR_2016(
                                  Country VARCHAR(100) NULL,
                                  Happiness_Score DECIMAL(6,3),
                                  Lower_Confidence_Interval DECIMAL(7,5),
                                  Upper_Confidence_Interval DECIMAL(7,5),
                                  GDP_per_Capita DECIMAL(7,5),
                                  Family DECIMAL(7,5),
                                  Health DECIMAL(7,5),
                                  Freedom DECIMAL(7,5),
                                  Trust DECIMAL(7,5),
                                  Generosity DECIMAL(7,5),
                                  Dystopia DECIMAL(7,5)
                                  )
```

- “TRUNCATE” is used to clear all rows in the table on creation before data is inserted in the table:

```
TRUNCATE TABLE dbo.Country
TRUNCATE TABLE dbo.Raw_WHR_2016
```

- “BULK INSERT ” will be used throughout “Project_BCM” to import “.csv” files in the tables / database:

```
BULK INSERT dbo.Country
FROM 'C:\Users\pkavi\Documents\MCB_Assignment\Country_List.csv'
WITH
(
       FIRSTROW=2, /* Import of data starts as from row 2, else header will be imported as well. */
       FORMAT='CSV'
)
```

-Task 1

Based on the data files provided implement a database using python library of your choice (preferably) or any other language with necessary tables, columns, data types and constraints. Create your tables in the database with appropriate naming convention.

```
/* Creating Database "Project_BCM" */

CREATE DATABASE Project_BCM

/* Using "Project_BCM" Database */

USE Project_BCM
```

-Table: “dbo.Country”

```
/* - Creating table for storing raw Country data */

CREATE TABLE dbo.Country(
                            Country VARCHAR(100) NULL,
                            Image_File VARCHAR(MAX) NULL,
                            Image_URL VARCHAR(MAX) NULL,
                            Alpha_2 VARCHAR(100) NULL,
                            Alpha_3 VARCHAR(100) NULL,
                            Country_Code VARCHAR(MAX) NULL,
                            iso_3166_2 VARCHAR(100) NULL,
                            Region VARCHAR(100) DEFAULT 'Nan',
                            Sub_Region VARCHAR(100) DEFAULT NULL,
                            Intermediate_Region VARCHAR(100) DEFAULT NULL,
                            Region_Code VARCHAR(MAX) NULL,
                            Sub_Region_Code VARCHAR(MAX) NULL,
                            Intermediate_Region_Code VARCHAR(MAX) NULL
                            )
```

-Table: “dbo.Raw_WHR_2016”

```
/* - Creating table "dbo.Raw_WHR_2016" to store raw data from "HR_2016.csv". */

CREATE TABLE dbo.Raw_WHR_2016(Country VARCHAR(100) NULL,
                                  Happiness_Score DECIMAL(6,3),
                                  Lower_Confidence_Interval DECIMAL(7,5),
                                  Upper_Confidence_Interval DECIMAL(7,5),
                                  GDP_per_Capita DECIMAL(7,5),
                                  Family DECIMAL(7,5),
                                  Health DECIMAL(7,5),
                                  Freedom DECIMAL(7,5),
                                  Trust DECIMAL(7,5),
                                  Generosity DECIMAL(7,5),
                                  Dystopia DECIMAL(7,5)
                                  )
```

-Table: “dbo.Raw_WHR_2017”

```
/* - Creating table "dbo.Raw_WHR_2017" to store raw data from "happiNess_report_2017.csv". */

CREATE TABLE dbo.Raw_WHR_2017(Country VARCHAR(100) NULL,
                                  Happiness_Score DECIMAL(6,3),
                                  Whisker_High DECIMAL(7,5),
                                  Whisker_Low DECIMAL(7,5),
                                  GDP_per_Capita DECIMAL(7,5),
                                  Family DECIMAL(7,5),
                                  Health DECIMAL(7,5),
                                  Freedom DECIMAL(7,5),
                                  Trust DECIMAL(7,5),
                                  Generosity DECIMAL(7,5),
                                  Dystopia DECIMAL(7,5)
                                  )
```

-Table: “dbo.Raw_WHR_2018”

```
/* - Creating table "dbo.Raw_WHR_2018" to store raw data from "2018.csv". */

CREATE TABLE dbo.Raw_WHR_2018(Country VARCHAR(100) NULL,
                                  Happiness_Score DECIMAL(6,3),
                                  GDP_per_Capita DECIMAL(7,5),
                                  Family DECIMAL(7,5),
                                  Health DECIMAL(7,5),
                                  Freedom DECIMAL(7,5),
                                  Generosity DECIMAL(7,5),
                                  Trust DECIMAL(7,5)
                                  )
```

-Table: “dbo.Raw_WHR_2019”

```
/* - Creating table "dbo.Raw_WHR_2019" to store raw data from "report_2019.csv". */

CREATE TABLE dbo.Raw_WHR_2019(Country VARCHAR(100) NULL,
                                    Happiness_Score DECIMAL(6,3),
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Trust DECIMAL(7,5)
                                    )
```

Before using the tables, we need to “TRUNCATE” each tables in the database so as to remove an residual data in the rows before importing the data from our “.csv” files.

```
/* Using "TRUNCATE" to remove all rows (data) from a table and "BULK INSERT" to populate the tables.
*/

TRUNCATE TABLE dbo.Country
TRUNCATE TABLE dbo.Raw_WHR_2016
TRUNCATE TABLE dbo.Raw_WHR_2017
TRUNCATE TABLE dbo.Raw_WHR_2018
TRUNCATE TABLE dbo.Raw_WHR_2019
```

DATA EXTRACTION

-Task 2

Develop an automated data pipeline (using python library of your choice (preferably) or any other language) to trigger the process that will consume the files and load them in the tables that you created in step 1 with proper data format. You are expected to create a package with appropriate functions or procedures. All the objects that you need to create shall be available on your working environment and properly compiled.

```
/* Inserting data into "dbo.Country" from "Country_List.csv". */

BULK INSERT dbo.Country
FROM 'C:\Users\pkavi\Documents\MCB_Assignment\Country_List.csv'
WITH
(
	FIRSTROW=2, /* Import of data starts as from row 2, else header will be imported as well. */
	FORMAT='CSV'
)

/* List the data which has been inserted in table "dbo.Country". */

SELECT *
FROM dbo.Country /* There are 273 records in table "dbo.Country" */
```

Results Messages

	Country	Image_File	Image_URL	Alpha_2	Alpha_3	Country_Code	iso_3166_2	Region	Sub_Region	Intermediate_Regi
1	Afghanistan	Flag_of_Afghanistan.svg	https://upload.wikimedia.org/wikipedia/commons/9...	AF	AFG	4	ISO 3166-2:AF	Asia	Southern Asia	NULL
2	Albania	Flag_of_Albania.svg	https://upload.wikimedia.org/wikipedia/commons/3...	AL	ALB	8	ISO 3166-2:AL	Europe	Southern Europe	NULL
3	Algeria	Flag_of_Algeria.svg	https://upload.wikimedia.org/wikipedia/commons/7...	DZ	DZA	12	ISO 3166-2:DZ	Africa	Northern Africa	NULL
4	Andorra	Flag_of_Andorra.svg	https://upload.wikimedia.org/wikipedia/commons/1...	AD	AND	20	ISO 3166-2:AD	Europe	Southern Europe	NULL
5	Angola	Flag_of_Angola.svg	https://upload.wikimedia.org/wikipedia/commons/9...	AO	AGO	24	ISO 3166-2:AO	Africa	Sub-Saharan Africa	Middle Africa
6	Antigua and Barbuda	Flag_of_Antigua_and_Barbuda.svg	https://upload.wikimedia.org/wikipedia/commons/8...	AG	ATG	28	ISO 3166-2:AG	Americas	Latin America and the Caribbean	Caribbean
7	Argentina	Flag_of_Argentina.svg	https://upload.wikimedia.org/wikipedia/commons/1...	AR	ARG	32	ISO 3166-2:AR	Americas	Latin America and the Caribbean	South America
8	Armenia	Flag_of_Armenia.svg	https://upload.wikimedia.org/wikipedia/commons/2...	AM	ARM	51	ISO 3166-2:AM	Asia	Western Asia	NULL
9	Australia	Flag_of_Australia_%28converted%29.svg	https://upload.wikimedia.org/wikipedia/commons/8...	AU	AUS	36	ISO 3166-2:AU	Oceania	Australia and New Zealand	NULL
10	Austria	Flag_of_Austria.svg	https://upload.wikimedia.org/wikipedia/commons/4...	AT	AUT	40	ISO 3166-2:AT	Europe	Western Europe	NULL

Query executed successfully. DESKTOP-S2LJ53N\SQLEXPRESS ... DESKTOP-S2LJ53N\pkavi ... Project_BCM 00:00:00 273 rows

```
/* Inserting data into the report tables. */

/* - Inserting data into "dbo.Raw_WHR_2016" from "HR_2016.csv". */

BULK INSERT dbo.Raw_WHR_2016
FROM 'C:\Users\pkavi\Documents\MCB_Assignment\Data Files\HR_2016.csv'
WITH
(

	FIRSTROW=2, /* Import of data starts as from row 2, else header will be imported as well. */
	FORMAT='CSV'

)

/* List all the data which has been inserted in table "dbo.Raw_WHR_2016". */

SELECT *
FROM dbo.Raw_WHR_2016  /* There are 157 records in table "dbo.Raw_WHR_2016" */
```

Results Messages

	Country	Happiness_Score	Lower_Confidence_Interval	Upper_Confidence_Interval	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Denmark	7.526	7.46000	7.59200	1.44178	1.16374	0.79504	0.57941	0.44453	0.36171	2.73939
2	Switzerland	7.509	7.42800	7.59000	1.52733	1.14524	0.86303	0.58557	0.41203	0.28083	2.69463
3	Iceland	7.501	7.33300	7.66900	1.42666	1.18326	0.86733	0.56624	0.14975	0.47678	2.83137
4	Norway	7.498	7.42100	7.57500	1.57744	1.12690	0.79579	0.59609	0.35776	0.37895	2.66465
5	Finland	7.413	7.35100	7.47500	1.40598	1.13464	0.81091	0.57104	0.41004	0.25492	2.82596
6	Canada	7.404	7.33500	7.47300	1.44015	1.09610	0.82760	0.57370	0.31329	0.44834	2.70485
7	Netherlands	7.339	7.28400	7.39400	1.46468	1.02912	0.81231	0.55211	0.29927	0.47416	2.70749
8	New Zealand	7.334	7.26400	7.40400	1.36066	1.17278	0.83096	0.58147	0.41904	0.49401	2.47553
9	Australia	7.313	7.24100	7.38500	1.44443	1.10476	0.85120	0.56837	0.32331	0.47407	2.54650
10	Sweden	7.291	7.22700	7.35500	1.45181	1.08764	0.83121	0.58218	0.40867	0.38254	2.54734
11	Israel	7.267	7.19900	7.33500	1.33766	0.99537	0.84917	0.36432	0.08728	0.32288	3.31029

Query executed successfully. DESKTOP-S2LJ53N\SQLEXPRESS ... DESKTOP-S2LJ53N\pkavi ... Project_BCM 00:00:00 157 rows

```
/* - Inserting data into "dbo.Raw_WHR_2017" from "happiNess_report_2017.csv". */

BULK INSERT dbo.Raw_WHR_2017
FROM 'C:\Users\pkavi\Documents\MCB_Assignment\Data Files\ happiNess_report_2017.csv'
WITH
(

	FIRSTROW=2, /* Import of data starts as from row 2, else header will be imported as well. */
	FORMAT='CSV'

)

/* List all the data which has been inserted in table "dbo.Raw_WHR_2017". */

SELECT *
FROM dbo.Raw_WHR_2017  /* There are 155 records in table "dbo.Raw_WHR_2017" */
```

Results | Messages

	Country	Happiness_Score	Whisker_High	Whisker_Low	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Norway	7.537	7.59444	7.47956	1.61646	1.53352	0.79667	0.63542	0.36201	0.31596	2.27703
2	Denmark	7.522	7.58173	7.46227	1.48238	1.55112	0.79257	0.62601	0.35528	0.40077	2.31371
3	Iceland	7.504	7.62203	7.38597	1.48063	1.61057	0.83355	0.62716	0.47554	0.15353	2.32272
4	Switzerland	7.494	7.56177	7.42623	1.56498	1.51691	0.85813	0.62007	0.29055	0.36701	2.27672
5	Finland	7.469	7.52754	7.41046	1.44357	1.54025	0.80916	0.61795	0.24548	0.38261	2.43018
6	Netherlands	7.377	7.42743	7.32657	1.50394	1.42894	0.81070	0.58538	0.47049	0.28266	2.29480

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 155 rows

```
/* Using "TRUNCATE" to remove all rows (data) from a table and "BULK INSERT" to populate the tables.
*/

TRUNCATE TABLE dbo.Raw_WHR_2018

/* Inserting data into the report tables. */

/* - Inserting data into "dbo.Raw_WHR_2018" from "2018.csv". */

BULK INSERT dbo.Raw_WHR_2018
FROM 'C:\Users\pkavi\Documents\MCB_Assignment\Data Files\2018.csv'
WITH
(

	FIRSTROW=2, /* Import of data starts as from row 2, else header will be imported as well. */
	FORMAT='CSV'
	/* Converting "Trust" column to VARCHAR(20) using CAST([Trust] AS VARCHAR(20)) */

)

/* List all the data which has been inserted in table "dbo.WHR_2018". */

SELECT *
FROM dbo.Raw_WHR_2018  /* There are 156 records in table "dbo.Raw_WHR_2018" */
```

Results | Messages

	Country	Happiness_Score	GDP_per_Capita	Family	Health	Freedom	Generosity	Trust
1	Finland	7.632	1.30500	1.59200	0.87400	0.68100	0.20200	0.393
2	Norway	7.594	1.45600	1.58200	0.86100	0.68600	0.28600	0.34
3	Denmark	7.555	1.35100	1.59000	0.86800	0.68300	0.28400	0.408
4	Iceland	7.495	1.34300	1.64400	0.91400	0.67700	0.35300	0.138
5	Switzerland	7.487	1.42000	1.54900	0.92700	0.66000	0.25600	0.357
6	Netherlands	7.441	1.36100	1.48800	0.87800	0.63800	0.33300	0.295
7	Canada	7.328	1.33000	1.53200	0.89600	0.65300	0.32100	0.291

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 156 rows

```
/* Using "TRUNCATE" to remove all rows (data) from a table and "BULK INSERT" to populate the tables.
*/

TRUNCATE TABLE dbo.Raw_WHR_2019

/* - Inserting data into "dbo.Raw_WHR_2019" from "report_2019.csv". */
```

```
BULK INSERT dbo.Raw_WHR_2019
FROM 'C:\Users\pkavi\Documents\MCB_Assignment\Data Files\report_2019.csv'
WITH
(

	FIRSTROW=2, /* Import of data starts as from row 2, else header will be imported as well. */
	FORMAT='CSV'

)

/* List all the data which has been inserted in table "dbo.WHR_2019". */

SELECT *
FROM dbo.Raw_WHR_2019  /* There are 156 records in table "dbo.Raw_WHR_2019" */
```

Results | Messages

	Country	Happiness_Score	GDP_per_Capita	Family	Health	Freedom	Generosity	Trust
1	Finland	7.769	1.34000	1.58700	0.98600	0.59600	0.15300	0.39300
2	Denmark	7.600	1.38300	1.57300	0.99600	0.59200	0.25200	0.41000
3	Norway	7.554	1.48800	1.58200	1.02800	0.60300	0.27100	0.34100
4	Iceland	7.494	1.38000	1.62400	1.02600	0.59100	0.35400	0.11800
5	Netherlands	7.488	1.39600	1.52200	0.99900	0.55700	0.32200	0.29800
6	Switzerland	7.480	1.45200	1.52600	1.05200	0.57200	0.26300	0.34300
7	Sweden	7.343	1.38700	1.48700	1.00900	0.57400	0.26700	0.37300

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 156 rows

DATA TRANSFORMATION

-Task 3

The data scientist wishes to have a modelling record in both csv and parquet format containing the details as per table below. Complement the existing data pipeline in step 2 to return the required information.

Column Name	Specifications
Year	
Country	
Country Url	
Region Code	
Region	UPPER CASE. To default to 'Nan' if not available.
Rank Per Region	Per Region Per Year
Overall Rank	Per Year
Happiness Score	As per record.
Happiness Status	Return the following value based on the Happiness Score If Happiness Score is > 5.6, display "Green". If Happiness Score is between 2.6 and 5.6, display "Amber". If Happiness Score is <2.6, display "Red".
GDP per capita	As per record.
Family	As per record.
Social support	As per record.
Healthy life expectancy	As per record.
Freedom to make life choices	As per record.
Generosity	As per record.
Perceptions of corruption	As per record.

Format above consist of 16 fields, namely; "Year", "Country", "Country_URL", "Region_Code", "Region", "Rank_per_Region", "Overall_Rank", "Happiness_Score", "Happiness_Status","GDP_per_Capita", "Family","Social_Support","Healthy_Life_Expectancy", "Freedom_to_make_Life_choices", "Generosity" and "Perceptions_of_Corruption".

Step 1. Rank.

- "Overall Rank" and "Rank Per Region" are found based on the "Happiness Score" of a Country.

```
RANK() OVER(

    ORDER BY Happiness_Score DESC
    ) AS Overall_Happiness_Rank,
```

Step 2. Happiness Status.

- "Happiness Status" of a Country is found according to the Country's "Happiness Score".

```
(CASE WHEN Happiness_Score < 2.6 THEN 'RED'
WHEN Happiness_Score BETWEEN 2.6 AND 5.6 THEN 'AMBER'
WHEN Happiness_Score > 5.6 THEN 'GREEN'
END) AS Happiness_Status,
```

Using table “dbo.Raw_WHR_2016” as example, we are going to implement Step 1 and Step 2 to list the countries in the table according to its respective “Happiness_Rank” and “Happiness_Status”.

```
SELECT 2016 AS Year, Country, Happiness_Score,

        RANK() OVER(

            ORDER BY Happiness_Score DESC
            ) AS Overall_Happiness_Rank,

            (CASE WHEN Happiness_Score < 2.6 THEN 'RED'
            WHEN Happiness_Score BETWEEN 2.6 AND 5.6 THEN 'AMBER'
            WHEN Happiness_Score > 5.6 THEN 'GREEN'
            END) AS Happiness_Status,

        Lower_Confidence_Interval, Upper_Confidence_Interval, GDP_per_Capita,
        Family, Health, Freedom, Trust, Generosity, Dystopia

FROM dbo.Raw_WHR_2016
```

Results | Messages

	Year	Country	Happiness_Score	Overall_Happiness_Rank	Happiness_Status	Lower_Confidence_Interval	Upper_Confidence_Interval	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	2016	Denmark	7.526	1	GREEN	7.46000	7.59200	1.44178	1.16374	0.79504	0.57941	0.44453	0.36171	2.73939
2	2016	Switzerland	7.509	2	GREEN	7.42800	7.59000	1.52733	1.14524	0.86303	0.58557	0.41203	0.28083	2.69463
3	2016	Iceland	7.501	3	GREEN	7.33300	7.66900	1.42666	1.18326	0.86733	0.56624	0.14975	0.47678	2.83137
4	2016	Norway	7.498	4	GREEN	7.42100	7.57500	1.57744	1.12690	0.79579	0.59609	0.35776	0.37895	2.66465
5	2016	Finland	7.413	5	GREEN	7.35100	7.47500	1.40598	1.13464	0.81091	0.57104	0.41004	0.25492	2.82596
6	2016	Canada	7.404	6	GREEN	7.33500	7.47300	1.44015	1.09610	0.82760	0.57370	0.31329	0.44834	2.70485
7	2016	Netherlan...	7.339	7	GREEN	7.28400	7.39400	1.46468	1.02912	0.81231	0.55211	0.29927	0.47416	2.70749
8	2016	New Zeal...	7.334	8	GREEN	7.26400	7.40400	1.36066	1.17278	0.83096	0.58147	0.41904	0.49401	2.47553

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 157 rows

Creating a new table “newRaw_WHR_2016” to list countries from initial table “dbo.Raw_WHR_2016” with “Happiness_Rank” and “Happiness_Status”.

```
/* - Creating table "dbo.newRaw_WHR_2016" to store raw data from "HR_2016.csv" together with
Overall_Happiness_Rank and Happiness_Status of Country. */

CREATE TABLE dbo.newRaw_WHR_2016(
                                Year INT DEFAULT NULL,
                                Country VARCHAR(100) NULL,
                                Happiness_Score DECIMAL(6,3),
                                Overall_Happiness_Rank INT DEFAULT NULL,
                                Happiness_Status VARCHAR(20) NULL,
                                Lower_Confidence_Interval DECIMAL(7,5),
                                Upper_Confidence_Interval DECIMAL(7,5),
                                GDP_per_Capita DECIMAL(7,5),
                                Family DECIMAL(7,5),
                                Health DECIMAL(7,5),
                                Freedom DECIMAL(7,5),
                                Trust DECIMAL(7,5),
                                Generosity DECIMAL(7,5),
                                Dystopia DECIMAL(7,5)
                                )

/* Using "TRUNCATE" to remove all rows (data) from a table. */

TRUNCATE TABLE dbo.newRaw_WHR_2016
```

```sql
/* Inserting data in table "dbo.newRaw_WHR_2016" together with Overall_Happiness_Rank of Country. */

INSERT INTO dbo.newRaw_WHR_2016

	SELECT 2016 AS Year, Country, Happiness_Score,

			RANK() OVER(

				ORDER BY Happiness_Score DESC
				) AS Overall_Happiness_Rank,

				(CASE WHEN Happiness_Score < 2.6 THEN 'RED'
				WHEN Happiness_Score BETWEEN 2.6 AND 5.6 THEN 'AMBER'
				WHEN Happiness_Score > 5.6 THEN 'GREEN'
				END) AS Happiness_Status,

			Lower_Confidence_Interval, Upper_Confidence_Interval, GDP_per_Capita,
			Family, Health, Freedom, Trust, Generosity, Dystopia

	FROM dbo.Raw_WHR_2016

/* List all the data which has been inserted in table "dbo.newRaw_WHR_2016" as per Task 3. */

SELECT *
FROM dbo.newRaw_WHR_2016  /* There are 157 records in table "dbo.newRaw_WHR_2016" */
```

Results | Messages

	Year	Country	Happiness_Score	Overall_Happiness_Rank	Happiness_Status	Lower_Confidence_Interval	Upper_Confidence_Interval	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	2016	Denmark	7.526	1	GREEN	7.46000	7.59200	1.44178	1.16374	0.79504	0.57941	0.44453	0.36171	2.73939
2	2016	Switzerland	7.509	2	GREEN	7.42800	7.59000	1.52733	1.14524	0.86303	0.58557	0.41203	0.28083	2.69463
3	2016	Iceland	7.501	3	GREEN	7.33300	7.66900	1.42666	1.18326	0.86733	0.56624	0.14975	0.47678	2.83137
4	2016	Norway	7.498	4	GREEN	7.42100	7.57500	1.57744	1.12690	0.79579	0.59609	0.35776	0.37895	2.66465
5	2016	Finland	7.413	5	GREEN	7.35100	7.47500	1.40598	1.13464	0.81091	0.57104	0.41004	0.25492	2.82596
6	2016	Canada	7.404	6	GREEN	7.33500	7.47300	1.44015	1.09610	0.82760	0.57370	0.31329	0.44834	2.70485
7	2016	Netherlan...	7.339	7	GREEN	7.28400	7.39400	1.46468	1.02912	0.81231	0.55211	0.29927	0.47416	2.70749
8	2016	New Zeal...	7.334	8	GREEN	7.26400	7.40400	1.36066	1.17278	0.83096	0.58147	0.41904	0.49401	2.47553

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 157 rows

STEP 3. Region.

- Using “ORDER BY Region” to find the list of “Region” from the table “dbo.Country”.

```sql
SELECT *
FROM dbo.Country
ORDER BY Region /* There are 6 regions in "dbo.Country" namely "Africa", "Americas", "Asia", "Europe",
"Nan" and "Oceania". */
```

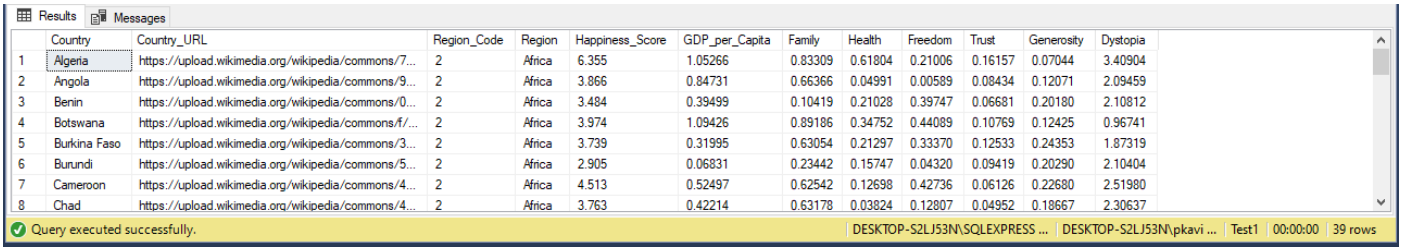
From the above statement, we can deduce that there are “5” regions and “Nan” that will accommodate for countries who’s region is “NULL”.

Regions:
- Africa,
- Americas,
- Asia,
- Europe,
- “Nan”,
- Oceania.

Countries in table “dbo.Raw_WHR_2016” are then divided into their respective regions, so as to be able to group them again according to “Region”, Region Code - “Region_Code” and Rank Per Region - “Regional_Happiness_Rank”. Same is repeated for “dbo.Raw_WHR_2017”, “dbo.Raw_WHR_2018” and “dbo.Raw_WHR_2019”.

- Using table “dbo.Raw_WHR_2016” as example, we are going to create tables “dbo.Raw_WHR_Africa_2016”, “dbo.Raw_WHR_Americas_2016”, “dbo.Raw_WHR_Asia_2016”, “dbo.Raw_WHR_Europe_2016”, “dbo.Raw_WHR_Nan_2016” and “dbo.Raw_WHR_Oceania_2016” for dividing the report into the different regions.

```
--- AFRICA ---

/* Breaking "dbo.Raw_WHR_2016" into separate regions; namely "Africa", "Americas", "Asia", "Europe",
"Nan" and "Oceania"; using "Africa" as example. */

SELECT a.Country, a.Image_URL AS Country_URL, a.Region_Code, a.Region, b.Happiness_Score,
b.GDP_per_Capita, b.Family, b.Health, b.Freedom, b.Trust, b.Generosity, b.Dystopia
FROM dbo.Country a
INNER JOIN dbo.Raw_WHR_2016 b
ON a.Country=b.Country
WHERE a.Region='Africa'  /* 39 rows displayed */
```



Results | Messages

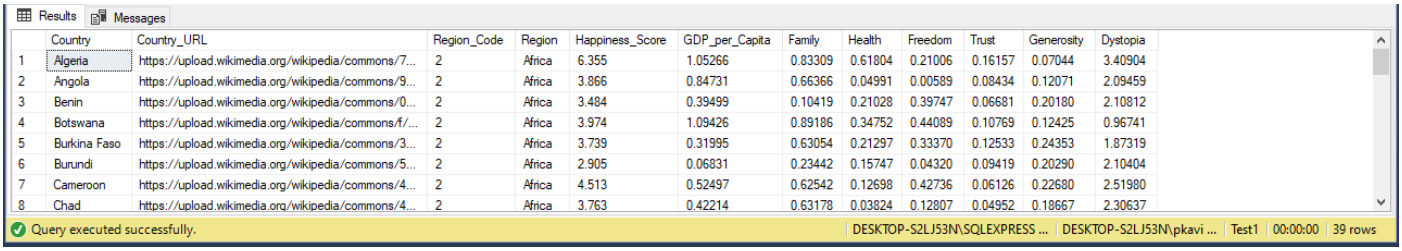
	Country	Country_URL	Region_Code	Region	Happiness_Score	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Algeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	Africa	6.355	1.05266	0.83309	0.61804	0.21006	0.16157	0.07044	3.40904
2	Angola	https://upload.wikimedia.org/wikipedia/commons/9...	2	Africa	3.866	0.84731	0.66366	0.04991	0.00589	0.08434	0.12071	2.09459
3	Benin	https://upload.wikimedia.org/wikipedia/commons/0...	2	Africa	3.484	0.39499	0.10419	0.21028	0.39747	0.06681	0.20180	2.10812
4	Botswana	https://upload.wikimedia.org/wikipedia/commons/f/...	2	Africa	3.974	1.09426	0.89186	0.34752	0.44089	0.10769	0.12425	0.96741
5	Burkina Faso	https://upload.wikimedia.org/wikipedia/commons/3...	2	Africa	3.739	0.31995	0.63054	0.21297	0.33370	0.12533	0.24353	1.87319
6	Burundi	https://upload.wikimedia.org/wikipedia/commons/5...	2	Africa	2.905	0.06831	0.23442	0.15747	0.04320	0.09419	0.20290	2.10404
7	Cameroon	https://upload.wikimedia.org/wikipedia/commons/4...	2	Africa	4.513	0.52497	0.62542	0.12698	0.42736	0.06126	0.22680	2.51980
8	Chad	https://upload.wikimedia.org/wikipedia/commons/4...	2	Africa	3.763	0.42214	0.63178	0.03824	0.12807	0.04952	0.18667	2.30637

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 39 rows

Table: "dbo.Raw_WHR_Africa_2016".

Creating table "dbo.Raw_WHR_Africa_2016" to store data pertaining to countries from the "Africa" region (using table “dbo.Raw_WHR_2016”).

```
/* Creating table "dbo.Raw_WHR_Africa_2016" to store data pertaining to countries from the "Africa"
region. */

CREATE TABLE dbo.Raw_WHR_Africa_2016(
                                    Country VARCHAR(100) NULL,
                                    Country_URL VARCHAR(MAX) NULL,
                                    Region_Code VARCHAR(MAX) NULL,
                                    Region VARCHAR(100) DEFAULT 'Nan',
                                    Happiness_Score DECIMAL(6,3),
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Trust DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Dystopia DECIMAL(7,5)
                                    )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.Raw_WHR_Africa_2016". */

TRUNCATE TABLE dbo.Raw_WHR_Africa_2016

/* - Inserting data into "dbo.Raw_WHR_Africa_2016" from "dbo.Raw_WHR_2016". */

INSERT INTO dbo.Raw_WHR_Africa_2016

SELECT a.Country, a.Image_URL AS Country_URL, a.Region_Code, a.Region, b.Happiness_Score,
b.GDP_per_Capita, b.Family, b.Health, b.Freedom, b.Trust, b.Generosity, b.Dystopia
FROM dbo.Country a
INNER JOIN dbo.Raw_WHR_2016 b
ON a.Country=b.Country
WHERE a.Region='Africa'  /* 39 rows inserted */

/* List all the data in table "dbo.Raw_WHR_Africa_2016". */

SELECT *
FROM dbo.Raw_WHR_Africa_2016
```



Results Messages

	Country	Country_URL	Region_Code	Region	Happiness_Score	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Algeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	Africa	6.355	1.05266	0.83309	0.61804	0.21006	0.16157	0.07044	3.40904
2	Angola	https://upload.wikimedia.org/wikipedia/commons/9...	2	Africa	3.866	0.84731	0.66366	0.04991	0.00589	0.08434	0.12071	2.09459
3	Benin	https://upload.wikimedia.org/wikipedia/commons/0...	2	Africa	3.484	0.39499	0.10419	0.21028	0.39747	0.06681	0.20180	2.10812
4	Botswana	https://upload.wikimedia.org/wikipedia/commons/f/...	2	Africa	3.974	1.09426	0.89186	0.34752	0.44089	0.10769	0.12425	0.96741
5	Burkina Faso	https://upload.wikimedia.org/wikipedia/commons/3...	2	Africa	3.739	0.31995	0.63054	0.21297	0.33370	0.12533	0.24353	1.87319
6	Burundi	https://upload.wikimedia.org/wikipedia/commons/5...	2	Africa	2.905	0.06831	0.23442	0.15747	0.04320	0.09419	0.20290	2.10404
7	Cameroon	https://upload.wikimedia.org/wikipedia/commons/4...	2	Africa	4.513	0.52497	0.62542	0.12698	0.42736	0.06126	0.22680	2.51980
8	Chad	https://upload.wikimedia.org/wikipedia/commons/4...	2	Africa	3.763	0.42214	0.63178	0.03824	0.12807	0.04952	0.18667	2.30637

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 39 rows

Step 4. UPPERCASE

- List values in “Region” column in uppercase.

This can be done by using the SQL Server UPPER() Function.
Example: UPPER(“Expression”) or UPPER(“Column Name”).

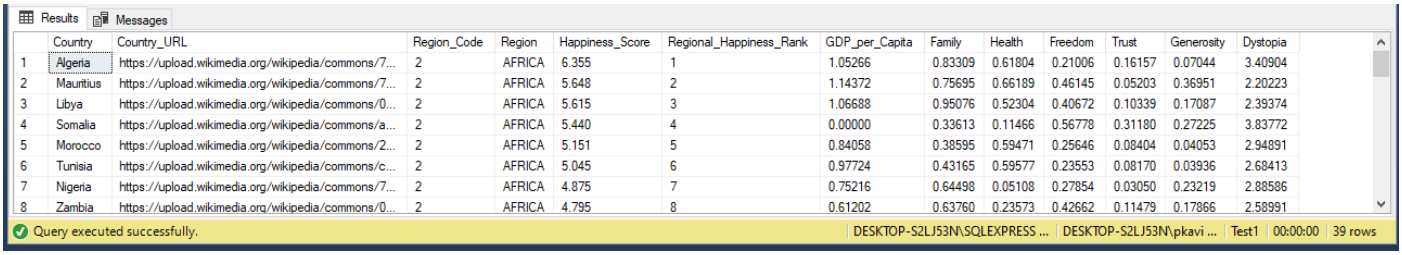
Step 5. Rank Per Region.

- “Rank Per Region” can be found based on the “Happiness Score” of a Country.

```
RANK() OVER(

    ORDER BY Happiness_Score DESC
    ) AS Overall_Happiness_Rank,
```

List of countries in the “dbo.Raw_WHR_Africa_2016” are grouped by Region, which is the “Africa” region.
Using the Rank() Function, we can find the rank of each country in the table.

```
/* --Region in UPPERCASE and Regional Rank for table "dbo.Raw_WHR_Africa_2016". */

SELECT Country, UPPER(Region) AS Region, Happiness_Score,

            RANK() OVER(

                    ORDER BY Happiness_Score DESC
                    ) AS Regional_Happiness_Rank,

            GDP_per_Capita, Family, Health, Freedom,
            Trust, Generosity, Dystopia

FROM dbo.Raw_WHR_Africa_2016
```



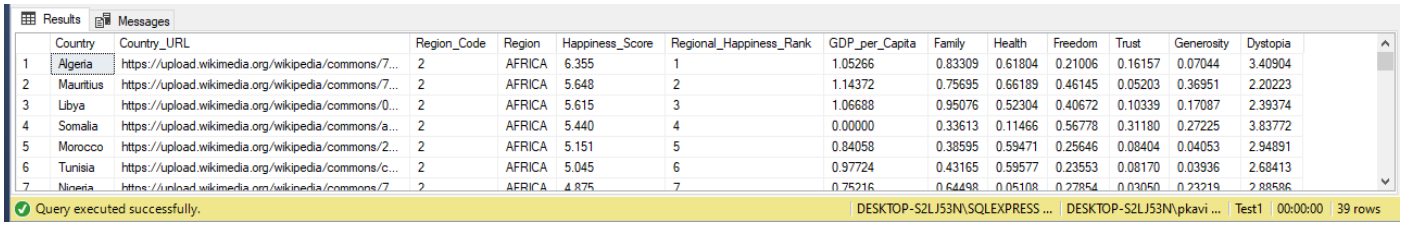
Results Messages

	Country	Country_URL	Region_Code	Region	Happiness_Score	Regional_Happiness_Rank	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Algeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	6.355	1	1.05266	0.83309	0.61804	0.21006	0.16157	0.07044	3.40904
2	Mauritius	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	5.648	2	1.14372	0.75695	0.66189	0.46145	0.05203	0.36951	2.20223
3	Libya	https://upload.wikimedia.org/wikipedia/commons/0...	2	AFRICA	5.615	3	1.06688	0.95076	0.52304	0.40672	0.10339	0.17087	2.39374
4	Somalia	https://upload.wikimedia.org/wikipedia/commons/a...	2	AFRICA	5.440	4	0.00000	0.33613	0.11466	0.56778	0.31180	0.27225	3.83772
5	Morocco	https://upload.wikimedia.org/wikipedia/commons/2...	2	AFRICA	5.151	5	0.84058	0.38595	0.59471	0.25646	0.08404	0.04053	2.94891
6	Tunisia	https://upload.wikimedia.org/wikipedia/commons/c...	2	AFRICA	5.045	6	0.97724	0.43165	0.59577	0.23553	0.08170	0.03936	2.68413
7	Nigeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	4.875	7	0.75216	0.64498	0.05108	0.27854	0.03050	0.23219	2.88586
8	Zambia	https://upload.wikimedia.org/wikipedia/commons/0...	2	AFRICA	4.795	8	0.61202	0.63760	0.23573	0.42662	0.11479	0.17866	2.58991

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 39 rows

A new table, "dbo.WHR_Africa_2016" is created, to reflect the implemented changes in Steps 3 – 5.

```
/* Creating table "dbo.WHR_Africa_2016" to store data pertaining to countries from the "Africa"
region in UPPERCASE with "Regional_Happiness_Rank". */

CREATE TABLE dbo.WHR_Africa_2016(
                                    Country VARCHAR(100) NULL,
                                    Country_URL VARCHAR(MAX) NULL,
                                    Region_Code VARCHAR(MAX) NULL,
                                    Region VARCHAR(100) DEFAULT 'Nan',
                                    Happiness_Score DECIMAL(6,3),
                                    Regional_Happiness_Rank INT DEFAULT NULL,
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Trust DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Dystopia DECIMAL(7,5)
                                    )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.WHR_Africa_2016". */

TRUNCATE TABLE dbo.WHR_Africa_2016

/* - Inserting data into "dbo.WHR_Africa_2016" from "dbo.Raw_WHR_Africa_2016". */

INSERT INTO dbo.WHR_Africa_2016

SELECT Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,

            RANK() OVER(

                        ORDER BY Happiness_Score DESC
                        ) AS Regional_Happiness_Rank,

            GDP_per_Capita, Family, Health, Freedom, Trust, Generosity, Dystopia

FROM dbo.Raw_WHR_Africa_2016  /* 39 rows inserted */

/* List all the data in table "dbo.WHR_Africa_2016". */

SELECT *
FROM dbo.WHR_Africa_2016
```



Results | Messages

	Country	Country_URL	Region_Code	Region	Happiness_Score	Regional_Happiness_Rank	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Algeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	6.355	1	1.05266	0.83309	0.61804	0.21006	0.16157	0.07044	3.40904
2	Mauritius	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	5.648	2	1.14372	0.75695	0.66189	0.46145	0.05203	0.36951	2.20223
3	Libya	https://upload.wikimedia.org/wikipedia/commons/0...	2	AFRICA	5.615	3	1.06688	0.95076	0.52304	0.40672	0.10339	0.17087	2.39374
4	Somalia	https://upload.wikimedia.org/wikipedia/commons/a...	2	AFRICA	5.440	4	0.00000	0.33613	0.11466	0.56778	0.31180	0.27225	3.83772
5	Morocco	https://upload.wikimedia.org/wikipedia/commons/2...	2	AFRICA	5.151	5	0.84058	0.38595	0.59471	0.25646	0.08404	0.04053	2.94891
6	Tunisia	https://upload.wikimedia.org/wikipedia/commons/c...	2	AFRICA	5.045	6	0.97724	0.43165	0.59577	0.23553	0.08170	0.03936	2.68413
7	Nigeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	4.875	7	0.75216	0.64498	0.05108	0.27854	0.03050	0.23219	2.88586

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 39 rows

Similarly, table "dbo.Raw_WHR_Americas_2016" is created, to accommodate countries belonging to the Americas region.

```
--- AMERICAS ---

/* Breaking "dbo.Raw_WHR_2016" into separate regions; namely "Africa", "Americas", "Asia", "Europe",
"Nan" and "Oceania"; using "Americas" as example. */

SELECT a.Country, a.Image_URL AS Country_URL, a.Region_Code, a.Region, b.Happiness_Score,
b.GDP_per_Capita, b.Family, b.Health, b.Freedom, b.Trust, b.Generosity, b.Dystopia
FROM dbo.Country a
INNER JOIN dbo.Raw_WHR_2016 b
ON a.Country=b.Country
WHERE a.Region='Americas'  /* 26 rows displayed */
```

```
/* Creating table "dbo.Raw_WHR_Americas_2016" to store data pertaining to countries from the
"Americas" region. */

CREATE TABLE dbo.Raw_WHR_Americas_2016(
                                        Country VARCHAR(100) NULL,
                                        Country_URL VARCHAR(MAX) NULL,
                                        Region_Code VARCHAR(MAX) NULL,
                                        Region VARCHAR(100) DEFAULT 'Nan',
                                        Happiness_Score DECIMAL(6,3),
                                        GDP_per_Capita DECIMAL(7,5),
                                        Family DECIMAL(7,5),
                                        Health DECIMAL(7,5),
                                        Freedom DECIMAL(7,5),
                                        Trust DECIMAL(7,5),
                                        Generosity DECIMAL(7,5),
                                        Dystopia DECIMAL(7,5)
                                        )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.Raw_WHR_Americas_2016". */

TRUNCATE TABLE dbo.Raw_WHR_Americas_2016

/* - Inserting data into "dbo.Raw_WHR_Americas_2016" from "dbo.Raw_WHR_2016". */

INSERT INTO dbo.Raw_WHR_Americas_2016

SELECT a.Country, a.Image_URL AS Country_URL, a.Region_Code, a.Region, b.Happiness_Score,
b.GDP_per_Capita, b.Family, b.Health, b.Freedom, b.Trust, b.Generosity, b.Dystopia
FROM dbo.Country a
INNER JOIN dbo.Raw_WHR_2016 b
ON a.Country=b.Country
WHERE a.Region='Americas'  /* 26 rows inserted */

/* List all the data in table "dbo.Raw_WHR_Americas_2016". */

SELECT *
FROM dbo.Raw_WHR_Americas_2016
```

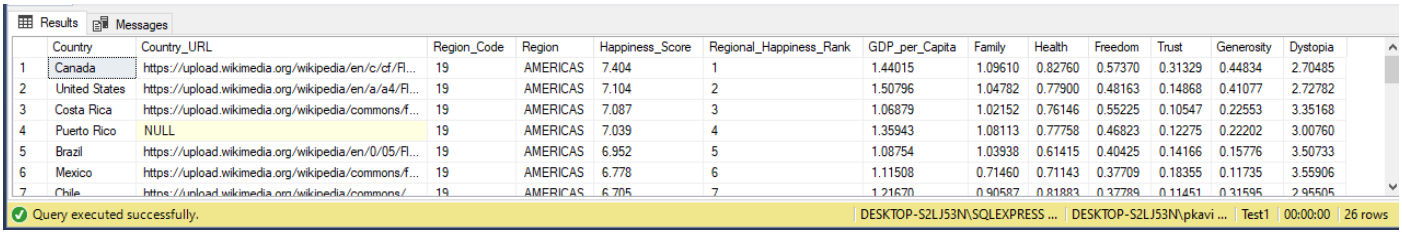
	Country	Country_URL	Region_Code	Region	Happiness_Score	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Argentina	https://upload.wikimedia.org/wikipedia/commons/1...	19	Americas	6.650	1.15137	1.06612	0.69711	0.42284	0.07296	0.10989	3.12985
2	Belize	https://upload.wikimedia.org/wikipedia/commons/e...	19	Americas	5.956	0.87616	0.68655	0.45569	0.51231	0.10771	0.23684	3.08039
3	Bolivia	https://upload.wikimedia.org/wikipedia/commons/4...	19	Americas	5.822	0.79422	0.83779	0.46970	0.50961	0.07746	0.21698	2.91635
4	Brazil	https://upload.wikimedia.org/wikipedia/en/0/05/Fla...	19	Americas	6.952	1.08754	1.03938	0.61415	0.40425	0.14166	0.15776	3.50733
5	Canada	https://upload.wikimedia.org/wikipedia/en/c/cf/Fla...	19	Americas	7.404	1.44015	1.09610	0.82760	0.57370	0.31329	0.44834	2.70485
6	Chile	https://upload.wikimedia.org/wikipedia/commons/7...	19	Americas	6.705	1.21670	0.90587	0.81883	0.37789	0.11451	0.31595	2.95505
7	Colombia	https://upload.wikimedia.org/wikipedia/commons/2...	19	Americas	6.481	1.03032	1.02169	0.59659	0.44735	0.05399	0.15626	3.17471
8	Costa R...	https://upload.wikimedia.org/wikipedia/commons/f/...	19	Americas	7.087	1.06879	1.02152	0.76146	0.55225	0.10547	0.22553	3.35168

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 26 rows

Table “dbo.WHR_Americas_2016” is created to reflect the implemented steps (steps 3 - 5) performed on table “dbo.Raw_WHR_Americas_2016” as below;

```
/* Creating table "dbo.WHR_Americas_2016" to store data pertaining to countries from the "Americas"
region in UPPERCASE with "Regional_Happiness_Rank". */

CREATE TABLE dbo.WHR_Americas_2016(
                                    Country VARCHAR(100) NULL,
                                    Country_URL VARCHAR(MAX) NULL,
                                    Region_Code VARCHAR(MAX) NULL,
                                    Region VARCHAR(100) DEFAULT 'Nan',
                                    Happiness_Score DECIMAL(6,3),
                                    Regional_Happiness_Rank INT DEFAULT NULL,
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Trust DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Dystopia DECIMAL(7,5)
                                    )
```

```
/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.WHR_Americas_2016". */

TRUNCATE TABLE dbo.WHR_Americas_2016

/* - Inserting data into "dbo.WHR_Americas_2016" from "dbo.Raw_WHR_Americas_2016". */

INSERT INTO dbo.WHR_Americas_2016

SELECT Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,

            RANK() OVER(

                        ORDER BY Happiness_Score DESC
                        ) AS Regional_Happiness_Rank,

            GDP_per_Capita, Family, Health, Freedom, Trust, Generosity, Dystopia

FROM dbo.Raw_WHR_Americas_2016  /* 26 rows inserted */

/* List all the data in table "dbo.WHR_Americas_2016". */

SELECT *
FROM dbo.WHR_Americas_2016
```



Results Messages

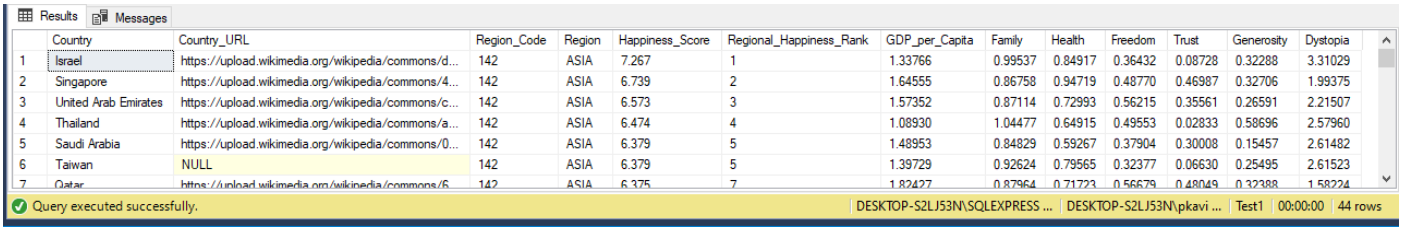
	Country	Country_URL	Region_Code	Region	Happiness_Score	Regional_Happiness_Rank	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Canada	https://upload.wikimedia.org/wikipedia/en/c/cf/Fl...	19	AMERICAS	7.404	1	1.44015	1.09610	0.82760	0.57370	0.31329	0.44834	2.70485
2	United States	https://upload.wikimedia.org/wikipedia/en/a/a4/Fl...	19	AMERICAS	7.104	2	1.50796	1.04782	0.77900	0.48163	0.14868	0.41077	2.72782
3	Costa Rica	https://upload.wikimedia.org/wikipedia/commons/f...	19	AMERICAS	7.087	3	1.06879	1.02152	0.76146	0.55225	0.10547	0.22553	3.35168
4	Puerto Rico	NULL	19	AMERICAS	7.039	4	1.35943	1.08113	0.77758	0.46823	0.12275	0.22202	3.00760
5	Brazil	https://upload.wikimedia.org/wikipedia/en/0/05/Fl...	19	AMERICAS	6.952	5	1.08754	1.03938	0.61415	0.40425	0.14166	0.15776	3.50733
6	Mexico	https://upload.wikimedia.org/wikipedia/commons/f...	19	AMERICAS	6.778	6	1.11508	0.71460	0.71143	0.37709	0.18355	0.11735	3.55906
7	Chile	https://upload.wikimedia.org/wikipedia/commons/...	19	AMERICAS	6.705	7	1.21670	0.90587	0.81883	0.37789	0.11451	0.31595	2.95505

Query executed successfully. DESKTOP-S2LJ53N\SQLEXPRESS ... DESKTOP-S2LJ53N\pkavi ... Test1 00:00:00 26 rows

The process in steps 3- 5, is repeated with tables “dbo.Raw_WHR_Asia_2016”, “dbo.Raw_WHR_Europe_2016”, “dbo.Raw_WHR_Nan_2016” and “dbo.Raw_WHR_Oceania_2016” to create tables “dbo.WHR_Asia_2016”, “dbo.WHR_Europe_2016”, “dbo.WHR_Nan_2016” and “dbo.WHR_Oceania_2016”, as per screenshot below;

Note: Statement of codes is available in Appendix I.

Table: “dbo.WHR_Asia_2016”

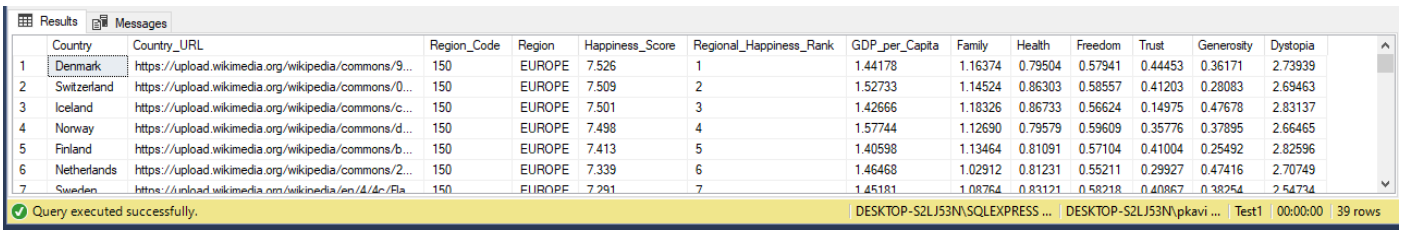


Results Messages

	Country	Country_URL	Region_Code	Region	Happiness_Score	Regional_Happiness_Rank	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Israel	https://upload.wikimedia.org/wikipedia/commons/d...	142	ASIA	7.267	1	1.33766	0.99537	0.84917	0.36432	0.08728	0.32288	3.31029
2	Singapore	https://upload.wikimedia.org/wikipedia/commons/4...	142	ASIA	6.739	2	1.64555	0.86758	0.94719	0.48770	0.46987	0.32706	1.99375
3	United Arab Emirates	https://upload.wikimedia.org/wikipedia/commons/c...	142	ASIA	6.573	3	1.57352	0.87114	0.72993	0.56215	0.35561	0.26591	2.21507
4	Thailand	https://upload.wikimedia.org/wikipedia/commons/a...	142	ASIA	6.474	4	1.08930	1.04477	0.64915	0.49553	0.02833	0.58696	2.57960
5	Saudi Arabia	https://upload.wikimedia.org/wikipedia/commons/0...	142	ASIA	6.379	5	1.48953	0.84829	0.59267	0.37904	0.30008	0.15457	2.61482
6	Taiwan	NULL	142	ASIA	6.379	5	1.39729	0.92624	0.79565	0.32377	0.06630	0.25495	2.61523
7	Qatar	https://upload.wikimedia.org/wikipedia/commons/6...	142	ASIA	6.375	7	1.82427	0.87964	0.71723	0.56679	0.48049	0.32388	1.58224

Query executed successfully. DESKTOP-S2LJ53N\SQLEXPRESS ... DESKTOP-S2LJ53N\pkavi ... Test1 00:00:00 44 rows

Table: : “dbo.WHR_Europe_2016”



Results Messages

	Country	Country_URL	Region_Code	Region	Happiness_Score	Regional_Happiness_Rank	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Denmark	https://upload.wikimedia.org/wikipedia/commons/9...	150	EUROPE	7.526	1	1.44178	1.16374	0.79504	0.57941	0.44453	0.36171	2.73939
2	Switzerland	https://upload.wikimedia.org/wikipedia/commons/0...	150	EUROPE	7.509	2	1.52733	1.14524	0.86303	0.58557	0.41203	0.28083	2.69463
3	Iceland	https://upload.wikimedia.org/wikipedia/commons/c...	150	EUROPE	7.501	3	1.42666	1.18326	0.86733	0.56624	0.14975	0.47678	2.83137
4	Norway	https://upload.wikimedia.org/wikipedia/commons/d...	150	EUROPE	7.498	4	1.57744	1.12690	0.79579	0.59609	0.35776	0.37895	2.66465
5	Finland	https://upload.wikimedia.org/wikipedia/commons/b...	150	EUROPE	7.413	5	1.40598	1.13464	0.81091	0.57104	0.41004	0.25492	2.82596
6	Netherlands	https://upload.wikimedia.org/wikipedia/commons/2...	150	EUROPE	7.339	6	1.46468	1.02912	0.81231	0.55211	0.29927	0.47416	2.70749
7	Sweden	https://upload.wikimedia.org/wikipedia/en/4/4c/Fla...	150	EUROPE	7.291	7	1.45181	1.08764	0.83121	0.58218	0.40867	0.38254	2.54734

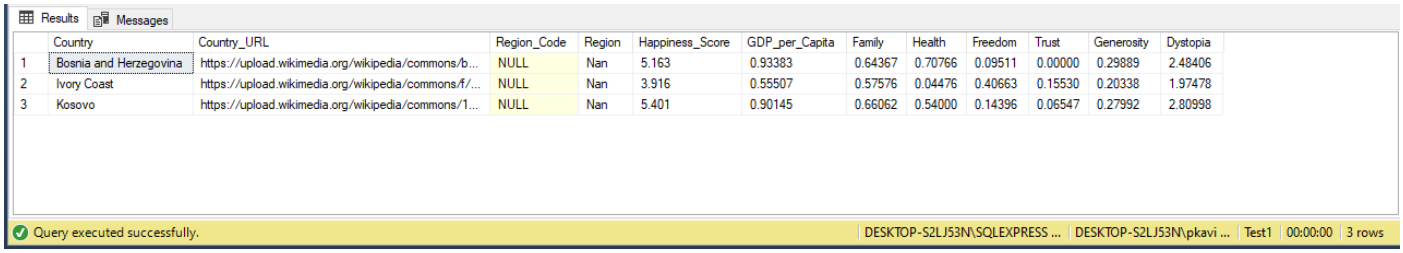
Query executed successfully. DESKTOP-S2LJ53N\SQLEXPRESS ... DESKTOP-S2LJ53N\pkavi ... Test1 00:00:00 39 rows

Step 6. 'Nan'

- Countries who's region are "Not Available" or Null are displayed in "Nan".

When table "dbo.Country" created, values in "Region" column was listed as "Nan" by default;

```
/* - Creating table for storing raw Country data */

CREATE TABLE dbo.Country(
                        Country VARCHAR(100) NULL,
                        Image_File VARCHAR(MAX) NULL,
                        Image_URL VARCHAR(MAX) NULL,
                        Alpha_2 VARCHAR(100) NULL,
                        Alpha_3 VARCHAR(100) NULL,
                        Country_Code VARCHAR(MAX) NULL,
                        iso_3166_2 VARCHAR(100) NULL,
                        Region VARCHAR(100) DEFAULT 'Nan',
                        Sub_Region VARCHAR(100) DEFAULT NULL,
                        Intermediate_Region VARCHAR(100) DEFAULT NULL,
                        Region_Code VARCHAR(MAX) NULL,
                        Sub_Region_Code VARCHAR(MAX) NULL,
                        Intermediate_Region_Code VARCHAR(MAX) NULL
                        )

/* Breaking "dbo.Raw_WHR_2016" into separate regions; namely "Africa", "Americas", "Asia", "Europe",
"Nan" and "Oceania"; using "Nan" as example. */

SELECT a.Country, a.Image_URL AS Country_URL, a.Region_Code, a.Region, b.Happiness_Score,
b.GDP_per_Capita, b.Family, b.Health, b.Freedom, b.Trust, b.Generosity, b.Dystopia
FROM dbo.Country a
INNER JOIN dbo.Raw_WHR_2016 b
ON a.Country=b.Country
WHERE a.Region='Nan'  /* 3 rows displayed */
```

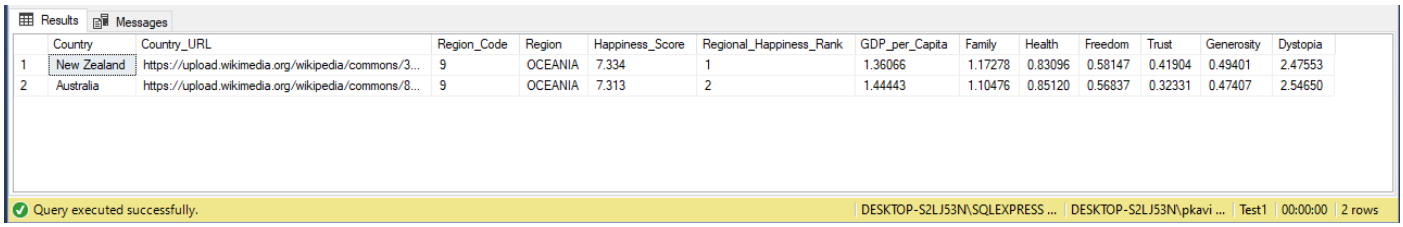


Results | Messages

	Country	Country_URL	Region_Code	Region	Happiness_Score	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	Bosnia and Herzegovina	https://upload.wikimedia.org/wikipedia/commons/b...	NULL	Nan	5.163	0.93383	0.64367	0.70766	0.09511	0.00000	0.29889	2.48406
2	Ivory Coast	https://upload.wikimedia.org/wikipedia/commons/f/...	NULL	Nan	3.916	0.55507	0.57576	0.04476	0.40663	0.15530	0.20338	1.97478
3	Kosovo	https://upload.wikimedia.org/wikipedia/commons/1...	NULL	Nan	5.401	0.90145	0.66062	0.54000	0.14396	0.06547	0.27992	2.80998

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 3 rows

Table: "dbo.WHR_Oceania_2016"



Results | Messages

	Country	Country_URL	Region_Code	Region	Happiness_Score	Regional_Happiness_Rank	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity	Dystopia
1	New Zealand	https://upload.wikimedia.org/wikipedia/commons/3...	9	OCEANIA	7.334	1	1.36066	1.17278	0.83096	0.58147	0.41904	0.49401	2.47553
2	Australia	https://upload.wikimedia.org/wikipedia/commons/8...	9	OCEANIA	7.313	2	1.44443	1.10476	0.85120	0.56837	0.32331	0.47407	2.54650

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 2 rows

Using tables "dbo.Raw_WHR_2017", "dbo.Raw_WHR_2018" and "dbo.Raw_WHR_2019", the enhancements as per steps 1 – 6 are implemented and the tables which were created as a result are displayed below with "Implemented Changes" described in the table below (codes / SQL statements available in APPENDIX I):

<Table below on next page.>

(Using report "HR_2016" and the derived tables as example, namely derived tables; dbo.WHR_Africa_2016, dbo.WHR_Americas_2016, dbo.WHR_Asia_2016, dbo.WHR_Europe_2016, dbo.WHR_Nan_2016 and dbo.WHR_Oceania_2016.)

Year	Table Name	Description	Derived Tables	Implemented Changes	Derived Table 1	Implemented Changes
2016	dbo.Raw_WHR_2016	Stores raw data from "HR_2016.csv".	dbo.newRaw_WHR_2016	Country's overall happiness rank and happiness status.	N/A	N/A
			dbo.Raw_WHR_Africa_2016	Divided into respective regions	dbo.WHR_Africa_2016	Includes Country_URL, Region_Code, Region column values displayed in uppercase and 'Nan', Regional_Happiness_Rank,
			dbo.Raw_WHR_Americas_2016		dbo.WHR_Americas_2016	
			dbo.Raw_WHR_Asia_2016		dbo.WHR_Asia_2016	
			dbo.Raw_WHR_Europe_2016		dbo.WHR_Europe_2016	
			dbo.Raw_WHR_Nan_2016		dbo.WHR_Nan_2016	
			dbo.Raw_WHR_Oceania_2016		dbo.WHR_Oceania_2016	
2017	dbo.Raw_WHR_2017	Stores raw data from "happiNess_report_2017.csv".	dbo.newRaw_WHR_2017	Country's overall happiness rank and happiness status.	N/A	N/A
			dbo.Raw_WHR_Africa_2017	Divided into respective regions	dbo.WHR_Africa_2017	Includes Country_URL, Region_Code, Region column values displayed in uppercase and 'Nan', Regional_Happiness_Rank,
			dbo.Raw_WHR_Americas_2017		dbo.WHR_Americas_2017	
			dbo.Raw_WHR_Asia_2017		dbo.WHR_Asia_2017	
			dbo.Raw_WHR_Europe_2017		dbo.WHR_Europe_2017	
			dbo.Raw_WHR_Nan_2017		dbo.WHR_Nan_2017	
			dbo.Raw_WHR_Oceania_2017		dbo.WHR_Oceania_2017	
2018	dbo.Raw_WHR_2018	Stores raw data from "2018.csv".	dbo.newRaw_WHR_2018	Country's overall happiness rank and happiness status.	N/A	N/A
			dbo.Raw_WHR_Africa_2018	Divided into respective regions	dbo.WHR_Africa_2018	Includes Country_URL, Region_Code, Region column values displayed in uppercase and 'Nan', Regional_Happiness_Rank,
			dbo.Raw_WHR_Americas_2018		dbo.WHR_Americas_2018	
			dbo.Raw_WHR_Asia_2018		dbo.WHR_Asia_2018	
			dbo.Raw_WHR_Europe_2018		dbo.WHR_Europe_2018	
			dbo.Raw_WHR_Nan_2018		dbo.WHR_Nan_2018	
			dbo.Raw_WHR_Oceania_2018		dbo.WHR_Oceania_2018	
2019	dbo.Raw_WHR_2019	Stores raw data from "report_2019.csv".	dbo.newRaw_WHR_2019	Country's overall happiness rank and happiness status.	N/A	N/A
			dbo.Raw_WHR_Africa_2019	Divided into respective regions	dbo.WHR_Africa_2019	Includes Country_URL, Region_Code, Region column values displayed in uppercase and 'Nan', Regional_Happiness_Rank,
			dbo.Raw_WHR_Americas_2019		dbo.WHR_Americas_2019	
			dbo.Raw_WHR_Asia_2019		dbo.WHR_Asia_2019	
			dbo.Raw_WHR_Europe_2019		dbo.WHR_Europe_2019	
			dbo.Raw_WHR_Nan_2019		dbo.WHR_Nan_2019	
			dbo.Raw_WHR_Oceania_2019		dbo.WHR_Oceania_2019	

- Extraction of tables "dbo.newRaw_WHR_2016", "dbo.newRaw_WHR_2017", "dbo.newRaw_WHR_2018" and "dbo.newRaw_WHR_2019".

Table Flow	
Table Name	**Extracted to ".csv" format for use with R STUDIO**
dbo.newRaw_WHR_2016	newWHR2016.csv
dbo.newRaw_WHR_2017	newWHR2017.csv
dbo.newRaw_WHR_2018	newWHR2018.csv
dbo.newRaw_WHR_2019	newWHR2019.csv

- Combination of tables "dbo.newRaw_WHR_2016", "dbo.newRaw_WHR_2017", "dbo.newRaw_WHR_2018" and "dbo.newRaw_WHR_2019" into dbo.WHR_Combined2".

Table Flow	
Table Name	**Combined into:**
dbo.newRaw_WHR_2016	"**dbo.WHR_Combined2**" for use with "Task 3" contains columns: Year, Country, Happiness_Score, Overall_Happiness_Rank, Happiness_Status, GDP_per_Capita, Family, Health, Freedom, Generosity, Trust
dbo.newRaw_WHR_2017	
dbo.newRaw_WHR_2018	
dbo.newRaw_WHR_2019	

- Combining all Regional data together from data frames "dbo.WHR_Africa_2016", "dbo.WHR_Americas_2016", "dbo.WHR_Asia_2016", "dbo.WHR_Europe_2016", "dbo.WHR_Nan_2016" and "dbo.WHR_Oceania_2016" into “dbo.WHR_Combined_2016”.

Table Flow	
Table Name	**Combined into:**
dbo.WHR_Africa_2016	"**dbo.WHR_Combined_2016**" for use with "Task 3" contains columns: Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score, Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS Perceptions_of_Corruption
dbo.WHR_Americas_2016	
dbo.WHR_Asia_2016	
dbo.WHR_Europe_2016	
dbo.WHR_Nan_2016	
dbo.WHR_Oceania_2016	

- Combining all Regional data together from data frames "dbo.WHR_Africa_2017", "dbo.WHR_Americas_2017", "dbo.WHR_Asia_2017", "dbo.WHR_Europe_2017", "dbo.WHR_Nan_2017" and "dbo.WHR_Oceania_2017" into “dbo.WHR_Combined_2017”.

Table Flow	
Table Name	**Combined into:**
dbo.WHR_Africa_2017	"**dbo.WHR_Combined_2017**" for use with "Task 3" contains columns: Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score, Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS Perceptions_of_Corruption
dbo.WHR_Americas_2017	
dbo.WHR_Asia_2017	
dbo.WHR_Europe_2017	
dbo.WHR_Nan_2017	
dbo.WHR_Oceania_2017	

- Combining all Regional data together from data frames "dbo.WHR_Africa_2018", "dbo.WHR_Americas_2018", "dbo.WHR_Asia_2018", "dbo.WHR_Europe_2018", "dbo.WHR_Nan_2018" and "dbo.WHR_Oceania_2018" into “dbo.WHR_Combined_2018”.

Table Flow	
Table Name	**Combined into:**
dbo.WHR_Africa_2018	"**dbo.WHR_Combined_2018**" for use with "Task 3" contains columns: Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score, Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS Perceptions_of_Corruption
dbo.WHR_Americas_2018	
dbo.WHR_Asia_2018	
dbo.WHR_Europe_2018	
dbo.WHR_Nan_2018	
dbo.WHR_Oceania_2018	

- Combining all Regional data together from data frames "dbo.WHR_Africa_2019", "dbo.WHR_Americas_2019", "dbo.WHR_Asia_2019", "dbo.WHR_Europe_2019", "dbo.WHR_Nan_2019" and "dbo.WHR_Oceania_2019" into “dbo.WHR_Combined_2019”.

Table Flow	
Table Name	**Combined into:**
dbo.WHR_Africa_2019	"**dbo.WHR_Combined_2019**" for use with "Task 3" contains columns: Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score, Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS Perceptions_of_Corruption
dbo.WHR_Americas_2019	
dbo.WHR_Asia_2019	
dbo.WHR_Europe_2019	
dbo.WHR_Nan_2019	
dbo.WHR_Oceania_2019	

- Combining tables "dbo.WHR_Combined_2016", "dbo.WHR_Combined_2017", "dbo.WHR_Combined_2018" and "dbo.WHR_Combined_2019" together in table "dbo.WHR_CombinedList".

Table Flow	
Table Name	**Combined into:**
dbo.newRaw_WHR_2016	"**dbo.WHR_CombinedList**" for use with "Task 3" contains columns: Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score, Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS Perceptions_of_Corruption
dbo.newRaw_WHR_2017	
dbo.newRaw_WHR_2018	
dbo.newRaw_WHR_2019	

Step 7. Year

Derived tables “dbo.WHR_Africa_2016”, “dbo.WHR_Americas_2016”, “dbo.WHR_Asia_2016”, “dbo.WHR_Europe_2016”, “dbo.WHR_Nan_2016” and “dbo.WHR_Oceania_2016” are combined together and the year of the report is inserted in order to display “Year” of report in table “dbo.WHR_Combined_2016”.

```
--------- 2016 ---------

/* Creating table "dbo.WHR_Combined_2016" to store data pertaining to countries from the tables
"dbo.WHR_Africa_2016", "dbo.WHR_Americas_2016", "dbo.WHR_Asia_2016", "dbo.WHR_Europe_2016",
"dbo.WHR_Nan_2016" and "dbo.WHR_Oceania_2016". */

CREATE TABLE dbo.WHR_Combined_2016(
                                    Year INT DEFAULT NULL,
                                    Country VARCHAR(100) NULL,
                                    Country_URL VARCHAR(MAX) NULL,
                                    Region_Code VARCHAR(MAX) NULL,
                                    Region VARCHAR(100) DEFAULT 'Nan',
                                    Happiness_Score DECIMAL(6,3),
                                    Regional_Happiness_Rank INT DEFAULT NULL,
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Trust DECIMAL(7,5)
                                    )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.WHR_Combined_2016". */

TRUNCATE TABLE dbo.WHR_Combined_2016

/* - Inserting data into "dbo.WHR_Combined_2016". */

INSERT INTO dbo.WHR_Combined_2016

SELECT 2016 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Africa_2016

UNION

SELECT 2016 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Americas_2016

UNION

SELECT 2016 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Asia_2016

UNION
```

```
SELECT 2016 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Europe_2016

UNION

SELECT 2016 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Nan_2016

UNION

SELECT 2016 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Oceania_2016

/* List all the data in table "dbo.WHR_Combined_2016". */

SELECT *
FROM dbo.WHR_Combined_2016  /* 153 rows displayed */
```

Results | Messages

	Year	Country	Country_URL	Region_Code	Region	Happiness_Score	Rank_per_Region	GDP_per_Capita	Family	Healthy_Life_Expectancy	Freedom_to_make_Life_Choices	Gener
1	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	3.360	43	0.38227	0.11037	0.17344	0.16430	0.312
2	2016	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	4.655	37	0.95530	0.50163	0.73007	0.31866	0.168
3	2016	Algeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	6.355	1	1.05266	0.83309	0.61804	0.21006	0.070
4	2016	Angola	https://upload.wikimedia.org/wikipedia/commons/9...	2	AFRICA	3.866	26	0.84731	0.66366	0.04991	0.00589	0.120
5	2016	Argentina	https://upload.wikimedia.org/wikipedia/commons/1...	19	AMERICAS	6.650	9	1.15137	1.06612	0.69711	0.42284	0.109
6	2016	Armenia	https://upload.wikimedia.org/wikipedia/commons/2...	142	ASIA	4.360	39	0.86086	0.62477	0.64083	0.14037	0.077
7	2016	Australia	https://upload.wikimedia.org/wikipedia/commons/8...	9	OCEANIA	7.313	2	1.44443	1.10476	0.85120	0.56837	0.474
8	2016	Austria	https://upload.wikimedia.org/wikipedia/commons/4...	150	EUROPE	7.119	8	1.45038	1.08383	0.80565	0.54355	0.328

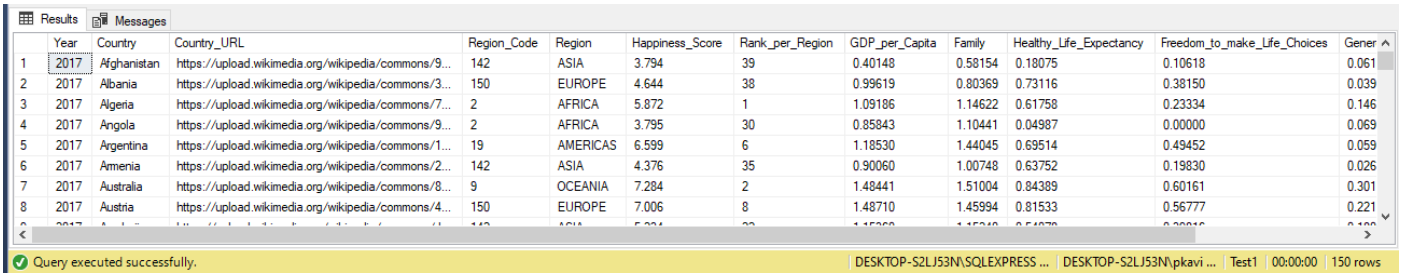
Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 153 rows

Similarly, step 7 is applied on derived tables “dbo.WHR_Africa_2017”, “dbo.WHR_Americas_2017”, “dbo.WHR_Asia_2017”, “dbo.WHR_Europe_2017”, “dbo.WHR_Nan_2017” and “dbo.WHR_Oceania_2017” to form the combined table “dbo.WHR_Combined_2017” and applied field “Year” as below;

```
--------- 2017 ---------

/* Creating table "dbo.WHR_Combined_2017" to store data pertaining to countries from the tables
"dbo.WHR_Africa_2017", "dbo.WHR_Americas_2017", "dbo.WHR_Asia_2017", "dbo.WHR_Europe_2017",
"dbo.WHR_Nan_2017" and "dbo.WHR_Oceania_2017". */

CREATE TABLE dbo.WHR_Combined_2017(
                                    Year INT DEFAULT NULL,
                                    Country VARCHAR(100) NULL,
                                    Country_URL VARCHAR(MAX) NULL,
                                    Region_Code VARCHAR(MAX) NULL,
                                    Region VARCHAR(100) DEFAULT 'Nan',
                                    Happiness_Score DECIMAL(6,3),
                                    Regional_Happiness_Rank INT DEFAULT NULL,
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Trust DECIMAL(7,5)
                                    )
```

```
/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.WHR_Combined_2017". */

TRUNCATE TABLE dbo.WHR_Combined_2017

/* - Inserting data into "dbo.WHR_Combined_2017". */

INSERT INTO dbo.WHR_Combined_2017

SELECT 2017 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Africa_2017

UNION

SELECT 2017 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Americas_2017

UNION

SELECT 2017 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Asia_2017

UNION

SELECT 2017 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Europe_2017

UNION

SELECT 2017 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Nan_2017

UNION

SELECT 2017 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Oceania_2017
```

```
/* List all the data in table "dbo.WHR_Combined_2017". */

SELECT *
FROM dbo.WHR_Combined_2017  /* 150 rows displayed */
```



Results | Messages

	Year	Country	Country_URL	Region_Code	Region	Happiness_Score	Rank_per_Region	GDP_per_Capita	Family	Healthy_Life_Expectancy	Freedom_to_make_Life_Choices	Gener
1	2017	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	3.794	39	0.40148	0.58154	0.18075	0.10618	0.061
2	2017	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	4.644	38	0.99619	0.80369	0.73116	0.38150	0.039
3	2017	Algeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	5.872	1	1.09186	1.14622	0.61758	0.23334	0.146
4	2017	Angola	https://upload.wikimedia.org/wikipedia/commons/9...	2	AFRICA	3.795	30	0.85843	1.10441	0.04987	0.00000	0.069
5	2017	Argentina	https://upload.wikimedia.org/wikipedia/commons/1...	19	AMERICAS	6.599	6	1.18530	1.44045	0.69514	0.49452	0.059
6	2017	Armenia	https://upload.wikimedia.org/wikipedia/commons/2...	142	ASIA	4.376	35	0.90060	1.00748	0.63752	0.19830	0.026
7	2017	Australia	https://upload.wikimedia.org/wikipedia/commons/8...	9	OCEANIA	7.284	2	1.48441	1.51004	0.84389	0.60161	0.301
8	2017	Austria	https://upload.wikimedia.org/wikipedia/commons/4...	150	EUROPE	7.006	8	1.48710	1.45994	0.81533	0.56777	0.221

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 150 rows

Similarly, step 7 is applied on derived tables “dbo.WHR_Africa_2018”, “dbo.WHR_Americas_2018”, “dbo.WHR_Asia_2018”, “dbo.WHR_Europe_2018”, “dbo.WHR_Nan_2018” and “dbo.WHR_Oceania_2018” to form the combined table “dbo.WHR_Combined_2018” and applied field “Year” as below;

```
--------- 2018 ---------

/* Creating table "dbo.WHR_Combined_2018" to store data pertaining to countries from the tables
"dbo.WHR_Africa_2018", "dbo.WHR_Americas_2018", "dbo.WHR_Asia_2018", "dbo.WHR_Europe_2018",
"dbo.WHR_Nan_2018" and "dbo.WHR_Oceania_2018". */

CREATE TABLE dbo.WHR_Combined_2018(
                                    Year INT DEFAULT NULL,
                                    Country VARCHAR(100) NULL,
                                    Country_URL VARCHAR(MAX) NULL,
                                    Region_Code VARCHAR(MAX) NULL,
                                    Region VARCHAR(100) DEFAULT 'Nan',
                                    Happiness_Score DECIMAL(6,3),
                                    Regional_Happiness_Rank INT DEFAULT NULL,
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Trust DECIMAL(7,5)
                                    )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.WHR_Combined_2018". */

TRUNCATE TABLE dbo.WHR_Combined_2018

/* - Inserting data into "dbo.WHR_Combined_2018". */

INSERT INTO dbo.WHR_Combined_2018

SELECT 2018 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Africa_2018

UNION

SELECT 2018 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Americas_2018

UNION
```

```
SELECT 2018 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Asia_2018

UNION

SELECT 2018 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Europe_2018

UNION

SELECT 2018 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Nan_2018

UNION

SELECT 2018 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Oceania_2018

/* List all the data in table "dbo.WHR_Combined_2018". */

SELECT *
FROM dbo.WHR_Combined_2018  /* 152 rows displayed. */
```

Results | Messages

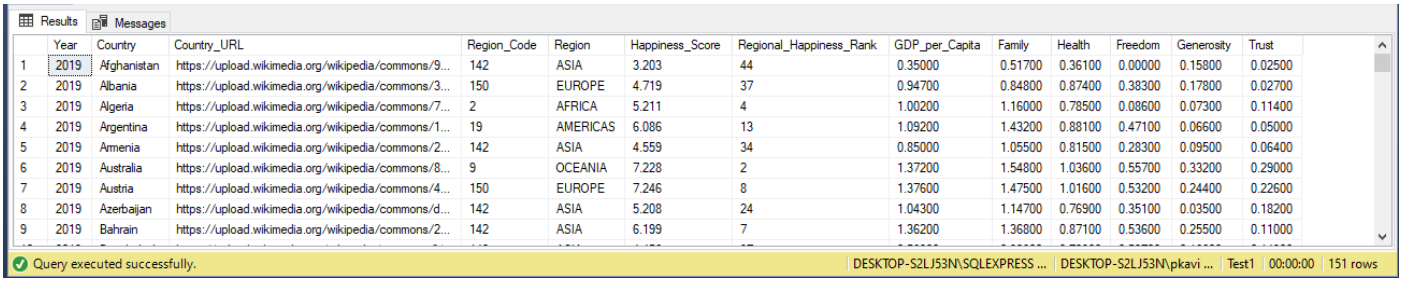
	Year	Country	Country_URL	Region_Code	Region	Happiness_Score	Rank_per_Region	GDP_per_Capita	Family	Healthy_Life_Expectancy	Freedom_to_make_Life_Choices	Gener
1	2018	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	3.632	42	0.33200	0.53700	0.25500	0.08500	0.191
2	2018	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	4.586	38	0.91600	0.81700	0.79000	0.41900	0.149
3	2018	Algeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	5.295	3	0.97900	1.15400	0.68700	0.07700	0.055
4	2018	Angola	https://upload.wikimedia.org/wikipedia/commons/9...	2	AFRICA	3.795	31	0.73000	1.12500	0.26900	0.00000	0.079
5	2018	Argentina	https://upload.wikimedia.org/wikipedia/commons/1...	19	AMERICAS	6.388	8	1.07300	1.46800	0.74400	0.57000	0.062
6	2018	Armenia	https://upload.wikimedia.org/wikipedia/commons/2...	142	ASIA	4.321	39	0.81600	0.99000	0.66600	0.26000	0.077
7	2018	Australia	https://upload.wikimedia.org/wikipedia/commons/8...	9	OCEANIA	7.272	2	1.34000	1.57300	0.91000	0.64700	0.361
8	2018	Austria	https://upload.wikimedia.org/wikipedia/commons/4...	150	EUROPE	7.139	9	1.34100	1.50400	0.89100	0.61700	0.242

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 152 rows

Similarly, step 7 is applied on derived tables “dbo.WHR_Africa_2019”, “dbo.WHR_Americas_201”, “dbo.WHR_Asia_2019”, “dbo.WHR_Europe_2019”, “dbo.WHR_Nan_2019” and “dbo.WHR_Oceania_2019” to form the combined table “dbo.WHR_Combined_2019” and applied field “Year” as below;

```
--------- 2019 ---------

/* Creating table "dbo.WHR_Combined_2019" to store data pertaining to countries from the tables
"dbo.WHR_Africa_2019", "dbo.WHR_Americas_2019", "dbo.WHR_Asia_2019", "dbo.WHR_Europe_2019",
"dbo.WHR_Nan_2019" and "dbo.WHR_Oceania_2019". */

CREATE TABLE dbo.WHR_Combined_2019(
                                    Year INT DEFAULT NULL,
                                    Country VARCHAR(100) NULL,
                                    Country_URL VARCHAR(MAX) NULL,
                                    Region_Code VARCHAR(MAX) NULL,
                                    Region VARCHAR(100) DEFAULT 'Nan',
                                    Happiness_Score DECIMAL(6,3),
                                    Regional_Happiness_Rank INT DEFAULT NULL,
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Trust DECIMAL(7,5)
                                    )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.WHR_Combined_2019". */

TRUNCATE TABLE dbo.WHR_Combined_2019

/* - Inserting data into "dbo.WHR_Combined_2019". */

INSERT INTO dbo.WHR_Combined_2019

SELECT 2019 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Africa_2019

UNION

SELECT 2019 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Americas_2019

UNION

SELECT 2019 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Asia_2019

UNION

SELECT 2019 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Europe_2019

UNION
```

```
SELECT 2019 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Nan_2019

UNION

SELECT 2019 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Oceania_2019

/* List all the data in table "dbo.WHR_Combined_2019". */

SELECT *
FROM dbo.WHR_Combined_2019  /* 151 rows displayed. */
```



Results | Messages

	Year	Country	Country_URL	Region_Code	Region	Happiness_Score	Regional_Happiness_Rank	GDP_per_Capita	Family	Health	Freedom	Generosity	Trust
1	2019	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	3.203	44	0.35000	0.51700	0.36100	0.00000	0.15800	0.02500
2	2019	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	4.719	37	0.94700	0.84800	0.87400	0.38300	0.17800	0.02700
3	2019	Algeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	5.211	4	1.00200	1.16000	0.78500	0.08600	0.07300	0.11400
4	2019	Argentina	https://upload.wikimedia.org/wikipedia/commons/1...	19	AMERICAS	6.086	13	1.09200	1.43200	0.88100	0.47100	0.06600	0.05000
5	2019	Armenia	https://upload.wikimedia.org/wikipedia/commons/2...	142	ASIA	4.559	34	0.85000	1.05500	0.81500	0.28300	0.09500	0.06400
6	2019	Australia	https://upload.wikimedia.org/wikipedia/commons/8...	9	OCEANIA	7.228	2	1.37200	1.54800	1.03600	0.55700	0.33200	0.29000
7	2019	Austria	https://upload.wikimedia.org/wikipedia/commons/4...	150	EUROPE	7.246	8	1.37600	1.47500	1.01600	0.53200	0.24400	0.22600
8	2019	Azerbaijan	https://upload.wikimedia.org/wikipedia/commons/d...	142	ASIA	5.208	24	1.04300	1.14700	0.76900	0.35100	0.03500	0.18200
9	2019	Bahrain	https://upload.wikimedia.org/wikipedia/commons/2...	142	ASIA	6.199	7	1.36200	1.36800	0.87100	0.53600	0.25500	0.11000

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 151 rows

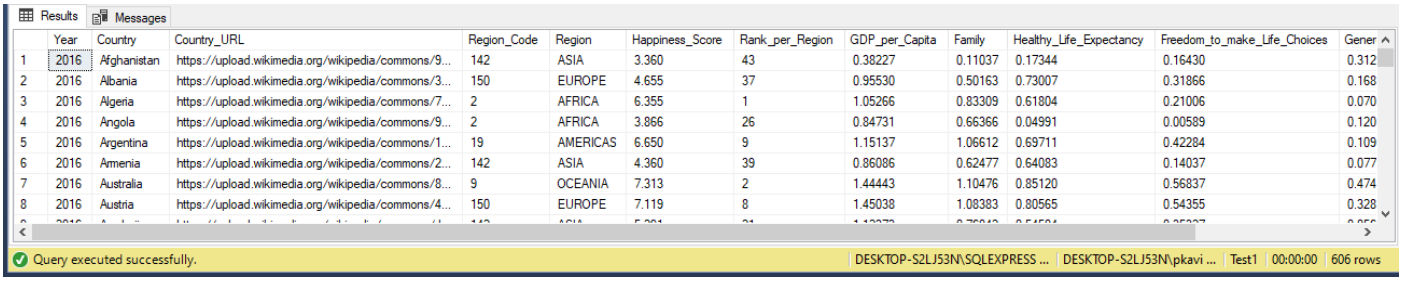
Step 8. Combining Yearly data frames into one data frame.

8.1 Combining from Regional data frames.

So as to be able to get the form as per requirement in “Task 3”, tables “dbo.WHR_Combined_2016”, “dbo.WHR_Combined_2017”, “dbo.WHR_Combined_2018” and “dbo.WHR_Combined_2019” is combined together to form table “dbo.WHR_CombinedList”.

```
/* - Table dbo.WHR_CombinedList contains data from the combined tables "dbo.WHR_Combined_2016",
"dbo.WHR_Combined_2017", "dbo.WHR_Combined_2018" and "dbo.WHR_Combined_2019" as per requirements of
Task 3. */

CREATE TABLE dbo.WHR_CombinedList(
                                  Year INT DEFAULT NULL,
                                  Country VARCHAR(100) NULL,
                                  Country_URL VARCHAR(MAX) NULL,
                                  Region_Code VARCHAR(MAX) NULL,
                                  Region VARCHAR(100) DEFAULT 'Nan',
                                  Happiness_Score DECIMAL(6,3),
                                  Regional_Happiness_Rank INT DEFAULT NULL,
                                  GDP_per_Capita DECIMAL(7,5),
                                  Family DECIMAL(7,5),
                                  Health DECIMAL(7,5),
                                  Freedom DECIMAL(7,5),
                                  Generosity DECIMAL(7,5),
                                  Trust DECIMAL(7,5)
                                  )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.WHR_CombinedList". */

TRUNCATE TABLE dbo.WHR_CombinedList
```

```
/* - Inserting data into "dbo.WHR_CombinedList". */
INSERT INTO dbo.WHR_CombinedList

SELECT 2016 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Combined_2016

UNION

SELECT 2017 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Combined_2017

UNION

SELECT 2018 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family AS Social_Support, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Combined_2018

UNION

SELECT 2019 AS Year, Country, Country_URL, Region_Code, UPPER(Region) AS Region, Happiness_Score,
Regional_Happiness_Rank AS Rank_per_Region, GDP_per_Capita, Family AS Social_Support, Health AS
Healthy_Life_Expectancy, Freedom AS Freedom_to_make_Life_Choices, Generosity, Trust AS
Perceptions_of_Corruption
/* Converting "Trust" column to VARCHAR(20) using CAST() - CAST([Trust] AS VARCHAR(20)) */

FROM dbo.WHR_Combined_2019

/* List all the data in table "dbo.WHR_CombinedList". */

SELECT *
FROM dbo.WHR_CombinedList  /* 606 rows displayed. */
```



Results | Messages

	Year	Country	Country_URL	Region_Code	Region	Happiness_Score	Rank_per_Region	GDP_per_Capita	Family	Healthy_Life_Expectancy	Freedom_to_make_Life_Choices	Gener
1	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	3.360	43	0.38227	0.11037	0.17344	0.16430	0.312
2	2016	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	4.655	37	0.95530	0.50163	0.73007	0.31866	0.168
3	2016	Algeria	https://upload.wikimedia.org/wikipedia/commons/7...	2	AFRICA	6.355	1	1.05266	0.83309	0.61804	0.21006	0.070
4	2016	Angola	https://upload.wikimedia.org/wikipedia/commons/9...	2	AFRICA	3.866	26	0.84731	0.66366	0.04991	0.00589	0.120
5	2016	Argentina	https://upload.wikimedia.org/wikipedia/commons/1...	19	AMERICAS	6.650	9	1.15137	1.06612	0.69711	0.42284	0.109
6	2016	Armenia	https://upload.wikimedia.org/wikipedia/commons/2...	142	ASIA	4.360	39	0.86086	0.62477	0.64083	0.14037	0.077
7	2016	Australia	https://upload.wikimedia.org/wikipedia/commons/8...	9	OCEANIA	7.313	2	1.44443	1.10476	0.85120	0.56837	0.474
8	2016	Austria	https://upload.wikimedia.org/wikipedia/commons/4...	150	EUROPE	7.119	8	1.45038	1.08383	0.80565	0.54355	0.328

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 606 rows

8.2 Combining from Yearly data frames.

In order to be able to get the “Overall_Happiness_Rank” and “Happiness_Status”, tables “dbo.newRaw_WHR_2016”, “dbo.newRaw_WHR_2017”, “dbo.newRaw_WHR_2018” and “dbo.newRaw_WHR_2019” are combined together to form table “dbo.WHR_Combined2”.

```sql
/* - Creating table "dbo.WHR_CombinedList" which will hold combined data from table
"dbo.newRaw_WHR_2016", "dbo.newRaw_WHR_2017", "dbo.newRaw_WHR_2018" and "dbo.newRaw_WHR_2019". */

CREATE TABLE dbo.WHR_Combined2(
                                    Year INT DEFAULT NULL,
                                    Country VARCHAR(100) NULL,
                                    Happiness_Score DECIMAL(6,3),
                                    Overall_Happiness_Rank INT DEFAULT NULL,
                                    Happiness_Status VARCHAR(20) NULL,
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Trust DECIMAL(7,5)
                                    )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.WHR_Combined2". */

TRUNCATE TABLE dbo.WHR_Combined2

/* - Inserting data in dbo.WHR_Combined2. */

INSERT INTO dbo.WHR_Combined2

SELECT Year, Country, Happiness_Score, Overall_Happiness_Rank, Happiness_Status, GDP_per_Capita,
Family, Health, Freedom, Generosity, Trust

FROM dbo.newRaw_WHR_2016

UNION

SELECT Year, Country, Happiness_Score, Overall_Happiness_Rank, Happiness_Status, GDP_per_Capita,
Family, Health, Freedom, Generosity, Trust

FROM dbo.newRaw_WHR_2017

UNION

SELECT Year, Country, Happiness_Score, Overall_Happiness_Rank, Happiness_Status, GDP_per_Capita,
Family, Health, Freedom, Generosity, Trust

FROM dbo.newRaw_WHR_2018

UNION

SELECT Year, Country, Happiness_Score, Overall_Happiness_Rank, Happiness_Status, GDP_per_Capita,
Family, Health, Freedom, Generosity, Trust

FROM dbo.newRaw_WHR_2019

/* List all the data in table "dbo.WHR_Combined2". */

SELECT *
FROM dbo.WHR_Combined2  /* 624 rows displayed. */
```

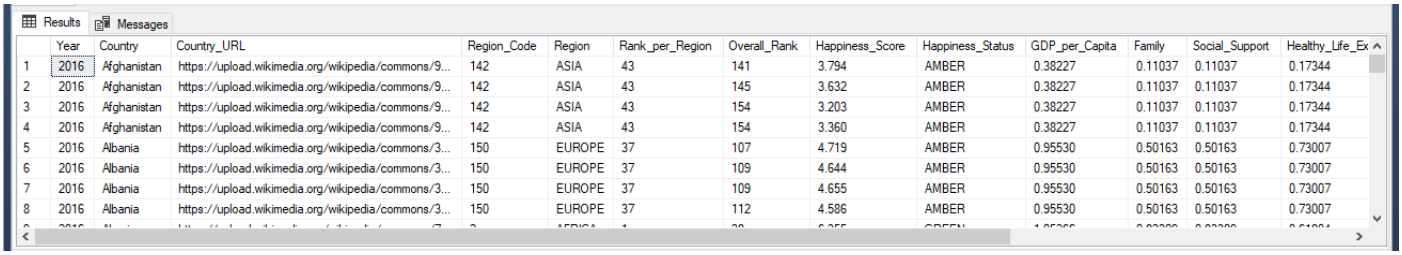
Results | Messages

	Year	Country	Happiness_Score	Overall_Happiness_Rank	Happiness_Status	GDP_per_Capita	Family	Health	Freedom	Generosity	Trust
1	2016	Afghanistan	3.360	154	AMBER	0.38227	0.11037	0.17344	0.16430	0.31268	0.07112
2	2016	Albania	4.655	109	AMBER	0.95530	0.50163	0.73007	0.31866	0.16840	0.05301
3	2016	Algeria	6.355	38	GREEN	1.05266	0.83309	0.61804	0.21006	0.07044	0.16157
4	2016	Angola	3.866	141	AMBER	0.84731	0.66366	0.04991	0.00589	0.12071	0.08434
5	2016	Argentina	6.650	26	GREEN	1.15137	1.06612	0.69711	0.42284	0.10989	0.07296
6	2016	Armenia	4.360	121	AMBER	0.86086	0.62477	0.64083	0.14037	0.07793	0.03616
7	2016	Australia	7.313	9	GREEN	1.44443	1.10476	0.85120	0.56837	0.47407	0.32331
8	2016	Austria	7.119	12	GREEN	1.45038	1.08383	0.80565	0.54355	0.32865	0.21348
9	2016	Azerbaijan	5.291	81	AMBER	1.12373	0.76042	0.54504	0.35327	0.05640	0.17914

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 624 rows

Step 9. Merging table “dbo.WHR_CombinedList” and “dbo.WHR_Combined2” using INNER JOINS.

```
----- Task 3. -Verifying the statement. ----
SELECT DISTINCT a.Year, a.Country, a.Country_URL, a.Region_Code, a.Region,
(a.Regional_Happiness_Rank) AS Rank_per_Region, (b.Overall_Happiness_Rank) AS Overall_Rank,
b.Happiness_Score, b.Happiness_Status, a.GDP_per_Capita, a.Family, (a.Family) AS Social_Support,
(a.Health) AS Healthy_Life_Expectancy, (a.Freedom) AS Freedom_to_make_life_choices,
a.Generosity, (a.Trust) AS Perceptions_of_Corruption

FROM dbo.WHR_CombinedList a
INNER JOIN dbo.WHR_Combined2 b
ON a.Country=b.Country
```

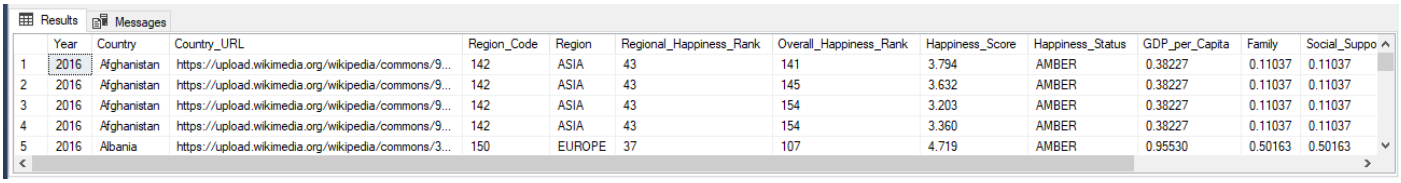


Results Messages

	Year	Country	Country_URL	Region_Code	Region	Rank_per_Region	Overall_Rank	Happiness_Score	Happiness_Status	GDP_per_Capita	Family	Social_Support	Healthy_Life_Ex
1	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	43	141	3.794	AMBER	0.38227	0.11037	0.11037	0.17344
2	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	43	145	3.632	AMBER	0.38227	0.11037	0.11037	0.17344
3	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	43	154	3.203	AMBER	0.38227	0.11037	0.11037	0.17344
4	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	43	154	3.360	AMBER	0.38227	0.11037	0.11037	0.17344
5	2016	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	37	107	4.719	AMBER	0.95530	0.50163	0.50163	0.73007
6	2016	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	37	109	4.644	AMBER	0.95530	0.50163	0.50163	0.73007
7	2016	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	37	109	4.655	AMBER	0.95530	0.50163	0.50163	0.73007
8	2016	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	37	112	4.586	AMBER	0.95530	0.50163	0.50163	0.73007

Creating table “dbo.Task_3” to create the form as requested in “Task 3”.

```
/* - Creating table "dbo.Task_3" to return values as per form. */

CREATE TABLE dbo.Task_3(
                        Year INT DEFAULT NULL,
                        Country VARCHAR(100) NULL,
                        Country_URL VARCHAR(MAX) NULL,
                        Region_Code VARCHAR(MAX) NULL,
                        Region VARCHAR(100) DEFAULT 'Nan',
                        Regional_Happiness_Rank INT DEFAULT NULL,
                        Overall_Happiness_Rank INT DEFAULT NULL,
                        Happiness_Score DECIMAL(6,3),
                        Happiness_Status VARCHAR(20) NULL,
                        GDP_per_Capita DECIMAL(7,5),
                        Family DECIMAL(7,5),
                        Social_Support DECIMAL(7,5) DEFAULT NULL,
                        Health DECIMAL(7,5),
                        Freedom DECIMAL(7,5),
                        Generosity DECIMAL(7,5),
                        Trust DECIMAL(7,5)
                        )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.Task_3". */

TRUNCATE TABLE dbo.Task_3

/* - Inserting data in dbo.Task_3. */

INSERT INTO dbo.Task_3

SELECT DISTINCT a.Year, a.Country, a.Country_URL, a.Region_Code, a.Region,
(a.Regional_Happiness_Rank) AS Rank_per_Region, (b.Overall_Happiness_Rank) AS Overall_Rank,
b.Happiness_Score, b.Happiness_Status, a.GDP_per_Capita, a.Family, (a.Family) AS Social_Support,
(a.Health) AS Healthy_Life_Expectancy, (a.Freedom) AS Freedom_to_make_life_choices,
a.Generosity, (a.Trust) AS Perceptions_of_Corruption

FROM dbo.WHR_CombinedList a
INNER JOIN dbo.WHR_Combined2 b
ON a.Country=b.Country

/* List all the data in table "dbo.WHR_Combined2". */

SELECT *
FROM dbo.Task_3
```



Results | Messages

	Year	Country	Country_URL	Region_Code	Region	Regional_Happiness_Rank	Overall_Happiness_Rank	Happiness_Score	Happiness_Status	GDP_per_Capita	Family	Social_Suppo
1	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	43	141	3.794	AMBER	0.38227	0.11037	0.11037
2	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	43	145	3.632	AMBER	0.38227	0.11037	0.11037
3	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	43	154	3.203	AMBER	0.38227	0.11037	0.11037
4	2016	Afghanistan	https://upload.wikimedia.org/wikipedia/commons/9...	142	ASIA	43	154	3.360	AMBER	0.38227	0.11037	0.11037
5	2016	Albania	https://upload.wikimedia.org/wikipedia/commons/3...	150	EUROPE	37	107	4.719	AMBER	0.95530	0.50163	0.50163

Task 4.

An application in BCM wants to consume an extract in JSON format containing the details as per table below. Complement the existing data pipeline in step 2 to return the required information.

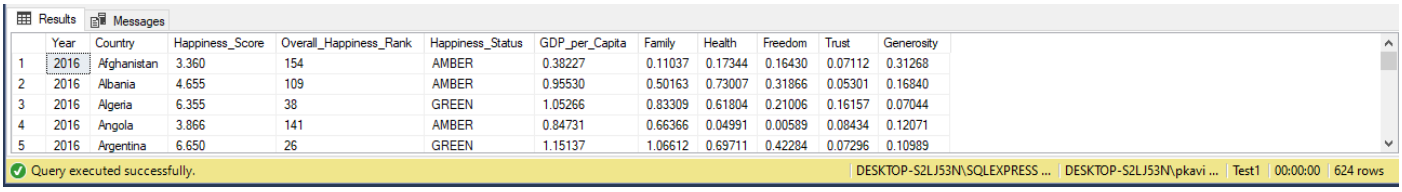
Column Name	Specifications
Country	
Highest Rank	Highest rank for all the years
Lowest Rank	Lowest rank for all the years
Highest Happiness Score	Highest Score for all the years
Lowest Happiness Score	Lowest Score for all the years

Since we are looking for the “Highest Rank”, “Lowest Rank”, “Highest Happiness Score” and “Lowest Happiness Score” for all the years, we need to have a combined list, which holds data for all the years.

Table: “dbo.WHR_Combined2”

```
/* - Creating table "dbo.WHR_Combined2" to store data for identifying Highest Rank, Lowest Rank,
Highest Happiness Score, Lowest Happiness Score per Country for all the years. */

CREATE TABLE dbo.WHR_Combined2(
                        Year INT DEFAULT NULL,
                        Country VARCHAR(100) NULL,
                        Happiness_Score DECIMAL(6,3),
                        Overall_Happiness_Rank INT DEFAULT NULL,
                        Happiness_Status VARCHAR(20) NULL,
                        GDP_per_Capita DECIMAL(7,5),
                        Family DECIMAL(7,5),
                        Health DECIMAL(7,5),
                        Freedom DECIMAL(7,5),
                        Trust DECIMAL(7,5),
                        Generosity DECIMAL(7,5)
                        )

/* Using "TRUNCATE" to remove all rows (data) from a table "dbo.WHR_Combined2". */

TRUNCATE TABLE dbo.WHR_Combined2

/* Inserting data in table "dbo.WHR_Combined2". */

INSERT INTO dbo.WHR_Combined2

      SELECT 2016 AS Year, Country, Happiness_Score, Overall_Happiness_Rank, Happiness_Status,
      GDP_per_Capita, Family, Health, Freedom, CAST([Trust] AS VARCHAR(20)) AS Trust, Generosity

      FROM dbo.newRaw_WHR_2016

UNION

      SELECT 2017 AS Year, Country, Happiness_Score, Overall_Happiness_Rank, Happiness_Status,
      GDP_per_Capita, Family, Health, Freedom, CAST([Trust] AS VARCHAR(20)) AS Trust, Generosity

      FROM dbo.newRaw_WHR_2017

UNION
```

```
	SELECT 2018 AS Year, Country, Happiness_Score, Overall_Happiness_Rank, Happiness_Status,
	GDP_per_Capita, Family, Health, Freedom, CAST([Trust] AS VARCHAR(20)) AS Trust, Generosity

	FROM dbo.newRaw_WHR_2018

UNION

	SELECT 2019 AS Year, Country, Happiness_Score, Overall_Happiness_Rank, Happiness_Status,
	GDP_per_Capita, Family, Health, Freedom, CAST([Trust] AS VARCHAR(20)) AS Trust, Generosity

	FROM dbo.newRaw_WHR_2019

/* List all the data which has been inserted in table "dbo.WHR_Combined2". */

SELECT *
FROM dbo.WHR_Combined2  /* There are 624 records in table "dbo.newRaw_WHR_2019" */
```

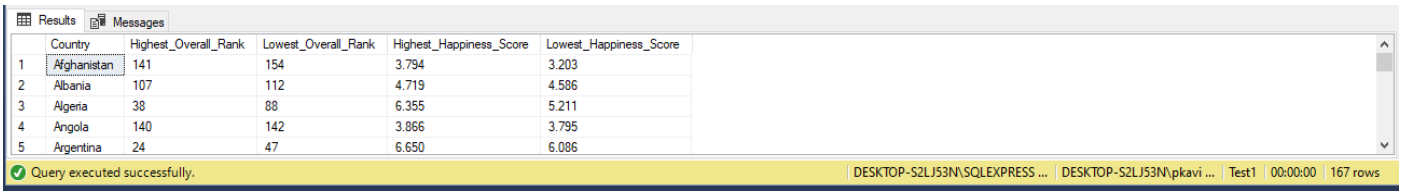


	Year	Country	Happiness_Score	Overall_Happiness_Rank	Happiness_Status	GDP_per_Capita	Family	Health	Freedom	Trust	Generosity
1	2016	Afghanistan	3.360	154	AMBER	0.38227	0.11037	0.17344	0.16430	0.07112	0.31268
2	2016	Albania	4.655	109	AMBER	0.95530	0.50163	0.73007	0.31866	0.05301	0.16840
3	2016	Algeria	6.355	38	GREEN	1.05266	0.83309	0.61804	0.21006	0.16157	0.07044
4	2016	Angola	3.866	141	AMBER	0.84731	0.66366	0.04991	0.00589	0.08434	0.12071
5	2016	Argentina	6.650	26	GREEN	1.15137	1.06612	0.69711	0.42284	0.07296	0.10989

Query executed successfully. DESKTOP-S2LJ53N\SQLEXPRESS ... DESKTOP-S2LJ53N\pkavi ... Test1 00:00:00 624 rows

SQL Statement to return “Highest Rank”, “Lowest Rank”, “Highest Happiness Score” and “Lowest Happiness Score” for all the years, we need to have a combined list, which holds data for all the years.

```
SELECT "Country", MIN(Overall_Happiness_Rank) AS Highest_Overall_Rank, MAX(Overall_Happiness_Rank)
AS Lowest_Overall_Rank,

		MAX(Happiness_Score) AS Highest_Happiness_Score, MIN(Happiness_Score) AS
Lowest_Happiness_Score

	FROM dbo.WHR_Combined2

	GROUP BY "Country"
```



	Country	Highest_Overall_Rank	Lowest_Overall_Rank	Highest_Happiness_Score	Lowest_Happiness_Score
1	Afghanistan	141	154	3.794	3.203
2	Albania	107	112	4.719	4.586
3	Algeria	38	88	6.355	5.211
4	Angola	140	142	3.866	3.795
5	Argentina	24	47	6.650	6.086

Query executed successfully. DESKTOP-S2LJ53N\SQLEXPRESS ... DESKTOP-S2LJ53N\pkavi ... Test1 00:00:00 167 rows

Table: “dbo.Task_4”

```
/* - Creating table "dbo.Task_4" to return Highest Rank, Lowest Rank, Highest Happiness Score,
Lowest Happiness Score per Country for all the years. */

CREATE TABLE dbo.Task_4(
			Country VARCHAR(100) NULL,
			Highest_Overall_Rank INT DEFAULT NULL,
			Lowest_Overall_Rank INT DEFAULT NULL,
			Highest_Happiness_Score DECIMAL(6,3),
			Lowest_Happiness_Score DECIMAL(6,3)
			)

/* Using "TRUNCATE" to remove all rows (data) from a table "dbo.Task_4". */

TRUNCATE TABLE dbo.Task_4
```

```
/* Inserting data into "dbo.Task_4". */

INSERT INTO dbo.Task_4

	SELECT "Country", MIN(Overall_Happiness_Rank) AS Highest_Overall_Rank,
MAX(Overall_Happiness_Rank) AS Lowest_Overall_Rank,

			MAX(Happiness_Score) AS Highest_Happiness_Score, MIN(Happiness_Score) AS
Lowest_Happiness_Score

	FROM dbo.WHR_Combined2

	GROUP BY "Country"

/* List the data which has been inserted in table "dbo.Country_Data". */

SELECT *
FROM dbo.Task_4 /* There are 167 records in table "dbo.Task_4" */
```

Results | Messages

	Country	Highest_Overall_Rank	Lowest_Overall_Rank	Highest_Happiness_Score	Lowest_Happiness_Score
1	Afghanistan	141	154	3.794	3.203
2	Albania	107	112	4.719	4.586
3	Algeria	38	88	6.355	5.211
4	Angola	140	142	3.866	3.795
5	Argentina	24	47	6.650	6.086

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 167 rows

Task 5.

Create a dataset containing the details as per table below (Complement the existing data pipeline in step 2 to return the required information) and use this dataset to create a small Data Visualization dashboard by plotting the data in a world map showing the evolution by year. When hovering on a particular location the map the actual score and country image should be displayed.

Column Name	Specifications
Year	
Country	
Happiness Score	
Happiness Status	Return the following value based on the Happiness Score If Happiness Score is > 5.6, display “Green”. If Happiness Score is between 2.6 and 5.6, display “Amber”. If Happiness Score is <2.6, display “Red”.

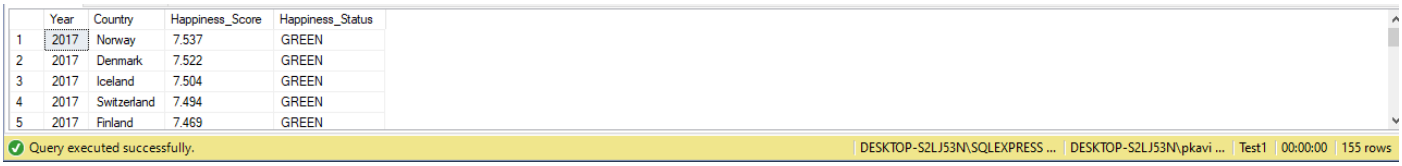
SQL Statement to return “Year”, “Country”,”Happiness Score” and “Happiness Status”.

```
-FOR 2016-

	SELECT 2016 AS Year, Country, Happiness_Score, Happiness_Status

	FROM dbo.newRaw_WHR_2016
```

	Year	Country	Happiness_Score	Happiness_Status
1	2016	Denmark	7.526	GREEN
2	2016	Switzerland	7.509	GREEN
3	2016	Iceland	7.501	GREEN
4	2016	Norway	7.498	GREEN
5	2016	Finland	7.413	GREEN

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 157 rows

```
-FOR 2017-

	SELECT 2017 AS Year, Country, Happiness_Score, Happiness_Status

	FROM dbo.newRaw_WHR_2017
```



	Year	Country	Happiness_Score	Happiness_Status
1	2017	Norway	7.537	GREEN
2	2017	Denmark	7.522	GREEN
3	2017	Iceland	7.504	GREEN
4	2017	Switzerland	7.494	GREEN
5	2017	Finland	7.469	GREEN

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 155 rows

```
-FOR 2018-

	SELECT 2018 AS Year, Country, Happiness_Score, Happiness_Status

	FROM dbo.newRaw_WHR_2018
```

	Year	Country	Happiness_Score	Happiness_Status
1	2018	Finland	7.632	GREEN
2	2018	Norway	7.594	GREEN
3	2018	Denmark	7.555	GREEN
4	2018	Iceland	7.495	GREEN
5	2018	Switzerland	7.487	GREEN

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 156 rows

```
-FOR 2019-

	SELECT 2019 AS Year, Country, Happiness_Score, Happiness_Status

	FROM dbo.newRaw_WHR_2019
```

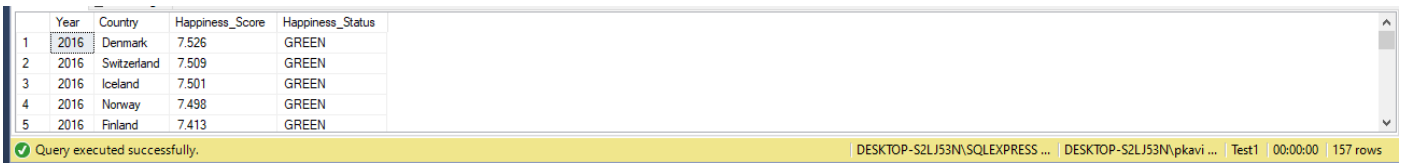
	Year	Country	Happiness_Score	Happiness_Status
1	2019	Finland	7.769	GREEN
2	2019	Denmark	7.600	GREEN
3	2019	Norway	7.554	GREEN
4	2019	Iceland	7.494	GREEN
5	2019	Netherlands	7.488	GREEN

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 156 rows

Since the SQL Statement returns the required values, we will create a table for each of the year so that a "Tableau" document can be created for each of the year.

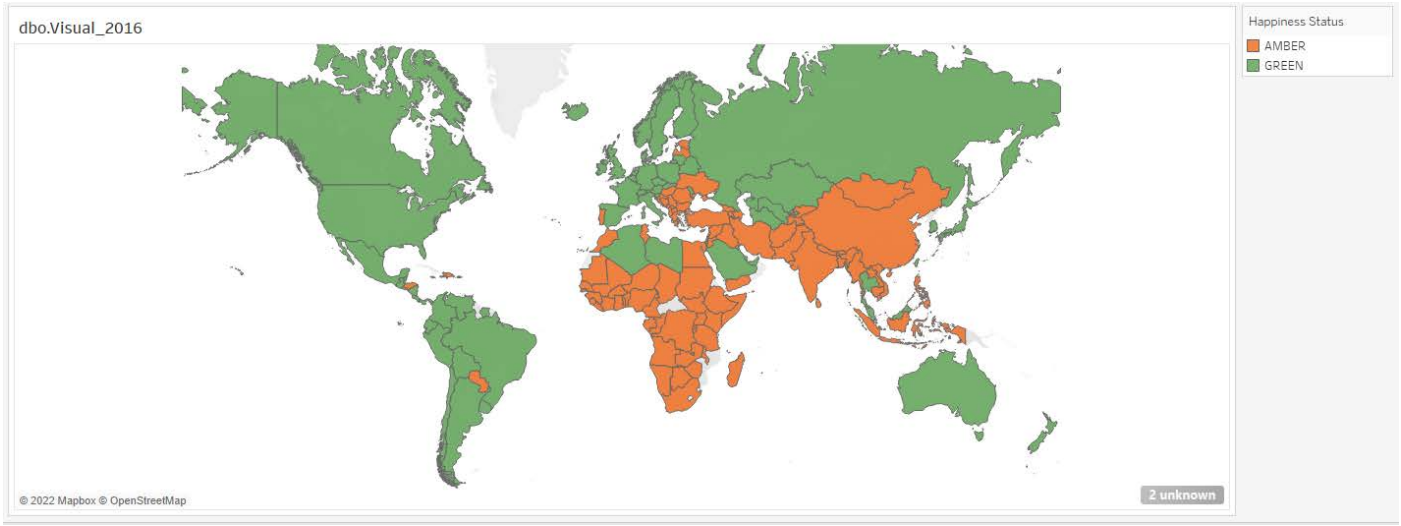
```
/* - Creating table "dbo.Visual_2016", "dbo.Visual_2017", "dbo.Visual_2018" and "dbo.Visual_2019" to
return Year, Country, Happiness_Score, Happiness_Status and complement it to create a small Data
Visualization dashboard by plotting the data in a World Map showing the evolution by year. */

-FOR 2016-

CREATE TABLE dbo.Visual_2016(
                                Year INT DEFAULT NULL,
                                Country VARCHAR(100) NULL,
                                Happiness_Score DECIMAL(6,3),
                                Happiness_Status VARCHAR(20) NULL
                                )

-FOR 2017-

CREATE TABLE dbo.Visual_2017(
                                Year INT DEFAULT NULL,
                                Country VARCHAR(100) NULL,
                                Happiness_Score DECIMAL(6,3),
                                Happiness_Status VARCHAR(20) NULL
                                )

-FOR 2018-

CREATE TABLE dbo.Visual_2018(
                                Year INT DEFAULT NULL,
                                Country VARCHAR(100) NULL,
                                Happiness_Score DECIMAL(6,3),
                                Happiness_Status VARCHAR(20) NULL
                                )
```

```
-FOR 2019-

CREATE TABLE dbo.Visual_2019(
                                    Year INT DEFAULT NULL,
                                    Country VARCHAR(100) NULL,
                                    Happiness_Score DECIMAL(6,3),
                                    Happiness_Status VARCHAR(20) NULL
                                    )

/* Using "TRUNCATE" to remove all rows (data) from the tables "dbo.Visual_2016", "dbo.Visual_2017",
"dbo.Visual_2018" and "dbo.Visual_2019". */

TRUNCATE TABLE dbo.Visual_2016
TRUNCATE TABLE dbo.Visual_2017
TRUNCATE TABLE dbo.Visual_2018
TRUNCATE TABLE dbo.Visual_2019

/* Inserting data into tables "dbo.Visual_2016", "dbo.Visual_2017", "dbo.Visual_2018" and
"dbo.Visual_2019". */

--FOR 2016--

INSERT INTO dbo.Visual_2016

       SELECT 2016 AS Year, Country, Happiness_Score, Happiness_Status

       FROM dbo.newRaw_WHR_2016  /* - 157 rows inserted in table "dbo.Visual_2016". */

--FOR 2017--

INSERT INTO dbo.Visual_2017

       SELECT 2017 AS Year, Country, Happiness_Score, Happiness_Status

       FROM dbo.newRaw_WHR_2017  /* - 155 rows inserted in table "dbo.Visual_2017". */

--FOR 2018--

INSERT INTO dbo.Visual_2018

       SELECT 2018 AS Year, Country, Happiness_Score, Happiness_Status

       FROM dbo.newRaw_WHR_2018  /* - 156 rows inserted in table "dbo.Visual_2018". */

--FOR 2019--

INSERT INTO dbo.Visual_2019

       SELECT 2019 AS Year, Country, Happiness_Score, Happiness_Status

       FROM dbo.newRaw_WHR_2019  /* - 156 rows inserted in table "dbo.Visual_2019". */

/* List the data which has been inserted in tables "dbo.Visual_2016", "dbo.Visual_2017",
"dbo.Visual_2018" and "dbo.Visual_2019". */

-- FOR 2016 --

SELECT *
FROM dbo.Visual_2016 /* There are 157 records in table "dbo.Visual_2016" */
```



	Year	Country	Happiness_Score	Happiness_Status
1	2016	Denmark	7.526	GREEN
2	2016	Switzerland	7.509	GREEN
3	2016	Iceland	7.501	GREEN
4	2016	Norway	7.498	GREEN
5	2016	Finland	7.413	GREEN

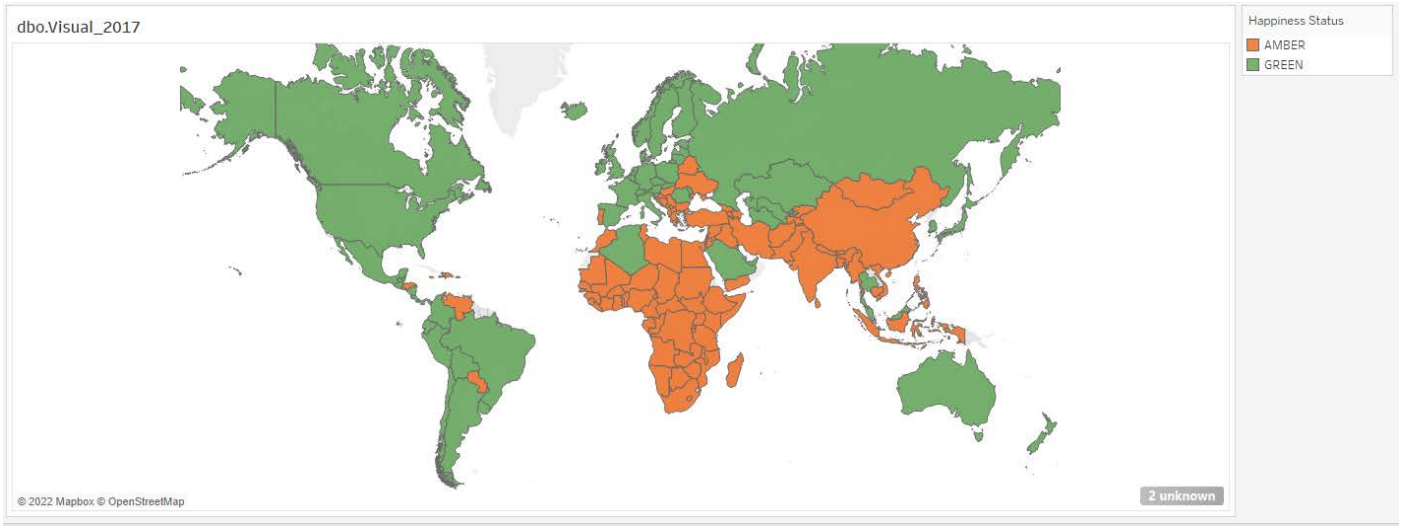
Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 157 rows




```sql
-- FOR 2017 --

SELECT *
FROM dbo.Visual_2017 /* There are 155 records in table "dbo.Visual_2017" */
```

	Year	Country	Happiness_Score	Happiness_Status
1	2017	Norway	7.537	GREEN
2	2017	Denmark	7.522	GREEN
3	2017	Iceland	7.504	GREEN
4	2017	Switzerland	7.494	GREEN
5	2017	Finland	7.469	GREEN

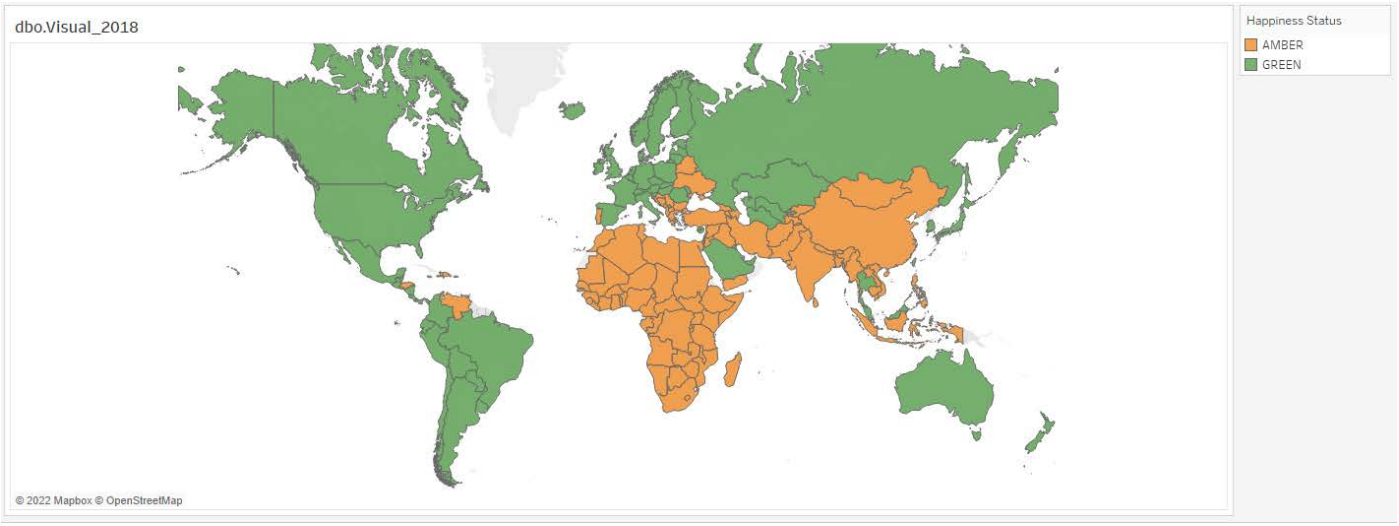
Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 155 rows




```sql
-- FOR 2018 --

SELECT *
FROM dbo.Visual_2018 /* There are 156 records in table "dbo.Visual_2018" */
```

	Year	Country	Happiness_Score	Happiness_Status
1	2018	Finland	7.632	GREEN
2	2018	Norway	7.594	GREEN
3	2018	Denmark	7.555	GREEN
4	2018	Iceland	7.495	GREEN
5	2018	Switzerland	7.487	GREEN

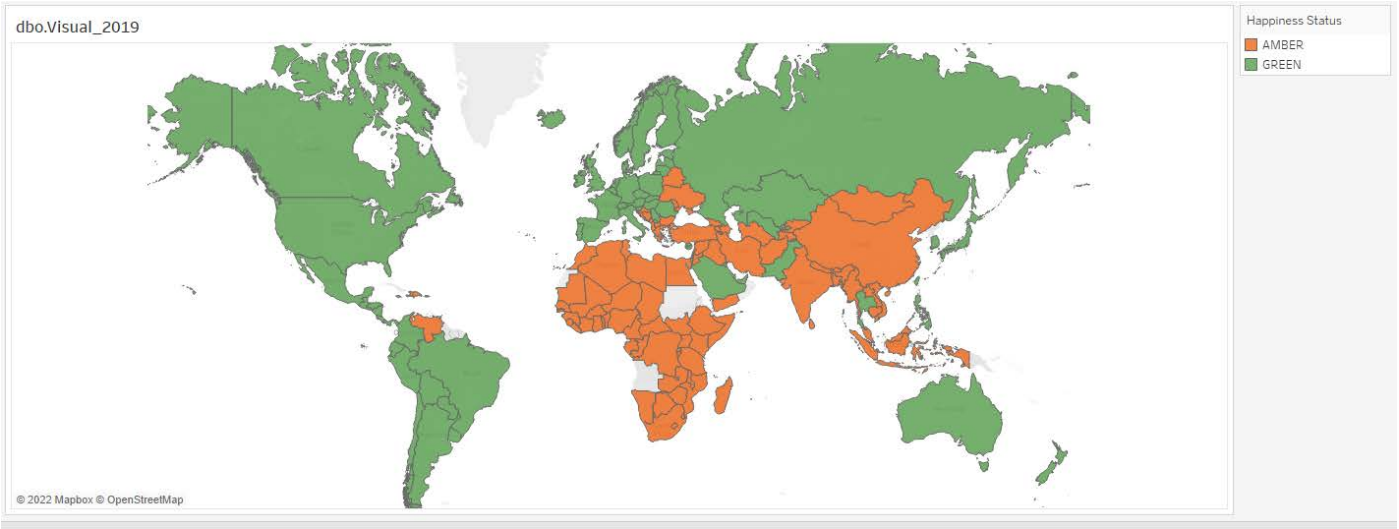
Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 156 rows




```
-- FOR 2019 --

SELECT *
FROM dbo.Visual_2019 /* There are 156 records in table "dbo.Visual_2019" */
```

	Year	Country	Happiness_Score	Happiness_Status
1	2019	Finland	7.769	GREEN
2	2019	Denmark	7.600	GREEN
3	2019	Norway	7.554	GREEN
4	2019	Iceland	7.494	GREEN
5	2019	Netherlands	7.488	GREEN

Query executed successfully. | DESKTOP-S2LJ53N\SQLEXPRESS ... | DESKTOP-S2LJ53N\pkavi ... | Test1 | 00:00:00 | 156 rows



Note: The Tableau for “dbo.Visual_2016”, “dbo.Visual_2017”, “dbo.Visual_2018” and “dbo.Visual_2019” is attached as part of the assignment. Please note that “Country Image” is available as per screenshot below:

Task 6.

The World Bank provides several APIs to have access global data. Complement the data pipeline in Step 3 to add three new columns 'Capital City', 'Longitude' and 'Latitude' using the API:-http://api.worldbank.org/v2/country. More information on how to use the API can be found on the following URL:- https://datahelpdesk.worldbank.org/knowledgebase/articles/898590-country-apiqueries

Creating table "dbo.Country_Data" to import the required Country data from link: http://api.worldbank.org/v2/country

```
/* --- Task 6:- Creating table "dbo.Country_Data" (downloaded from
"https://datahelpdesk.worldbank.org/knowledgebase/articles/898590-country-api-queries") for storing
Country data with Capital City, Longitude and Latitude. --- */

CREATE TABLE dbo.Country_Data(
                                    Country_ID VARCHAR(20) NULL,
                                    iso2Code VARCHAR(20) NULL,
                                    Country_Name VARCHAR(100) NULL,
                                    Region VARCHAR(MAX) NULL,
                                    IncomeLevel VARCHAR(MAX) NULL,
                                    LendingType VARCHAR(MAX) NULL,
                                    Capital_City VARCHAR(MAX) NULL,
                                    Lng DECIMAL(12,9) DEFAULT NULL,
                                    Lat DECIMAL(12,9) DEFAULT NULL
                                    )

/* Using "TRUNCATE" to remove all rows (data) from a table and "BULK INSERT" to populate the tables.
*/

TRUNCATE TABLE dbo.Country_Data

/* Inserting data into "dbo.Country_Data" from "Country_Data.csv". */

BULK INSERT dbo.Country
FROM 'C:\Users\pkavi\Documents\MCB_Assignment\Country_Data.csv'

WITH
(

       FIELDQUOTE = '"',
       FIRSTROW=2, /* Import of data starts as from row 2, else header will be imported as well. */
       FORMAT='CSV',
       FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n'

)

/* List the data which has been inserted in table "dbo.Country_Data". */

SELECT *
FROM dbo.Country_Data /* There are 624 records in table "dbo.Country_Data" */
```

Merging "Capital_City" AS Capital City, "Lng" AS Longitude and "Lat" AS Latitude in the dataframe "dbo.WHR_Combined2" to get Country, Happiness Score, Happiness Status and Overall Happiness Rank together.

SQL Statement:

```
SELECT a.Year, a.Country, (b.Capital_City) AS Capital_City, (b.Lng) AS Longitude, (b.Lat) AS
Latitude, a.Happiness_Score, a.Overall_Happiness_Rank,
a.Happiness_Status, a.GDP_per_Capita, a.Family, a.Health, a.Freedom, a.Generosity, a.Trust

FROM a.dbo.WHR_Combined2 a
INNER JOIN dbo.Country_Data b
ON a.Country=b.Country
```

Table: “dbo.CombinedCountry”:

```
/* Creating table "dbo.CombinedCountry" to return Capital City as well as Latitute and Longitude
with data in "dbo.WHR_Combined2" table. */

CREATE TABLE dbo.CombinedCountry(
                                    Year INT DEFAULT NULL,
                                    Country VARCHAR(100) NULL,
                                    Capital_City VARCHAR(MAX) NULL,
                                    Lng DECIMAL(12,9) DEFAULT NULL,
                                    Lat DECIMAL(12,9) DEFAULT NULL,
                                    Happiness_Score DECIMAL(6,3),
                                    Overall_Happiness_Rank INT DEFAULT NULL,
                                    Happiness_Status VARCHAR(20) NULL,
                                    GDP_per_Capita DECIMAL(7,5),
                                    Family DECIMAL(7,5),
                                    Health DECIMAL(7,5),
                                    Freedom DECIMAL(7,5),
                                    Generosity DECIMAL(7,5),
                                    Trust DECIMAL(7,5),
                                    )

/* Using "TRUNCATE" to remove all rows (data) from table "dbo.CombinedCountry". */

TRUNCATE TABLE dbo.CombinedCountry

/* Inserting data into "dbo.CombinedCountry" from "dbo.Country_Data" and "dbo.WHR_Combined2" tables.
*/

INSERT INTO dbo.CombinedCountry

SELECT a.Year, a.Country, (b.Capital_City) AS Capital_City, (b.Lat) AS Latitude, (b.Lng) AS
Longitude, a.Happiness_Score, a.Overall_Happiness_Rank, a.Happiness_Status, a.GDP_per_Capita,
a.Family, a.Health, a.Freedom, a.Generosity, a.Trust

FROM a.dbo.WHR_Combined2 a
INNER JOIN dbo.Country_Data b
ON a.Country=b.Country


/* List the data which has been inserted in table "dbo.Country_Data". */

SELECT *
FROM dbo.CombinedCountry /* There are 624 records in table "dbo.Country_Data" */
```

-R PROGRAMMING

R Studio-

Tables used for analysis and review of regression and cluster model:

- dbo.Analysis1 and
- dbo.Combined2.

SQL Statement:

Table: "newRaw_WHR_2016".

```
------- R STUDIO -------

/* - Creating table "dbo.Analysis1" which will be used in "RStudio" for analysis purposes. */

CREATE TABLE dbo.Analysis1(
                            Year INT DEFAULT NULL,
                            Country VARCHAR(100) NULL,
                            Happiness_Score DECIMAL(6,3),
                            Overall_Happiness_Rank INT DEFAULT NULL,
                            Happiness_Status VARCHAR(20) NULL,
                            GDP_per_Capita DECIMAL(7,5),
                            Family DECIMAL(7,5),
                            Health DECIMAL(7,5),
                            Freedom DECIMAL(7,5),
                            Generosity DECIMAL(7,5),
                            Trust DECIMAL(7,5)
                            )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.Analysis1". */

TRUNCATE TABLE dbo.Analysis1

/* - Inserting data in "dbo.Analysis1". */

INSERT INTO dbo.Analysis1

SELECT a.Year, a.Country, b.Happiness_Score, b.Overall_Happiness_Rank, b.Happiness_Status,
b.GDP_per_Capita, b.Family, b.Health, b.Freedom, b.Generosity, b.Trust

FROM (

       SELECT DISTINCT c.Country, d.Year
       FROM
              (
              SELECT e.Country
              FROM dbo.WHR_Combined_2016 e

              INNER JOIN

                     dbo.WHR_Combined_2017 f

              ON e.Country=f.Country

              INNER JOIN

                     dbo.WHR_Combined_2018 g

              ON e.Country=g.Country

              INNER JOIN

                     dbo.WHR_Combined_2019 h

              ON e.Country=h.Country) c
```

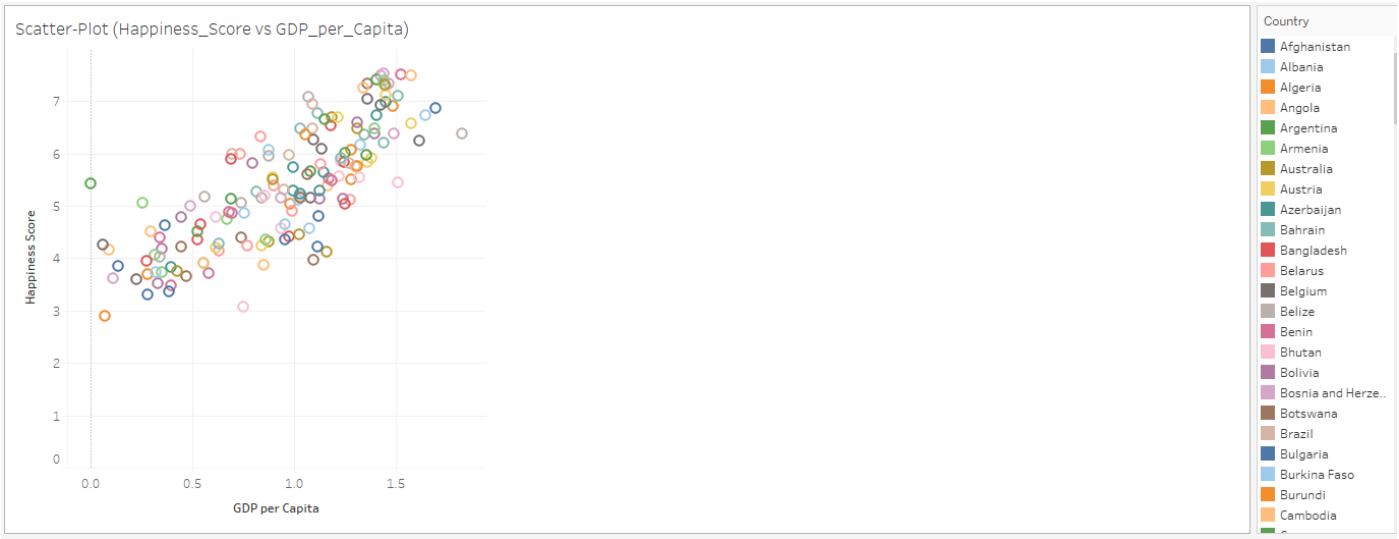
```
        FULL JOIN

                (SELECT DISTINCT Year, Country, Region, Regional_Happiness_Rank
                FROM dbo.WHR_CombinedList) d

                ON c.Country=d.Country

                WHERE c.Country is not NULL AND
                                d.Country is not NULL) a

        LEFT JOIN
                        dbo.WHR_Combined2 b

        ON a.Country=b.Country AND a.Year=b.Year

/* List all the data in table "dbo.Analysis1". */

SELECT *
FROM dbo.Analysis1  /* 564 rows displayed. */


/* - Creating table "dbo.Combined2" which will be used in "RStudio" for analysis purposes. */

/* - Creating table "dbo.Combined2" to return values to be used in R Studio. */

CREATE TABLE dbo.Combined2(
                                Year INT DEFAULT NULL,
                                Country VARCHAR(100) NULL,
                                Region VARCHAR(100) NULL,
                                Happiness_Score DECIMAL(6,3),
                                Overall_Happiness_Rank INT DEFAULT NULL,
                                Regional_Happiness_Rank INT DEFAULT NULL,
                                Happiness_Status VARCHAR(20) NULL,
                                GDP_per_Capita DECIMAL(7,5),
                                Family DECIMAL(7,5),
                                Health DECIMAL(7,5),
                                Freedom DECIMAL(7,5),
                                Generosity DECIMAL(7,5),
                                Trust DECIMAL(7,5)
                                )

/* Using "TRUNCATE" to remove all rows (data) from a table"dbo.Combined2". */

TRUNCATE TABLE dbo.Combined2

/* - Inserting data in "dbo.Analysis1". */

INSERT INTO dbo.Combined2

SELECT a.Year, a.Country, b.Region, a.Happiness_Score, a.Overall_Happiness_Rank,
b.Regional_Happiness_Rank, a.Happiness_Status, a.GDP_per_Capita, a.Family, a.Health, a.Freedom,
a.Generosity, a.Trust

FROM dbo.Analysis1 a
INNER JOIN dbo.WHR_CombinedList b
ON a.Country=b.Country


/* List all the data in table "dbo.Analysis1". */

SELECT *
FROM dbo.Combined2  /* 606 rows displayed. */
```

For Year 2016 – “newWHR2016.r”:

```
# FOR NUMERICAL ANALYTICS
import numpy as np

# TO STORE AND PROCESS DATA IN DATAFRAME
import pandas as pd
import os

# BASIC VISUALIZATION PACKAGE
import matplotlib.pyplot as plt

# ADVANCED PLOTING
import seaborn as seabornInstance

# TRAIN TEST SPLIT
from sklearn.model_selection import train_test_split

# INTERACTIVE VISUALIZATION
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly.express as px
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)

import statsmodels.formula.api as stats
from statsmodels.formula.api import ols
from sklearn import datasets
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from discover_feature_relationships import discover

#newWHR2016 data
df_16 = pd.read_csv('newWHR2016.csv')
#df_16.describe()
#df_16.info()
Usecols =
['Year','Country','Happiness_Score','Overall_Happiness_Rank','GDP_per_Capita','Family','Health','Freedom','Trus
t','Generosity']
df_16.drop(['Happiness_Status','Lower_Confidence_Interval','Upper_Confidence_Interval','Dystopia'],axis=1,inpl
ace=True)
df_16.columns =
['Year','Country','Happiness_Score','Overall_Happiness_Rank','GDP_per_Capita','Family','Health','Freedom','Trus
t','Generosity']

#df_16['Year'] = 2016 #add year column (IF WE HAD TO ADD 'YEAR')

df_16.head()

target = ['Top','Top-Mid', 'Low-Mid', 'Low' ]
target_n = [4, 3, 2, 1]

df_16["target"] = pd.qcut(df_16['Overall_Happiness_Rank'], len(target), labels=target)
df_16["target_n"] = pd.qcut(df_16['Overall_Happiness_Rank'], len(target), labels=target_n)
```

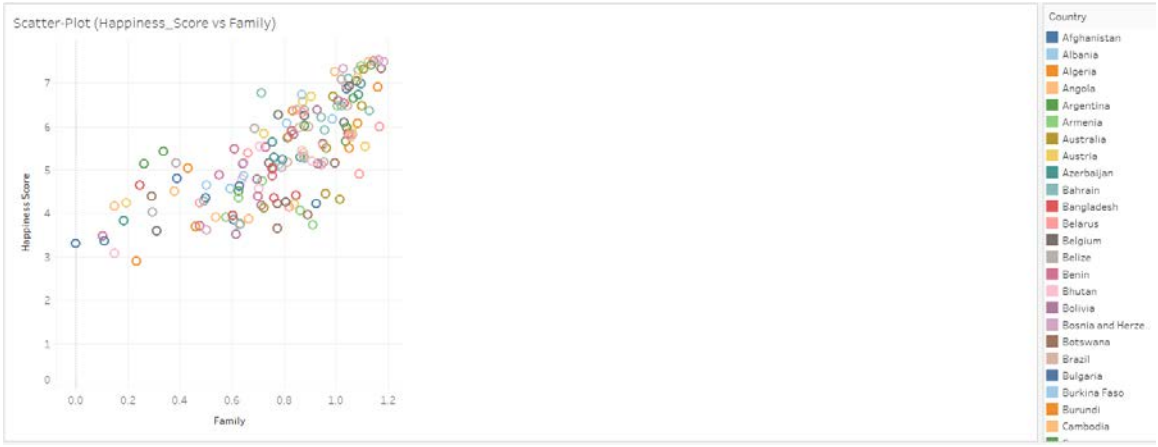
```
#COMBINING ALL DATA FILE TO finaldf16

# APPENDING ALL TOGETHER
finaldf16 = df_16.append([df_16,df_17,df_18,df_19])
# finaldf16.dropna(inplace = True)

#CHECKING FOR MISSING DATA
finaldf16.isnull().any()

# FILLING MISSING VALUES OF "TRUST" WITH ITS MEAN
finaldf16.Trust.fillna((finaldf16.Trust.mean()), inplace = True)
finaldf16.head(10)

# Statistical details can be seen using "describe()" function.
# Defining an empty dataframe "DataFrame16". This dataframe includes Root Mean Squared Error (RMSE), R-squared, Adjusted R-squared, and mean of the R-squared values obtained by the k-Fold Cross-Validation, which are the essential metrics to compare different models.
# Having an R-squared value closer to one and smaller RMSE means a better fit.
# Filling this dataframe with the results.

evaluation = pd.DataFrame16({'Model':[],
                  'Details':[],
                  'Root Mean Squared Error (RMSE)': [],
                  'R-squared (training)': [],
                  'Adjusted R-squared (training)': [],
                  'R-squared (test)':[],
                  'Adjusted R-squared(test)':[],
                   '5-Fold Cross Validation':[]
                 })

#How Happiness_Score is distributed.
# Relationship of different variables with Happiness_Score.
#----- "Happiness_Score vs GDP_per_Capita" using a Scatter plot. -----
px.scatter(finaldf16, x="GDP_per_Capita", y="Happiness_Score", animation_frame="Year", animation_group="Country", size="Overall_Happiness_Rank", color="Country", hover_name="Country", trendline= "ols")
# TRAIN THE DATA SET.
train_data, test_data = train_test_split(finaldf16, train_size = 0.8, random_state = 3)
lr = LinearRegression()
X_train = np.array(train_data['GDP_per_Capita'], dtype = pd.Series).reshape(-1,1)
Y_train = np.array(train_data['Happiness_Score'], dtype = pd.Series)
lr.fit(X_train, Y_train)
# TEST DATA SET.
X_test = np.array(test_data['GDP_per_Capita'], dtype = pd.Series).reshape(-1,1)
Y_test = np.array(test_data['Happiness_Score'], dtype = pd.Series)

pred = lr.predict(X_test)

#ROOT MEAN SQUARED ERROR
rmsesm = float(format(np.sqrt(metrics.mean_squared_error(Y_test,pred)),'.3f'))
#R-SQUARED (TRAINING)
rtrsm = float(format(lr.score(X_train, Y_train),'.3f'))
#R-SQUARED (TEST)
rtesm = float(format(lr.score(X_test, Y_test),'.3f'))
cv = float(format(cross_val_score(lr,finaldf16[['GDP_per_Capita']],finaldf16['Happiness_Score'],cv=5).mean(),'.3f'))
print ("Average Score for Test Data: {:.3f}".format(Y_test.mean()))
print('Intercept: {}'.format(lr.intercept_))
print('Coefficient: {}'.format(lr.coef_))
```

```
r = evaluation.shape[0]
evaluation.loc[r] = ['Simple Linear Regression','-',rmsesm,rtrsm,'-',rtesm,'-',cv]
evaluation
```



(Scatter Plot devised on Tableau using table “dbo.newRaw_WHR_2016”.)

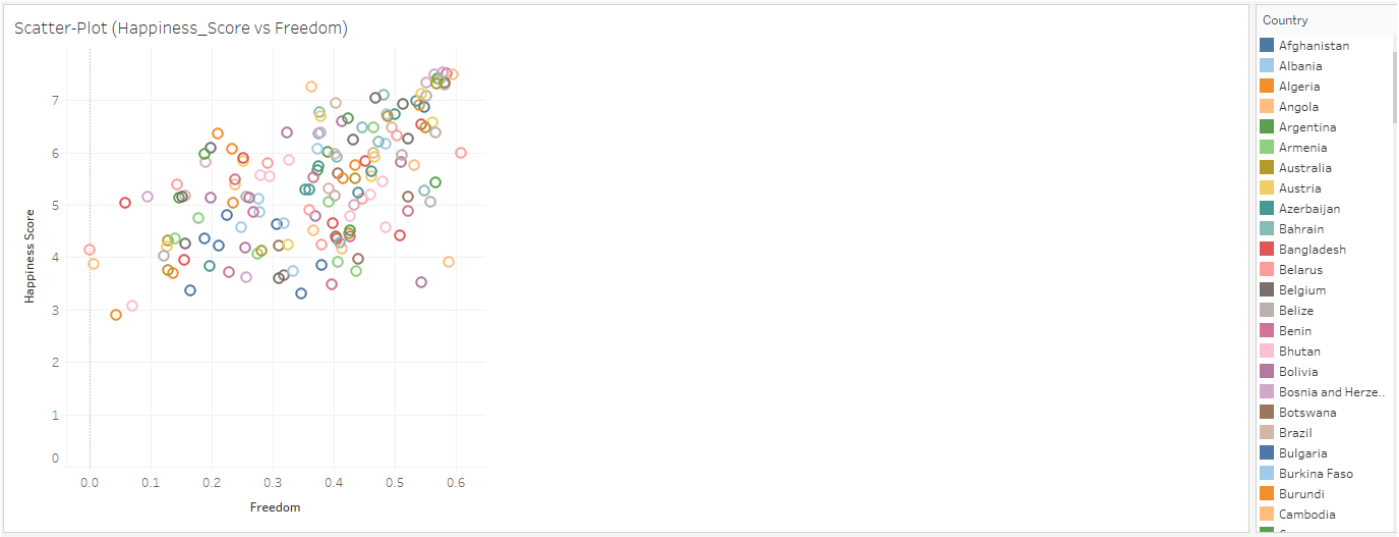
```
#----- Chart to determine result of "Simple Regression". -----
seabornInstance.set_style(style='whitegrid')
plt.figure(figsize=(12,6))
plt.scatter(X_test,Y_test,color='blue',label="Data", s = 12)
plt.plot(X_test,lr.predict(X_test),color="red",label="Predicted Regression Line")
plt.xlabel("GDP_per_Capita", fontsize=15)
plt.ylabel("Happiness_Score", fontsize=15)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.legend()
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)


#----- "Happiness_Score vs Family" using a Scatter plot. -----
px.scatter(finaldf16, x="Family", y="Happiness_Score", animation_frame="Year", animation_group="Country",
size="Overall_Happiness_Rank", color="Country", hover_name="Country", trendline= "ols")

# TRAIN THE DATA SET.

train_data, test_data = train_test_split(finaldf16, train_size = 0.8, random_state = 3)
lr = LinearRegression()
X_train = np.array(train_data['Family'], dtype = pd.Series).reshape(-1,1)
Y_train = np.array(train_data['Happiness_Score'], dtype = pd.Series)
lr.fit(X_train, Y_train)
# TEST DATA SET.

X_test = np.array(test_data['Family'], dtype = pd.Series).reshape(-1,1)
Y_test = np.array(test_data['Happiness_Score'], dtype = pd.Series)

pred = lr.predict(X_test)
```




```
#-----Happiness_Score vs Family-----
#----- Chart to determine result of "Simple Regression". -----
seabornInstance.set_style(style='whitegrid')
plt.figure(figsize=(12,6))
plt.scatter(X_test,Y_test,color='blue',label="Data", s = 12)
plt.plot(X_test,lr.predict(X_test),color="red",label="Predicted Regression Line")
plt.xlabel("Family", fontsize=15)
plt.ylabel("Happiness_Score", fontsize=15)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.legend()
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
```

```
#----- "Happiness_Score vs Health" using a Scatter plot. -----
px.scatter(finaldf16, x="Health", y="Happiness_Score", animation_frame="Year", animation_group="Country",
size="Overall_Happiness_Rank", color="Country", hover_name="Country", trendline= "ols")
# TRAIN THE DATA SET.
train_data, test_data = train_test_split(finaldf16, train_size = 0.8, random_state = 3)
lr = LinearRegression()
X_train = np.array(train_data['Health'], dtype = pd.Series).reshape(-1,1)
Y_train = np.array(train_data['Happiness_Score'], dtype = pd.Series)
lr.fit(X_train, Y_train)
# TEST DATA SET.
X_test = np.array(test_data['Health'], dtype = pd.Series).reshape(-1,1)
Y_test = np.array(test_data['Happiness_Score'], dtype = pd.Series)

pred = lr.predict(X_test)
```

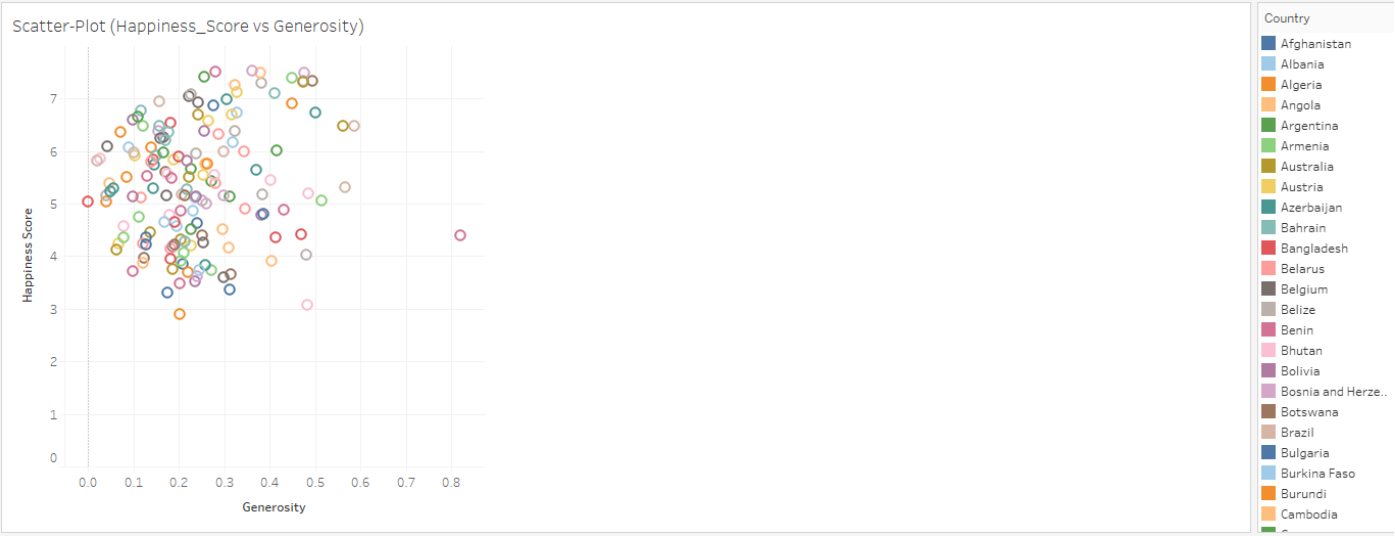
```
#-----Happiness_Score vs Health-----
#----- Chart to determine result of "Simple Regression". -----
seabornInstance.set_style(style='whitegrid')
plt.figure(figsize=(12,6))
plt.scatter(X_test,Y_test,color='blue',label="Data", s = 12)
plt.plot(X_test,lr.predict(X_test),color="red",label="Predicted Regression Line")
plt.xlabel("Health", fontsize=15)
plt.ylabel("Happiness_Score", fontsize=15)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.legend()
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)

#----- "Happiness_Score vs Freedom" using a Scatter plot. -----
px.scatter(finaldf16, x="Freedom", y="Happiness_Score", animation_frame="Year", animation_group="Country",
size="Overall_Happiness_Rank", color="Country", hover_name="Country", trendline= "ols")
# TRAIN THE DATA SET.
train_data, test_data = train_test_split(finaldf16, train_size = 0.8, random_state = 3)
lr = LinearRegression()
X_train = np.array(train_data['Freedom'], dtype = pd.Series).reshape(-1,1)
Y_train = np.array(train_data['Happiness_Score'], dtype = pd.Series)
lr.fit(X_train, Y_train)
# TEST DATA SET.
X_test = np.array(test_data['Freedom'], dtype = pd.Series).reshape(-1,1)
Y_test = np.array(test_data['Happiness_Score'], dtype = pd.Series)

pred = lr.predict(X_test)
```




```
#-----Happiness_Score vs Freedom-----
#----- Chart to determine result of "Simple Regression". -----
seabornInstance.set_style(style='whitegrid')
plt.figure(figsize=(12,6))
plt.scatter(X_test,Y_test,color='blue',label="Data", s = 12)
plt.plot(X_test,lr.predict(X_test),color="red",label="Predicted Regression Line")
plt.xlabel("Freedom", fontsize=15)
plt.ylabel("Happiness_Score", fontsize=15)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.legend()
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
```

#----- "Happiness_Score vs Generosity" using a Scatter plot. -----

```
px.scatter(finaldf16, x="Generosity", y="Happiness_Score", animation_frame="Year", animation_group="Country", size="Overall_Happiness_Rank", color="Country", hover_name="Country", trendline= "ols")
# TRAIN THE DATA SET.
train_data, test_data = train_test_split(finaldf16, train_size = 0.8, random_state = 3)
lr = LinearRegression()
X_train = np.array(train_data['Generosity'], dtype = pd.Series).reshape(-1,1)
Y_train = np.array(train_data['Happiness_Score'], dtype = pd.Series)
lr.fit(X_train, Y_train)
# TEST DATA SET.
X_test = np.array(test_data['Generosity'], dtype = pd.Series).reshape(-1,1)
Y_test = np.array(test_data['Happiness_Score'], dtype = pd.Series)

pred = lr.predict(X_test)
```



#-----Happiness_Score vs Generosity-----

#----- Chart to determine result of "Simple Regression". -----

```
seabornInstance.set_style(style='whitegrid')
plt.figure(figsize=(12,6))
plt.scatter(X_test,Y_test,color='blue',label="Data", s = 12)
plt.plot(X_test,lr.predict(X_test),color="red",label="Predicted Regression Line")
plt.xlabel("Generosity", fontsize=15)
plt.ylabel("Happiness_Score", fontsize=15)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.legend()
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
```

Note: (Scatter Plot, Linear Regression, Distribution of numeric data, Seaborn Instance Bar Plot and Pearson Correlation Matrix unavailable due to values in "Trust" table in "VARCHAR" due to "N/A" value for "United Arab Emirates" in 2018.csv.)

```
#----- "Happiness_Score vs Trust" using a Scatter plot. -----

px.scatter(finaldf16, x="Trust", y="Happiness_Score", animation_frame="Year", animation_group="Country",
size="Overall_Happiness_Rank", color="Country", hover_name="Country", trendline= "ols")
# TRAIN THE DATA SET.
train_data, test_data = train_test_split(finaldf16, train_size = 0.8, random_state = 3)
lr = LinearRegression()
X_train = np.array(train_data['Trust'], dtype = pd.Series).reshape(-1,1)
Y_train = np.array(train_data['Happiness_Score'], dtype = pd.Series)
lr.fit(X_train, Y_train)
# TEST DATA SET.
X_test = np.array(test_data['Trust'], dtype = pd.Series).reshape(-1,1)
Y_test = np.array(test_data['Happiness_Score'], dtype = pd.Series)

pred = lr.predict(X_test)

#-----Happiness_Score vs Trust-----

#----- Chart to determine result of "Simple Regression". -----

seabornInstance.set_style(style='whitegrid')
plt.figure(figsize=(12,6))
plt.scatter(X_test,Y_test,color='blue',label="Data", s = 12)
plt.plot(X_test,lr.predict(X_test),color="red",label="Predicted Regression Line")
plt.xlabel("Trust", fontsize=15)
plt.ylabel("Happiness_Score", fontsize=15)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.legend()
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
```

Visualise and Examining data.

```
#Visualise and Examine data.

# DISTRIBUTION OF ALL NUMERIC DATA

plt.rcParams['figure.figsize'] = (15, 15)
df1 = finaldf16[['GDP_per_Capita', 'Family', 'Health', 'Freedom','Generosity', 'Trust']]
h = df1.hist(bins = 25, figsize = (16,16), xlabelsize = '10', ylabelsize = '10')
seabornInstance.despine(left = True, bottom = True)
[x.title.set_size(12) for x in h.ravel()];
[x.yaxis.tick_left() for x in h.ravel()]
```

Seaborn Instance Bar Plot

```
#SEABORNINSTANCE BARPLOT OF EACH VARIABLE

fig, axes = plt.subplots(nrows=3, ncols=2,constrained_layout=True,figsize=(10,10))
seabornInstance.barplot(x='GDP_per_Capita',y='Country',
                        data=finaldf.nlargest(10,'GDP'),
```

```
                ax=axes[0,0],palette="Blues_r")
seabornInstance.barplot(x='Health' ,y='Country',
                data=finaldf.nlargest(10,'Health'),
                ax=axes[0,1],palette='Blues_r')
seabornInstance.barplot(x='Happiness_Score' ,y='Country',
                data=finaldf.nlargest(10,'Score'),
                ax=axes[1,0],palette='Blues_r')
seabornInstance.barplot(x='Generosity' ,y='Country',
                data=finaldf.nlargest(10,'Generosity'),
                ax=axes[1,1],palette='Blues_r')
seabornInstance.barplot(x='Freedom' ,y='Country',
                data=finaldf.nlargest(10,'Freedom'),
                ax=axes[2,0],palette='Blues_r')
seabornInstance.barplot(x='Trust' ,y='Country',
                data=finaldf.nlargest(10,'Corruption'),
                ax=axes[2,1],palette='Blues_r')
```

Pearson Correlation Matrix

```
#Checking the Correlation Among Explanatory Variables using "PEARSON CORRELATION MATRIX".

mask = np.zeros_like(finaldf[usecols].corr(), dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(16, 12))
plt.title('Pearson Correlation Matrix',fontsize=25)
seabornInstance.heatmap(finaldf[usecols].corr(),          linewidths=0.25,vmax=0.7,square=True,cmap="Blues",
linecolor='w',annot=True,annot_kws={"size":8},mask=mask,cbar_kws={"shrink": .9});


#--- Visualising hidden relationships in data. ---

classifier_overrides = set()
df16_results = discover.discover(finaldf.drop(['target', 'target_n'],axis=1).sample(frac=1), classifier_overrides)


# ----- Using heat maps to visualise how features are clustered / vary over space. -----

fig, ax = plt.subplots(ncols=2,figsize=(24, 8))

seabornInstance.heatmap(df16_results.pivot(index = 'target', columns = 'feature', values =
'Happiness_Score').fillna(1).loc[finaldf.drop(['target', 'target_n'],axis = 1).columns,finaldf.drop(['target',
'target_n'],axis = 1).columns], annot=True, center = 0, ax = ax[0], vmin = -1, vmax = 1, cmap = "Blues")

seabornInstance.heatmap(df_results.pivot(index = 'target', columns = 'feature', values =
'score').fillna(1).loc[finaldf.drop(
                ['target', 'target_n'],axis=1).columns,finaldf.drop(['target', 'target_n'],axis=1).columns],
annot=True, center=0, ax=ax[1], vmin=-0.25, vmax=1, cmap="Blues_r")
plt.plot()
```

Creating a Model having all features.

```
# ----- Creating a Model having all features. -----

# --- MULTIPLE LINEAR REGRESSION 1 ---

train_data_dm,test_data_dm = train_test_split(finaldf16,train_size = 0.8,random_state=3)
independent_var = ['GDP_per_Capita','Family','Health','Freedom','Generosity','Trust']
complex_model_1 = LinearRegression()
complex_model_1.fit(train_data_dm[independent_var],train_data_dm['Happiness_Score'])
print('Intercept: {}'.format(complex_model_1.intercept_))
print('Coefficients: {}'.format(complex_model_1.coef_))
print('Happiness_Score = ',np.round(complex_model_1.intercept_,4),
    '+',np.round(complex_model_1.coef_[0],4),'* Family',
    '+',np.round(complex_model_1.coef_[1],4),'* GDP_per_Capita',
    '+',np.round(complex_model_1.coef_[2],4),'* Health',
    '+',np.round(complex_model_1.coef_[3],4),'* Freedom',
     '+',np.round(complex_model_1.coef_[4],4),'* Generosity',
    '+',np.round(complex_model_1.coef_[5],4),'* Trust')

pred = complex_model_1.predict(test_data_dm[independent_var])
rmsecm = float(format(np.sqrt(metrics.mean_squared_error(test_data_dm['Happiness_Score'],pred)),'.3f'))

rtrcm = float(format(complex_model_1.score(train_data_dm[independent_var],train_data_dm['Happiness_Score']),'.3f'))
artrcm = float(format(adjustedR2(complex_model_1.score(train_data_dm[independent_var],train_data_dm['Happiness_Score']),train_data_dm.shape[0],len(independent_var)),'.3f'))

rtecm = float(format(complex_model_1.score(test_data_dm[independent_var],test_data_dm['Happiness_Score']),'.3f'))
artecm = float(format(adjustedR2(complex_model_1.score(test_data_dm[independent_var],test_data['Happiness_Score']),test_data_dm.shape[0],len(independent_var)),'.3f'))
cv = float(format(cross_val_score(complex_model_1,finaldf16[independent_var],finaldf16['Happiness_Score'],cv=5).mean(),'.3f'))
r = evaluation.shape[0]
evaluation.loc[r] = ['Multiple Linear Regression-1','selected features',rmsecm,rtrcm,artrcm,rtecm,artecm,cv]
evaluation.sort_values(by = '5-Fold Cross Validation', ascending=False)
```

REFERENCE.

1. R Tutorial. Available at: https://www.tutorialspoint.com/r/index.htm (Accessed at 25th March 2022).

2. SQL Tutorial. Available at: https://www.w3schools.com/sql/ (Accessed at 26th March 2022).

3. SQL Server Tutorial. Available at: https://www.sqlservertutorial.net/ (Accessed at 26th March 2022).

4. R - XML Files. Available at: https://www.tutorialspoint.com/r/r_xml_files.htm (Accessed at 28th March 2022).

5. The University of Sheffield - Harvard referencing. Available at: https://www.librarydevelopment.group.shef.ac.uk/referencing/harvard.html#collapseReferenceList (Accessed at 28th March 2022).

6. WORLD HAPPINESS REPORT☺ || EDA || CLUSTERING||. Available at: https://www.kaggle.com/code/avnika22/world-happiness-report-eda-clustering/notebook (Accessed at 29th March 2022).

7. R Tutorial.Available at: https://www.w3schools.com/r/default.asp (Accessed at 29th March 2022).

APPENDIX I.

Available at: https://github.com/pkavi01/Project_BCM_World_Happiness_Report

(Attached and to be sent separately.)