

Name – Pritam Raj Hazarika

Email – pritam98h@gmail.com

SQL Project

Dataset - World Happiness Index

1) Which are the top 10 happiest countries in the given dataset?

```
SELECT Country, Happiness_Score FROM Happiness_Index ORDER BY Happiness_Score  
DESC LIMIT 10;
```

The screenshot shows a SQLite database interface with a dark theme. On the left, a sidebar lists the database structure: 'demo' database containing a 'Happiness_Index' table with columns like Country, Region, Happiness_Rank, Happiness_Score, Standard_Error, Economy_GDP_per..., Family, Health_Life_Expecta..., Freedom, Trust_Government..., Generosity, and Dystopia_Residual R... Below the sidebar, the main editor displays the SQL query: `SELECT Country, Happiness_Score FROM Happiness_Index ORDER BY Happiness_Score DESC LIMIT 10;`. The query results are shown in a table with two columns: 'Country' and 'Happiness_Score'. The top 10 results are: Switzerland (7.587), Iceland (7.561), Denmark (7.527), Norway (7.522), Canada (7.427), Finland (7.406), Netherlands (7.378), and Sweden (7.364). The interface also shows tabs for SQLite, SQLite.2, SQLite.3, and SQLite.4 at the top.

| Country | Happiness_Score |
|-------------|-----------------|
| Switzerland | 7.587 |
| Iceland | 7.561 |
| Denmark | 7.527 |
| Norway | 7.522 |
| Canada | 7.427 |
| Finland | 7.406 |
| Netherlands | 7.378 |
| Sweden | 7.364 |

2) What is the average happiness score across all countries and years?

```
SELECT AVG(happiness_score) FROM Happiness_Index;
```

The screenshot shows a SQLite database interface. On the left, a sidebar lists the database 'SQLite' and its tables: 'demo' and 'Happiness_Index'. The 'Happiness_Index' table is expanded, showing columns: Country (TEXT), Region (TEXT), Happiness_Rank (INT...), Happiness_Score (REAL), Standard_Error (REAL), Economy_GDP (REAL), Family (REAL), Life_Expectancy (REAL), Freedom (REAL), Trust_Government_C... (REAL), Generosity (REAL), and Dystopia_Residual R... (REAL). The main area displays the SQL query: `SELECT AVG(happiness_score) FROM Happiness_Index;`. Below the query, the result is shown as a single row with the value `5.37573417721519`.

3) Which are the bottom 10 Countries with the lowest Happiness Score?

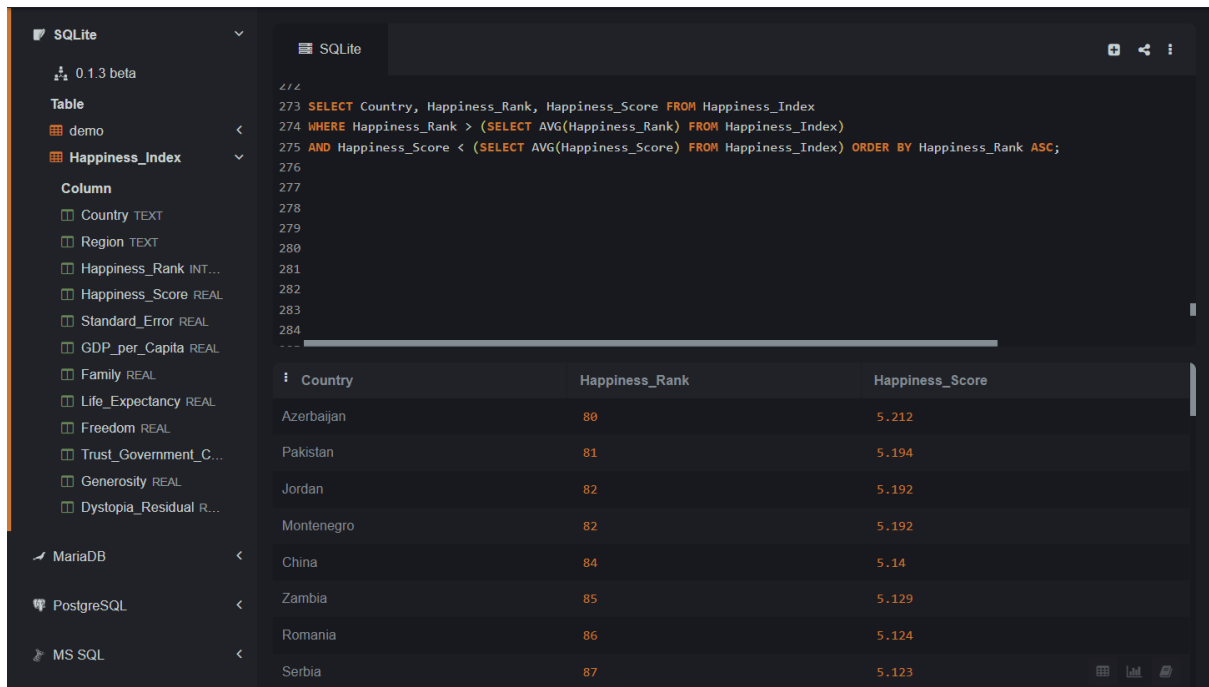
```
SELECT Country, Happiness_Rank, Happiness_Score FROM Happiness_Index ORDER BY Happiness_Rank DESC LIMIT 10;
```

The screenshot shows a SQLite database interface. On the left, a sidebar lists the database 'SQLite' and its tables: 'demo' and 'Happiness_Index'. The 'Happiness_Index' table is expanded, showing columns: Country (TEXT), Region (TEXT), Happiness_Rank (INT...), Happiness_Score (REAL), Standard_Error (REAL), GDP_per_Capita (REAL), Family (REAL), Life_Expectancy (REAL), Freedom (REAL), Trust_Government_C... (REAL), Generosity (REAL), and Dystopia_Residual R... (REAL). The main area displays the SQL query: `SELECT Country, Happiness_Rank, Happiness_Score FROM Happiness_Index ORDER BY Happiness_Rank DESC LIMIT 10;`. Below the query, the results are shown as a table with 3 columns: Country, Happiness_Rank, and Happiness_Score. The results are as follows:

| Country | Happiness_Rank | Happiness_Score |
|--------------|----------------|-----------------|
| Togo | 158 | 2.839 |
| Burundi | 157 | 2.905 |
| Syria | 156 | 3.006 |
| Benin | 155 | 3.34 |
| Rwanda | 154 | 3.465 |
| Afghanistan | 153 | 3.575 |
| Burkina Faso | 152 | 3.587 |
| Ivory Coast | 151 | 3.655 |

- 4) Which are the countries with Happiness Rank and Happiness Score below the Global Average?

SELECT Country, Happiness_Rank, Happiness_Score FROM Happiness_Index WHERE Happiness_Rank > (SELECT AVG(Happiness_Rank) FROM Happiness_Index) AND Happiness_Score < (SELECT AVG(Happiness_Score) FROM Happiness_Index) ORDER BY Happiness_Rank ASC;

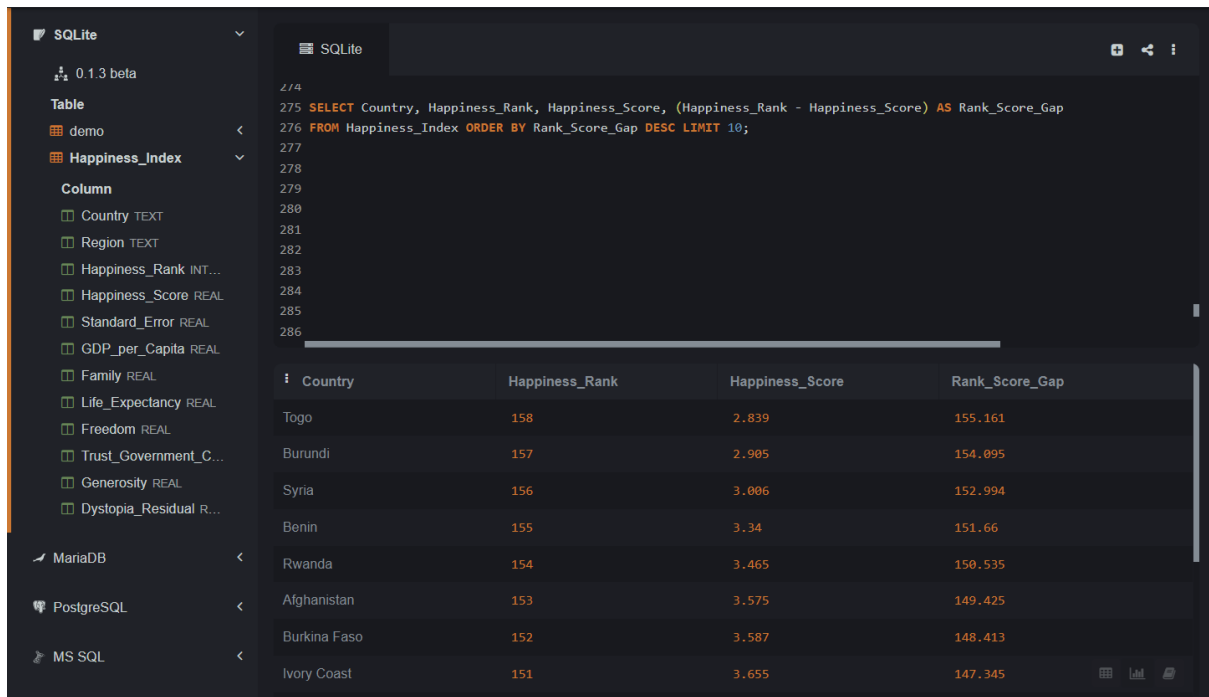


The screenshot shows a SQLite database interface. On the left, a sidebar lists the database 'demo' and its tables, including 'Happiness_Index'. The main area displays a SQL query and its results. The query is:
SELECT Country, Happiness_Rank, Happiness_Score FROM Happiness_Index WHERE Happiness_Rank > (SELECT AVG(Happiness_Rank) FROM Happiness_Index) AND Happiness_Score < (SELECT AVG(Happiness_Score) FROM Happiness_Index) ORDER BY Happiness_Rank ASC;
The results are shown in a table with three columns: Country, Happiness_Rank, and Happiness_Score. The data is as follows:

| Country | Happiness_Rank | Happiness_Score |
|------------|----------------|-----------------|
| Azerbaijan | 80 | 5.212 |
| Pakistan | 81 | 5.194 |
| Jordan | 82 | 5.192 |
| Montenegro | 82 | 5.192 |
| China | 84 | 5.14 |
| Zambia | 85 | 5.129 |
| Romania | 86 | 5.124 |
| Serbia | 87 | 5.123 |

5) Which are the countries with significant gaps between Happiness Rank and Happiness Score?

```
SELECT Country, Happiness_Rank, Happiness_Score, (Happiness_Rank - Happiness_Score) AS Rank_Score_Gap FROM Happiness_Index ORDER BY Rank_Score_Gap DESC LIMIT 10;
```



The screenshot shows a SQLite database interface. On the left, a sidebar lists the database 'demo' and its table 'Happiness_Index'. The table's columns are listed: Country (TEXT), Region (TEXT), Happiness_Rank (INTEGER), Happiness_Score (REAL), Standard_Error (REAL), GDP_per_Capita (REAL), Family (REAL), Life_Expectancy (REAL), Freedom (REAL), Trust_Government_C... (REAL), Generosity (REAL), and Dystopia_Residual R... (REAL). The main area displays the SQL query: `SELECT Country, Happiness_Rank, Happiness_Score, (Happiness_Rank - Happiness_Score) AS Rank_Score_Gap FROM Happiness_Index ORDER BY Rank_Score_Gap DESC LIMIT 10;`. Below the query, the results are shown in a table with 4 columns: Country, Happiness_Rank, Happiness_Score, and Rank_Score_Gap. The results list the top 10 countries with the largest gaps between their rank and score.

| Country | Happiness_Rank | Happiness_Score | Rank_Score_Gap |
|--------------|----------------|-----------------|----------------|
| Togo | 158 | 2.839 | 155.161 |
| Burundi | 157 | 2.905 | 154.095 |
| Syria | 156 | 3.006 | 152.994 |
| Benin | 155 | 3.34 | 151.66 |
| Rwanda | 154 | 3.465 | 150.535 |
| Afghanistan | 153 | 3.575 | 149.425 |
| Burkina Faso | 152 | 3.587 | 148.413 |
| Ivory Coast | 151 | 3.655 | 147.345 |

6) Find the countries with the maximum and minimum happiness scores in the dataset.

SELECT country, Happiness_Score FROM Happiness_Index ORDER BY Happiness_Score DESC LIMIT 1;

SELECT Country, Happiness_Score FROM Happiness_Index ORDER BY Happiness_Score ASC LIMIT 1;

The screenshot shows the SQLite IDE interface. On the left, the 'Happiness_Index' table is selected, showing columns: Country (TEXT), Region (TEXT), Happiness_Rank (INTEGER), Happiness_Score (REAL), Standard_Error (REAL), Economy_(GDP_per... (REAL), Family (REAL), Health_(Life_Expecta... (REAL), Freedom (REAL), Trust_(Government_... (REAL), Generosity (REAL), and Dystopia_Residual R... (REAL). The main editor displays the SQL query: `SELECT country, Happiness_Score FROM Happiness_Index ORDER BY Happiness_Score DESC LIMIT 1;`. The results pane at the bottom shows a single row with the country 'Switzerland' and a happiness score of 7.587.

| Country | Happiness_Score |
|-------------|-----------------|
| Switzerland | 7.587 |

The screenshot shows the SQLite IDE interface. On the left, the 'Happiness_Index' table is selected, showing the same columns as the previous screenshot. The main editor displays the SQL query: `SELECT Country, Happiness_Score FROM Happiness_Index ORDER BY Happiness_Score ASC LIMIT 1;`. The results pane at the bottom shows a single row with the country 'Togo' and a happiness score of 2.839.

| Country | Happiness_Score |
|---------|-----------------|
| Togo | 2.839 |

7) Which regions have the highest average happiness score?

SELECT Region, AVG(Happiness_Score) AS Avg_Happiness_Score FROM Happiness_Index GROUP BY Region ORDER BY Avg_Happiness_Score DESC;

The screenshot shows a database client interface with a sidebar on the left containing a tree view of databases (SQLite, MariaDB, PostgreSQL, MS SQL) and a table named 'Happiness_Index'. The main pane displays the SQL query and its results. The query is: `SELECT Region, AVG(Happiness_Score) AS Avg_Happiness_Score FROM Happiness_Index GROUP BY Region ORDER BY Avg_Happiness_Score DESC;` The results table has two columns: 'Region' and 'Avg_Happiness_Score'. The data is as follows:

| Region | Avg_Happiness_Score |
|---------------------------------|---------------------|
| Australia and New Zealand | 7.285 |
| North America | 7.273 |
| Western Europe | 6.689619047619048 |
| Latin America and Caribbean | 6.144681818181818 |
| Eastern Asia | 5.626166666666666 |
| Middle East and Northern Africa | 5.406899999999999 |
| Central and Eastern Europe | 5.332931034482758 |
| Southeastern Asia | 5.317444444444444 |

8) What countries have high freedom scores < 0.6 and high happiness greater than 7?

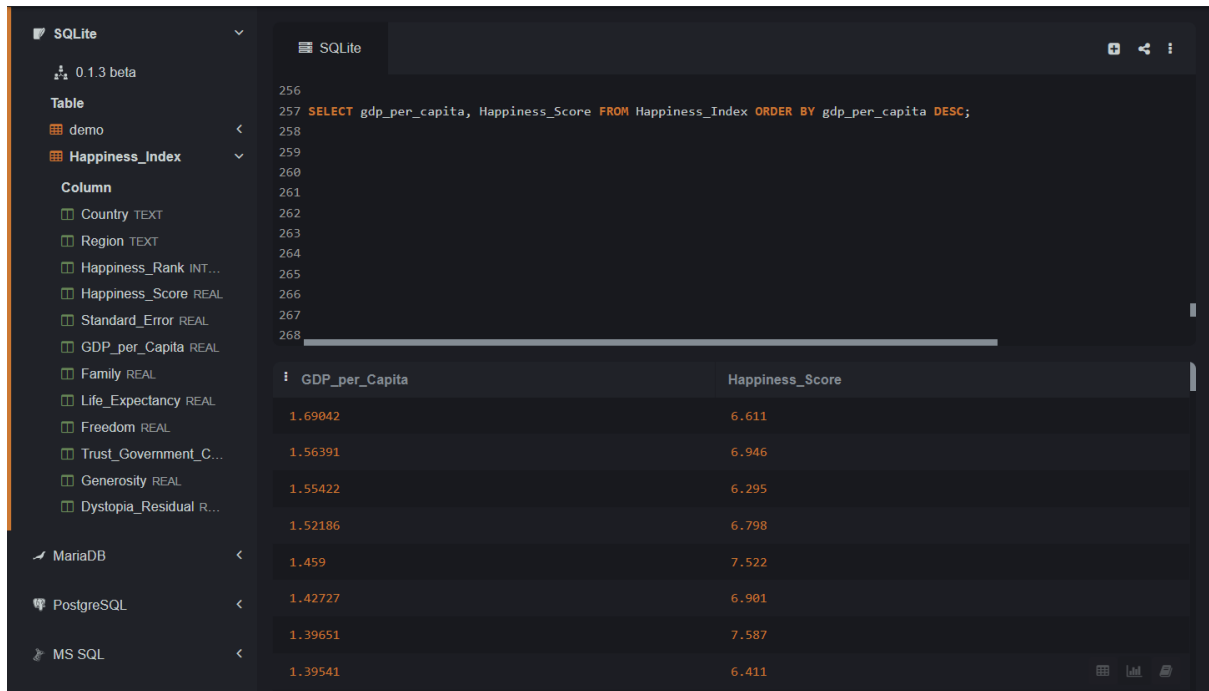
SELECT Country, Happiness_Score, Freedom FROM Happiness_Index WHERE Freedom > 0.6 AND Happiness_Score > 7 ORDER BY Happiness_Score DESC;

The screenshot shows the same database client interface with a different SQL query. The query is: `SELECT Country, Happiness_Score, Freedom FROM Happiness_Index WHERE Freedom > 0.6 AND Happiness_Score > 7 ORDER BY Happiness_Score DESC;` The results table has three columns: 'Country', 'Happiness_Score', and 'Freedom'. The data is as follows:

| Country | Happiness_Score | Freedom |
|-------------|-----------------|---------|
| Switzerland | 7.587 | 0.66557 |
| Iceland | 7.561 | 0.62877 |
| Denmark | 7.527 | 0.64938 |
| Norway | 7.522 | 0.66973 |
| Canada | 7.427 | 0.63297 |
| Finland | 7.406 | 0.64169 |
| Netherlands | 7.378 | 0.61576 |
| Sweden | 7.364 | 0.6598 |

9) Find the correlation between GDP per capita and happiness score.

```
SELECT gdp_per_capita, Happiness_Score FROM Happiness_Index ORDER BY  
gdp_per_capita DESC;
```

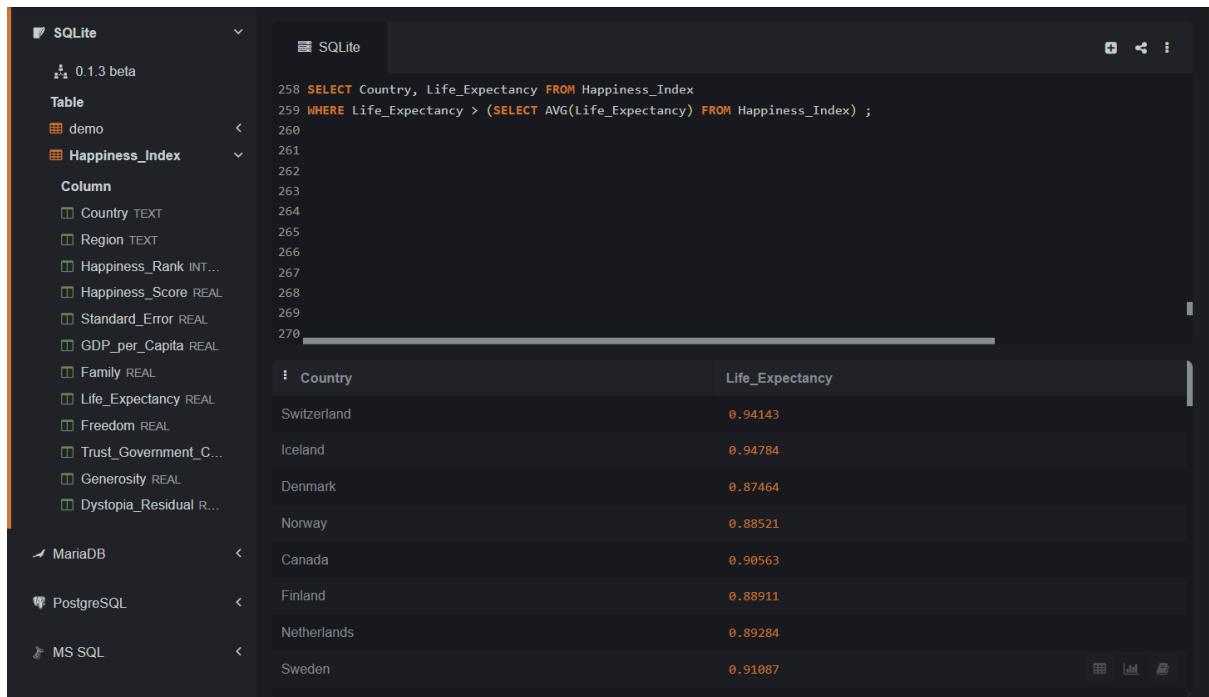


The screenshot shows a SQLite database interface. On the left, a sidebar lists the database structure: SQLite 0.1.3 beta, Table demo, and Happiness_Index with columns: Country TEXT, Region TEXT, Happiness_Rank INT..., Happiness_Score REAL, Standard_Error REAL, GDP_per_Capita REAL, Family REAL, Life_Expectancy REAL, Freedom REAL, Trust_Government_C..., Generosity REAL, and Dystopia_Residual R... The main area displays the SQL query: `SELECT gdp_per_capita, Happiness_Score FROM Happiness_Index ORDER BY gdp_per_capita DESC;` and the resulting table.

| GDP_per_Capita | Happiness_Score |
|----------------|-----------------|
| 1.69842 | 6.611 |
| 1.56391 | 6.946 |
| 1.55422 | 6.295 |
| 1.52186 | 6.798 |
| 1.459 | 7.522 |
| 1.42727 | 6.901 |
| 1.39651 | 7.587 |
| 1.39541 | 6.411 |

10) Which are the countries with Life Expectancy above average?

```
SELECT Country, Life_Expectancy FROM Happiness_Index WHERE Life_Expectancy > (SELECT AVG(Life_Expectancy) FROM Happiness_Index);
```

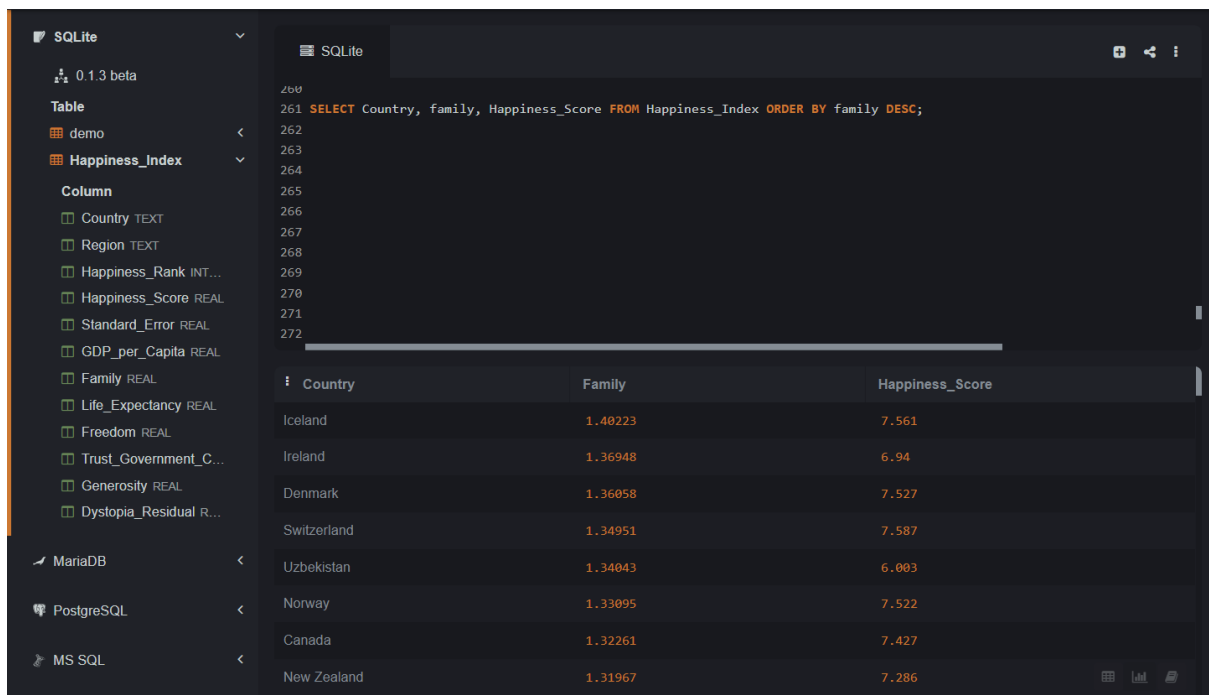


The screenshot shows the SQLite IDE interface. On the left, a sidebar lists the database structure: SQLite (0.1.3 beta), Tables (demo, Happiness_Index), and Columns (Country TEXT, Region TEXT, Happiness_Rank INT..., Happiness_Score REAL, Standard_Error REAL, GDP_per_Capita REAL, Family REAL, Life_Expectancy REAL, Freedom REAL, Trust_Government_C..., Generosity REAL, Dystopia_Residual R...). The main editor displays the SQL query: `SELECT Country, Life_Expectancy FROM Happiness_Index WHERE Life_Expectancy > (SELECT AVG(Life_Expectancy) FROM Happiness_Index);`. Below the query, the results are shown in a table with two columns: Country and Life_Expectancy. The results are sorted by Life_Expectancy in descending order.

| Country | Life_Expectancy |
|-------------|-----------------|
| Switzerland | 0.94143 |
| Iceland | 0.94784 |
| Denmark | 0.87464 |
| Norway | 0.88521 |
| Canada | 0.90563 |
| Finland | 0.88911 |
| Netherlands | 0.89284 |
| Sweden | 0.91087 |

11) Find the Correlation Between Family Support and Happiness Score.

```
SELECT Country, family, Happiness_Score FROM Happiness_Index ORDER BY family DESC;
```



The screenshot shows the SQLite IDE interface. On the left, the same sidebar as the previous image is visible. The main editor displays the SQL query: `SELECT Country, family, Happiness_Score FROM Happiness_Index ORDER BY family DESC;`. Below the query, the results are shown in a table with three columns: Country, Family, and Happiness_Score. The results are sorted by Family in descending order.

| Country | Family | Happiness_Score |
|-------------|---------|-----------------|
| Iceland | 1.40223 | 7.561 |
| Ireland | 1.36948 | 6.94 |
| Denmark | 1.36058 | 7.527 |
| Switzerland | 1.34951 | 7.587 |
| Uzbekistan | 1.34043 | 6.003 |
| Norway | 1.33095 | 7.522 |
| Canada | 1.32261 | 7.427 |
| New Zealand | 1.31967 | 7.286 |

12) How does Freedom compare to other factors (like corruption or generosity) across countries?

SELECT country, freedom, corruption, generosity FROM Happiness_Index ORDER BY freedom DESC LIMIT 10;

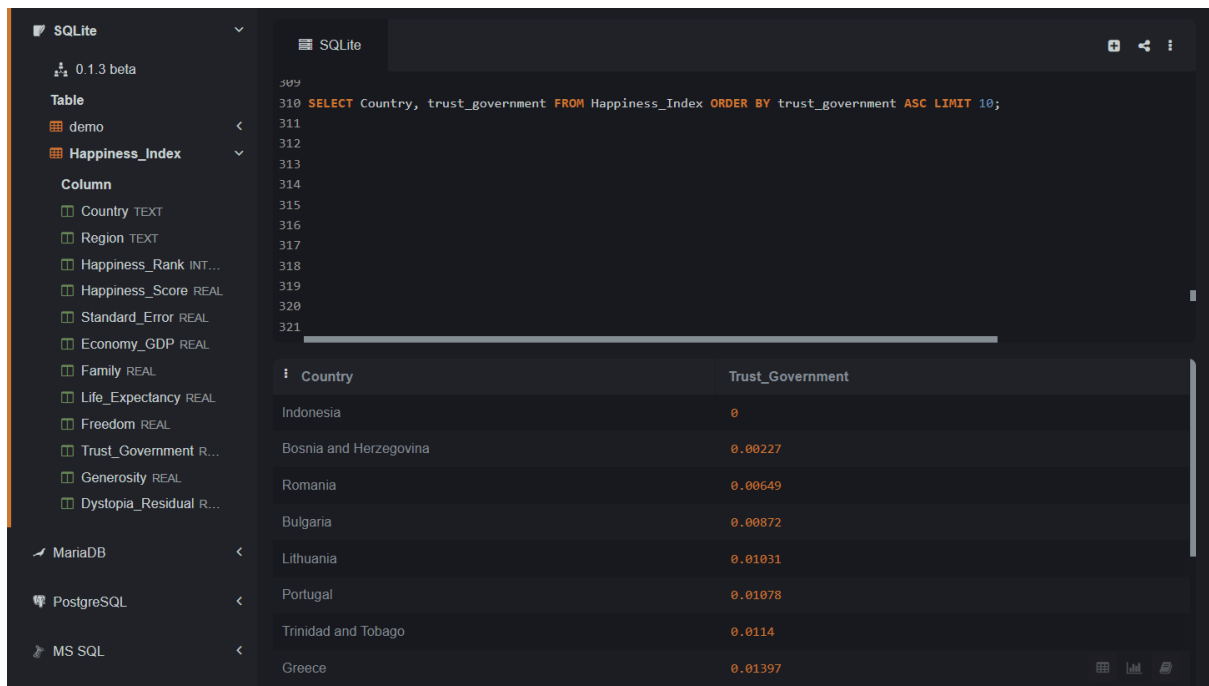


The screenshot shows the SQLite IDE interface. On the left, a sidebar lists the database structure: SQLite (0.1.3 beta) with a table named Happiness_Index. The table columns are: Country (TEXT), Region (TEXT), Happiness_Rank (INTEGER), Happiness_Score (REAL), Standard_Error (REAL), GDP_per_Capita (REAL), Family (REAL), Life_Expectancy (REAL), Freedom (REAL), Trust_Government_Corruption (REAL), Generosity (REAL), and Dystopia_Residual (REAL). The main editor displays the SQL query: `SELECT Country, Generosity, GDP_per_Capita FROM Happiness_Index ORDER BY Generosity DESC;`. The results pane shows the following data:

| Country | Generosity | GDP_per_Capita |
|--------------------|------------|----------------|
| Myanmar | 0.79588 | 0.27108 |
| Thailand | 0.5763 | 0.9669 |
| United Kingdom | 0.51912 | 1.26637 |
| Malta | 0.51752 | 1.2074 |
| Indonesia | 0.51535 | 0.82827 |
| Somaliiland region | 0.50318 | 0.18847 |
| Bhutan | 0.47998 | 0.77042 |
| Netherlands | 0.4761 | 1.32944 |

13) Which are the top 10 countries with the Lowest Government Trust?

SELECT Country, trust_government_corruption FROM Happiness_Index ORDER BY trust_government ASC LIMIT 10;

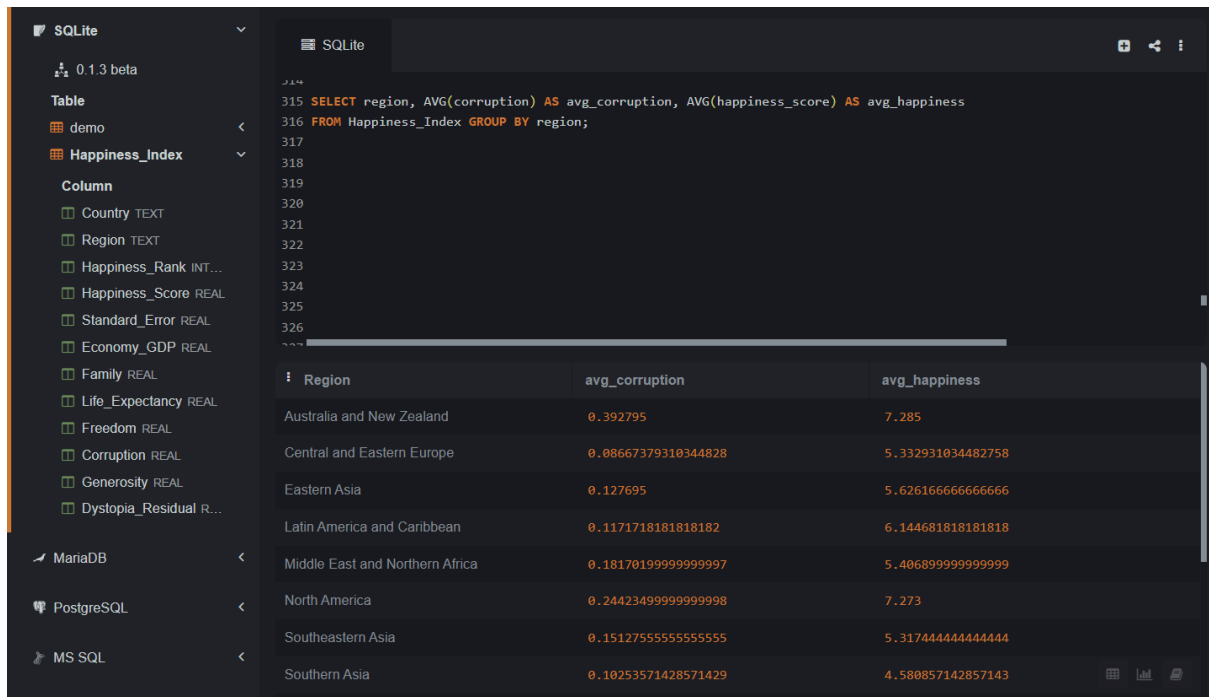


The screenshot shows the SQLite IDE interface. On the left, a sidebar lists the database structure: SQLite (0.1.3 beta) with a table named Happiness_Index. The table columns are: Country (TEXT), Region (TEXT), Happiness_Rank (INTEGER), Happiness_Score (REAL), Standard_Error (REAL), Economy_GDP (REAL), Family (REAL), Life_Expectancy (REAL), Freedom (REAL), Trust_Government (REAL), Generosity (REAL), and Dystopia_Residual (REAL). The main editor displays the SQL query: `SELECT Country, trust_government FROM Happiness_Index ORDER BY trust_government ASC LIMIT 10;`. The results pane shows the following data:

| Country | Trust_Government |
|------------------------|------------------|
| Indonesia | 0 |
| Bosnia and Herzegovina | 0.00227 |
| Romania | 0.00649 |
| Bulgaria | 0.00872 |
| Lithuania | 0.01031 |
| Portugal | 0.01078 |
| Trinidad and Tobago | 0.0114 |
| Greece | 0.01397 |

14) How does the level of corruption affect the happiness score in different regions?

SELECT region, AVG(corruption) AS avg_corruption, AVG(happiness_score) AS avg_happiness FROM Happiness_Index GROUP BY region;

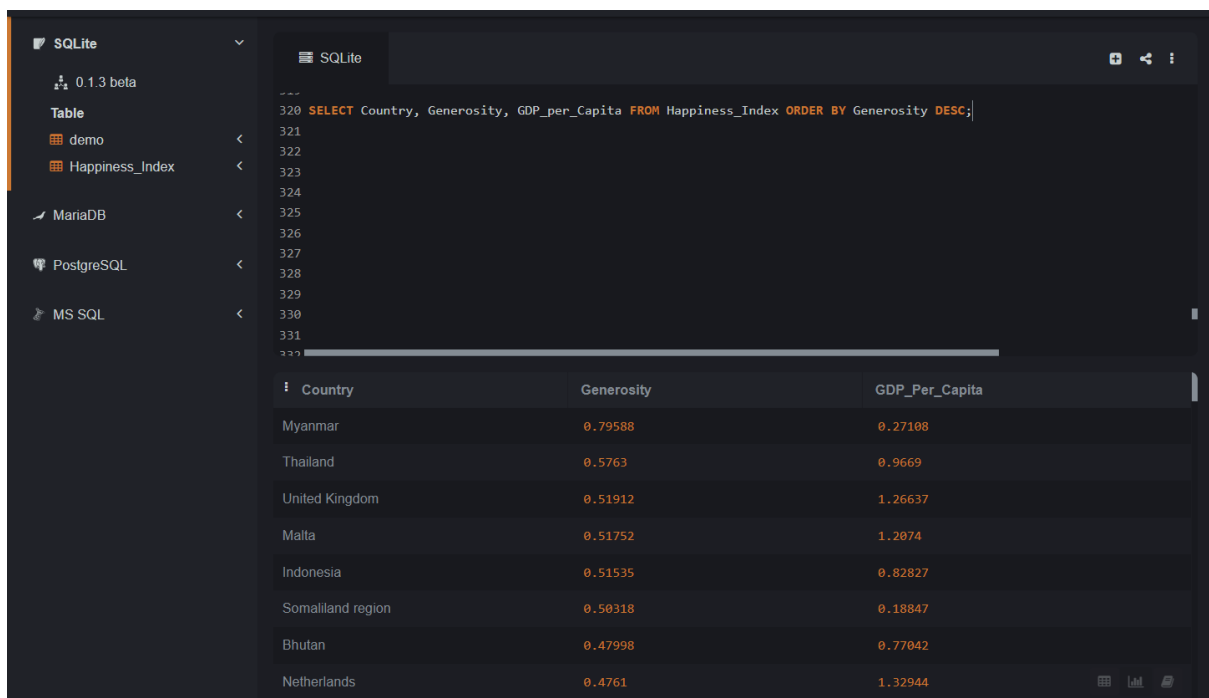


The screenshot shows a SQLite database interface with a query editor on the left and a results table on the right. The query is: `SELECT region, AVG(corruption) AS avg_corruption, AVG(happiness_score) AS avg_happiness FROM Happiness_Index GROUP BY region;` The results table has three columns: Region, avg_corruption, and avg_happiness. The data is grouped by region, showing the average corruption and average happiness score for each region.

| Region | avg_corruption | avg_happiness |
|---------------------------------|---------------------|-------------------|
| Australia and New Zealand | 0.392795 | 7.285 |
| Central and Eastern Europe | 0.08667379310344828 | 5.332931034482758 |
| Eastern Asia | 0.127695 | 5.626166666666666 |
| Latin America and Caribbean | 0.1171718181818182 | 6.144681818181818 |
| Middle East and Northern Africa | 0.18170199999999997 | 5.406899999999999 |
| North America | 0.24423499999999998 | 7.273 |
| Southeastern Asia | 0.15127555555555555 | 5.317444444444444 |
| Southern Asia | 0.10253571428571429 | 4.580857142857143 |

15) Find the relationship between generosity and economic wealth.

SELECT Country, Generosity, GDP_per_Capita FROM Happiness_Index ORDER BY Generosity DESC;

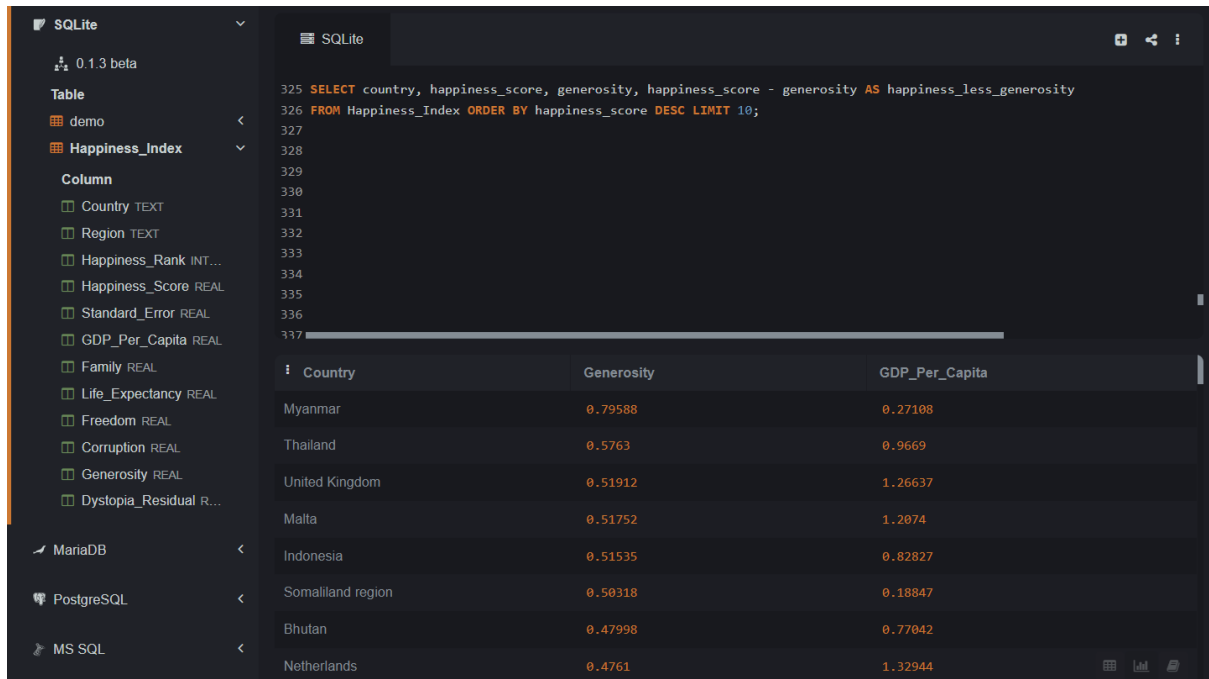


The screenshot shows a SQLite database interface with a query editor on the left and a results table on the right. The query is: `SELECT Country, Generosity, GDP_per_Capita FROM Happiness_Index ORDER BY Generosity DESC;` The results table has three columns: Country, Generosity, and GDP_Per_Capita. The data is ordered by generosity in descending order, showing the relationship between generosity and economic wealth for different countries.

| Country | Generosity | GDP_Per_Capita |
|-------------------|------------|----------------|
| Myanmar | 0.79588 | 0.27108 |
| Thailand | 0.5763 | 0.9669 |
| United Kingdom | 0.51912 | 1.26637 |
| Malta | 0.51752 | 1.2074 |
| Indonesia | 0.51535 | 0.82827 |
| Somaliland region | 0.50318 | 0.18847 |
| Bhutan | 0.47998 | 0.77042 |
| Netherlands | 0.4761 | 1.32944 |

16) For the countries with the highest happiness scores, how much does Generosity contribute to the happiness score?

SELECT country, happiness_score, generosity, happiness_score - generosity AS happiness_less_generosity FROM Happiness_Index ORDER BY happiness_score DESC LIMIT 10;

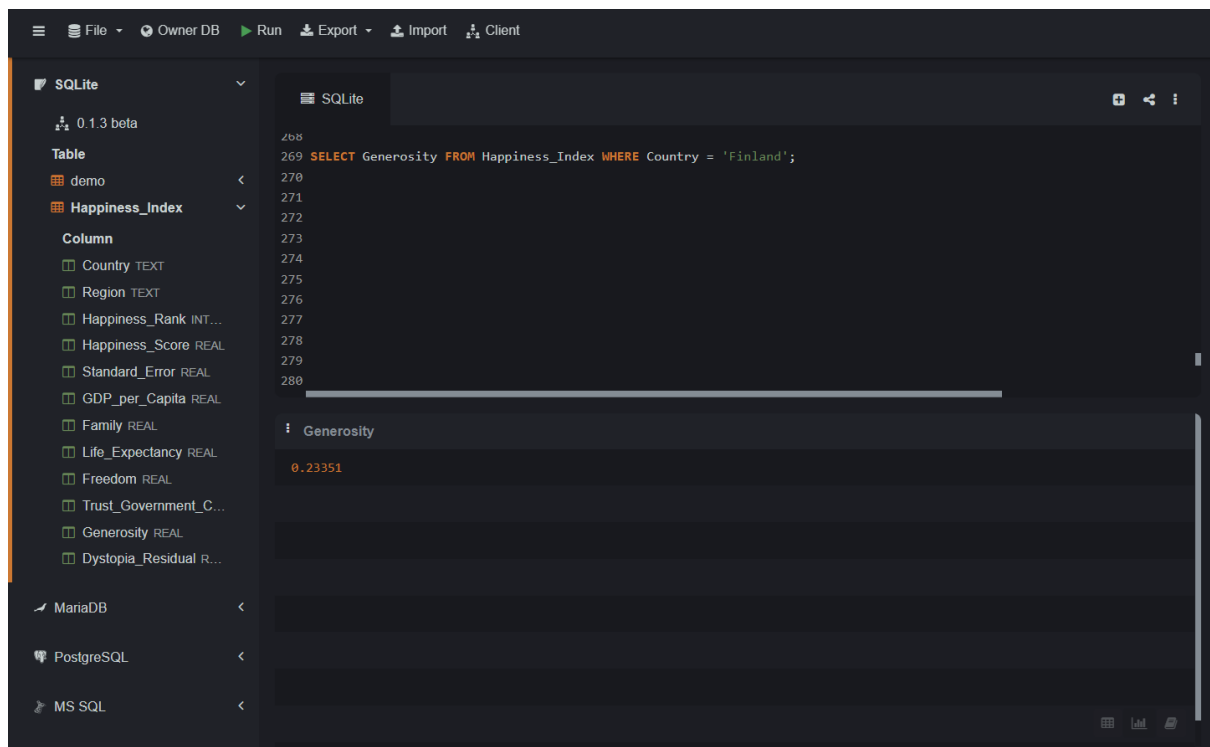


The screenshot shows a SQLite database interface. On the left, a sidebar lists the database 'demo' and its table 'Happiness_Index'. The table's columns are listed: Country (TEXT), Region (TEXT), Happiness_Rank (INTEGER), Happiness_Score (REAL), Standard_Error (REAL), GDP_Per_Capita (REAL), Family (REAL), Life_Expectancy (REAL), Freedom (REAL), Corruption (REAL), Generosity (REAL), and Dystopia_Residual (REAL). The main area displays the SQL query: `SELECT country, happiness_score, generosity, happiness_score - generosity AS happiness_less_generosity FROM Happiness_Index ORDER BY happiness_score DESC LIMIT 10;`. Below the query, the results are shown in a table with three columns: Country, Generosity, and GDP_Per_Capita. The results list the top 10 countries by happiness score.

| Country | Generosity | GDP_Per_Capita |
|-------------------|------------|----------------|
| Myanmar | 0.79588 | 0.27108 |
| Thailand | 0.5763 | 0.9669 |
| United Kingdom | 0.51912 | 1.26637 |
| Malta | 0.51752 | 1.2074 |
| Indonesia | 0.51535 | 0.82827 |
| Somaliland region | 0.50318 | 0.18847 |
| Bhutan | 0.47998 | 0.77042 |
| Netherlands | 0.4761 | 1.32944 |

17) Find the Generosity in Finland.

```
SELECT Generosity FROM Happiness_Index WHERE Country = 'Finland';
```



18) Which are the countries with High Happiness and Low Dystopia Residual?

SELECT Country, Happiness_Score, Dystopia_Residual FROM Happiness_Index WHERE Dystopia_Residual < 1.0 ORDER BY Happiness_Score DESC;

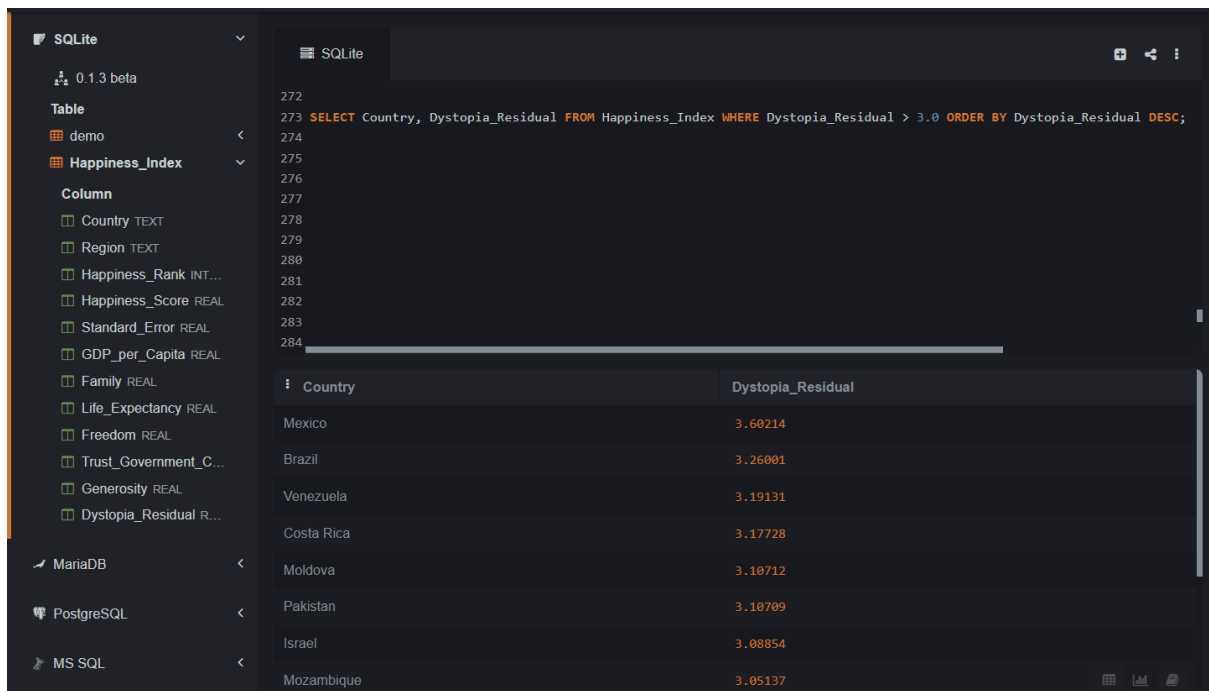


The screenshot shows a SQLite database interface with a query editor on the left and a results table on the right. The query is: `SELECT Country, Happiness_Score, Dystopia_Residual FROM Happiness_Index WHERE Dystopia_Residual < 1.0 ORDER BY Happiness_Score DESC;` The results table displays the following data:

| Country | Happiness_Score | Dystopia_Residual |
|-----------|-----------------|-------------------|
| Hong Kong | 5.474 | 0.65429 |
| Sri Lanka | 4.271 | 0.67108 |
| Bulgaria | 4.218 | 0.89991 |
| Gabon | 3.896 | 0.99895 |
| Cambodia | 3.819 | 0.98195 |
| Rwanda | 3.465 | 0.67042 |
| Syria | 3.006 | 0.32858 |

19) Which are the countries with dystopia_residual above threshold 3.0?

SELECT Country, Dystopia_Residual FROM Happiness_Index WHERE Dystopia_Residual > 3.0 ORDER BY Dystopia_Residual DESC;



The screenshot shows a SQLite database interface with a query editor on the left and a results table on the right. The query is: `SELECT Country, Dystopia_Residual FROM Happiness_Index WHERE Dystopia_Residual > 3.0 ORDER BY Dystopia_Residual DESC;` The results table displays the following data:

| Country | Dystopia_Residual |
|------------|-------------------|
| Mexico | 3.60214 |
| Brazil | 3.26001 |
| Venezuela | 3.19131 |
| Costa Rica | 3.17728 |
| Moldova | 3.10712 |
| Pakistan | 3.10709 |
| Israel | 3.08854 |
| Mozambique | 3.05137 |

20) List the countries with the highest negative Dystopia Residuals.

```
SELECT country, dystopia_residual, happiness_score FROM Happiness_Index WHERE  
dystopia_residual < 1 ORDER BY dystopia_residual ASC LIMIT 10;
```

The screenshot shows the SQLiteonline.com web interface. The left sidebar displays the database structure for 'demo', including the 'Happiness_Index' table with columns like Country, Region, Happiness_Rank, Happiness_Score, Standard_Error, GDP_per_capita, Family, Life_Expectancy, Freedom, Corruption, Generosity, and Dystopia_Residual. The main editor contains the following SQL query:

```
291 /*  
292 --SELECT country FROM Happiness_Index  
293 --WHERE gdp_per_capita IS NULL OR family IS NULL OR life_expectancy IS NULL;  
294  
295 SELECT country, dystopia_residual, happiness_score FROM Happiness_Index  
296 WHERE dystopia_residual < 1 ORDER BY dystopia_residual ASC LIMIT 10;  
297  
298  
299  
300  
301  
302  
303
```

The results are displayed in a table with the following data:

| Country | Dystopia_Residual | Happiness_Score |
|-----------|-------------------|-----------------|
| Syria | 0.32858 | 3.006 |
| Hong Kong | 0.65429 | 5.474 |
| Rwanda | 0.67042 | 3.465 |
| Sri Lanka | 0.67108 | 4.271 |
| Bulgaria | 0.89991 | 4.218 |
| Cambodia | 0.90195 | 3.819 |
| Gabon | 0.99895 | 3.896 |

The right sidebar shows the 'History' tab with three recent queries, all of which are the same as the one in the main editor.