**CSC263 Database Systems**

**Midterm Examination**

**Hands-on (Max Points: 40 pts)**

NAME: \_\_\_Steven Moody\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ SCORE: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/40

Create a “**mini\_world**” database by using the script provided. The schema for the “mini\_world” database is shown below.

**COUNTRY (name, continent, population, area, gdp, life\_exp)**

**CITY (name, country, population, capital)**

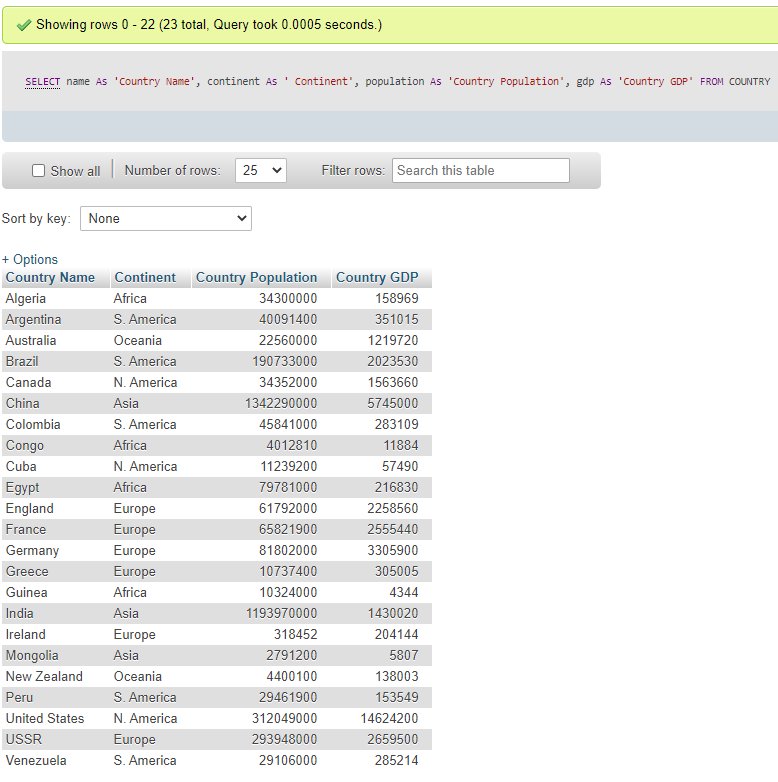
**RIVER (name, origin, total\_length, drainage\_area, discharge)**

Write and execute queries to accomplish the following tasks. Show SQL queries you used and show screenshots of the query execution results. You must write queries that follow the specified headings to display the results.

1. **List complete information of every country that is in the database.**

**Query Used “SELECT name As 'Country Name', continent As ' Continent', population As 'Country Population', gdp As 'Country GDP' FROM COUNTRY”**

|  |  |  |  |
| --- | --- | --- | --- |
| Country Name | Continent | Country Population | Country GDP |



**2. In alphabetical order of the name, list the complete information of those countries that are in Asia.**

**Query Used “SELECT name As 'Country Name', continent As ' Continent', population As 'Country Population', gdp As 'Country GDP' FROM COUNTRY WHERE continent LIKE '%Asia%' ORDER BY name ASC”**

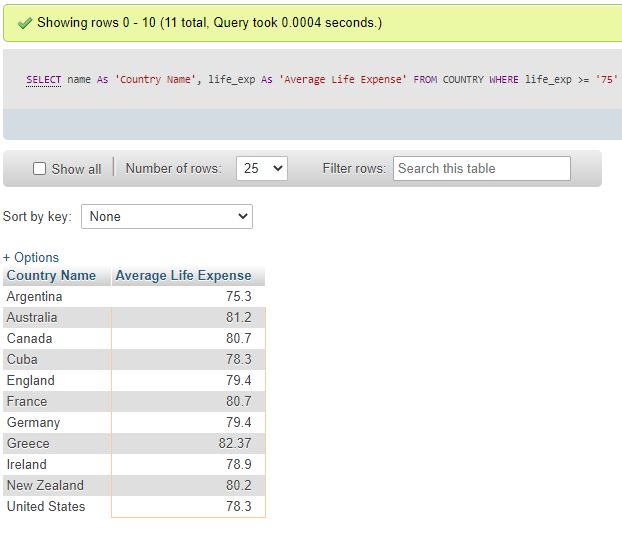
|  |  |  |  |
| --- | --- | --- | --- |
| Country Name | Continent | Country Population | Country GDP |



1. **In descending order of life expense, list complete information of those countries that have average life expense of 75 or higher.**

**Query Used: “SELECT name As 'Country Name', life\_exp As 'Average Life Expense' FROM COUNTRY WHERE life\_exp >= '75' “**

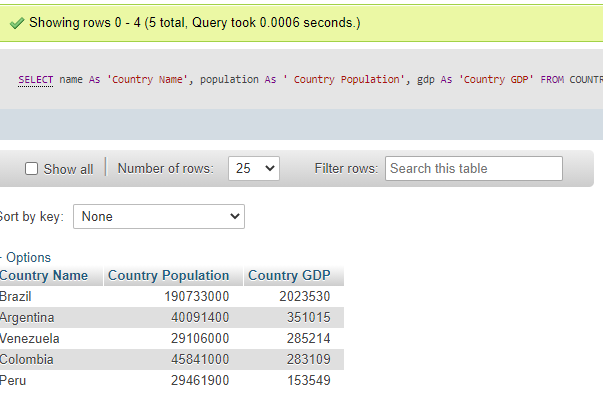
|  |  |
| --- | --- |
| Country Name | Average Life Expense |



1. **In descending order of the GDP, list those South American countries by their names and GDP's.**

**Query Used: “SELECT name As 'Country Name', population As ' Country Population', gdp As 'Country GDP' FROM COUNTRY WHERE continent LIKE '%S. America%' ORDER BY gdp DESC “**

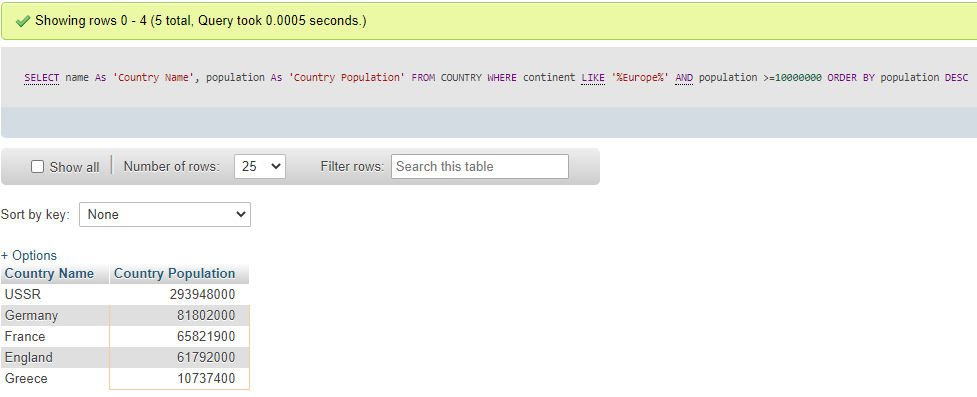
|  |  |  |
| --- | --- | --- |
| Country Name | Country Population | Country GDP |



1. **In descending order of the population, list all European countries that have population over 10 Million.**

**“Query Used “SELECT name As 'Country Name', population As 'Country Population' FROM COUNTRY WHERE continent LIKE '%Europe%' AND population >10000000 ORDER BY population DESC”**

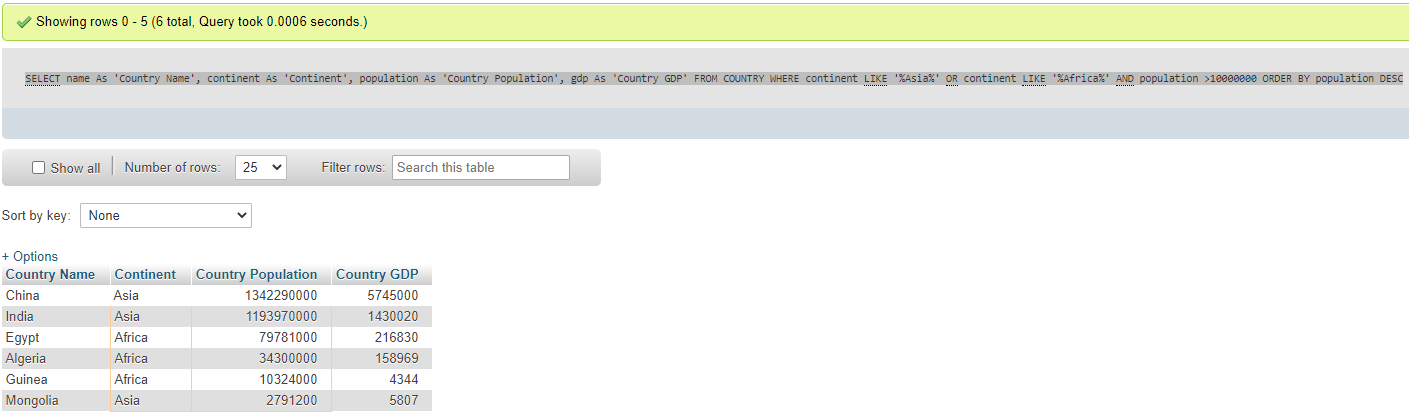
|  |  |
| --- | --- |
| Country Name | Country Population |



1. **In descending order of the GDP, list the African and Asian countries that have a population of over 10 Million.**

**Query Used “SELECT name As 'Country Name', continent As 'Continent', population As 'Country Population', gdp As 'Country GDP' FROM COUNTRY WHERE continent LIKE '%Asia%' OR continent LIKE '%Africa%' AND population >10000000 ORDER BY population DESC “**

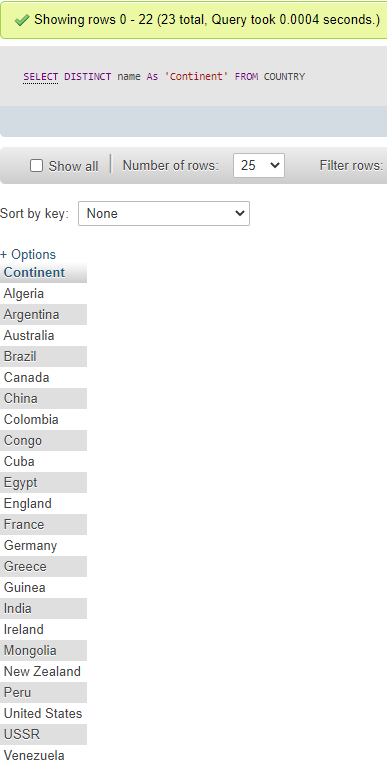
|  |  |  |  |
| --- | --- | --- | --- |
| Country Name | Continent | Country Population | Country GDP |



1. **List all continent names (without duplications).**

**Query Used “SELECT DISTINCT name As 'Continent' FROM COUNTRY”**

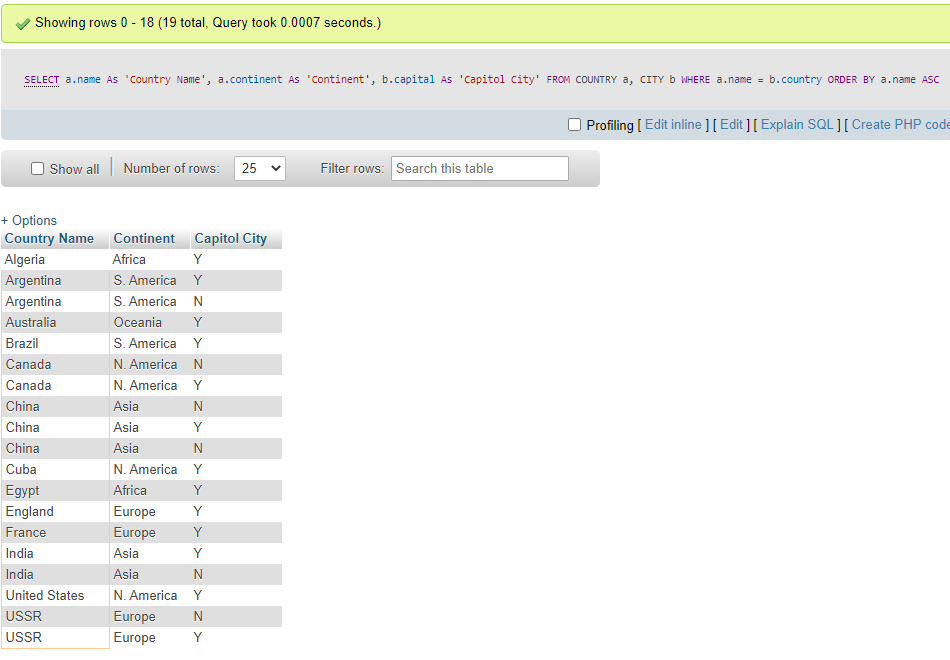
|  |
| --- |
| Continent |



1. **In ascending order, list all countries, the continents they are in and their capital city.**

**Query Used “ SELECT a.name As 'Country Name', a.continent As 'Continent', b.capital As 'Capitol City' FROM COUNTRY a, CITY b WHERE a.name = b.country ORDER BY a.name ASC”**

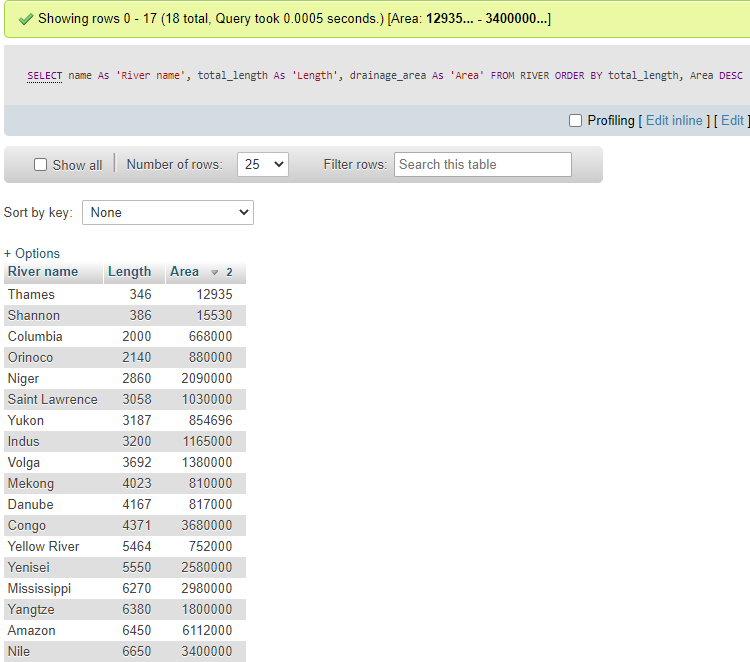
|  |  |  |
| --- | --- | --- |
| Country Name | Continent | Capital City |



1. **In descending order of first the lengths and secondly the areas, list all rivers that are longer than 3000 kilometers.**

**Query Used “SELECT name As 'River name', total\_length As 'Length', drainage\_area As 'Area' FROM RIVER ORDER BY total\_length, Area DESC”**

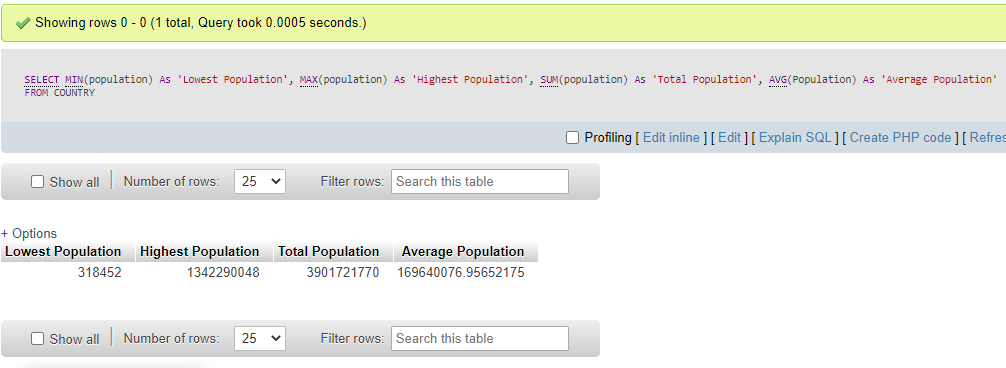
|  |  |  |
| --- | --- | --- |
| River Name | Length | Area |



1. **Write ONE query to display the smallest population of a country, the largest population of a country, the total population and the average population in world (of course, the “mini-world”).**

**Query Used “SELECT MIN(population) As 'Lowest Population', MAX(population) As 'Highest Population', SUM(population) As 'Total Population', AVG(Population) As 'Average Population' FROM COUNTRY”**

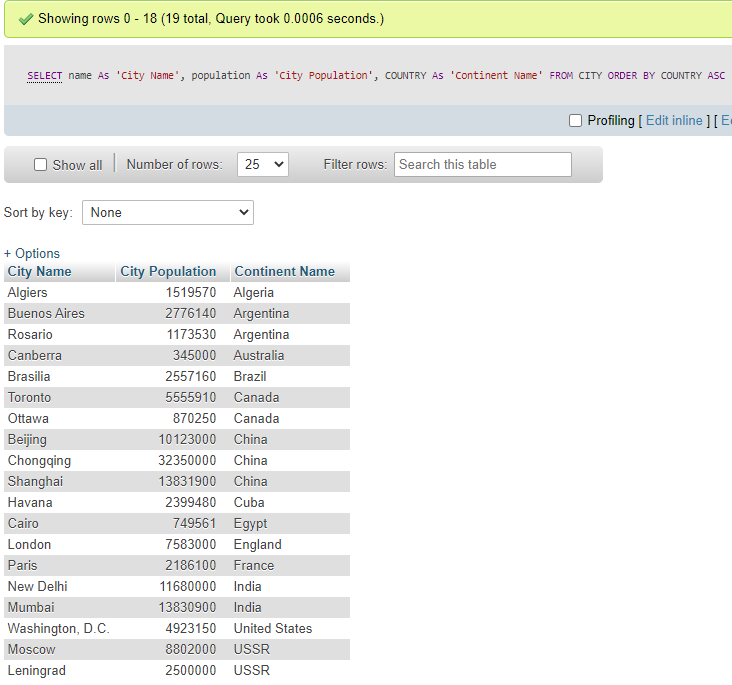
|  |  |  |  |
| --- | --- | --- | --- |
| Lowest Population | Highest Population | Total Population | Average Population |



1. **List the cities, their population and the continents they are in. List them in the alphabetical order of the Continent name.**

**Query Used “SELECT name As 'City Name', population As 'City Population', COUNTRY As 'Continent Name' FROM CITY ORDER BY COUNTRY ASC”**

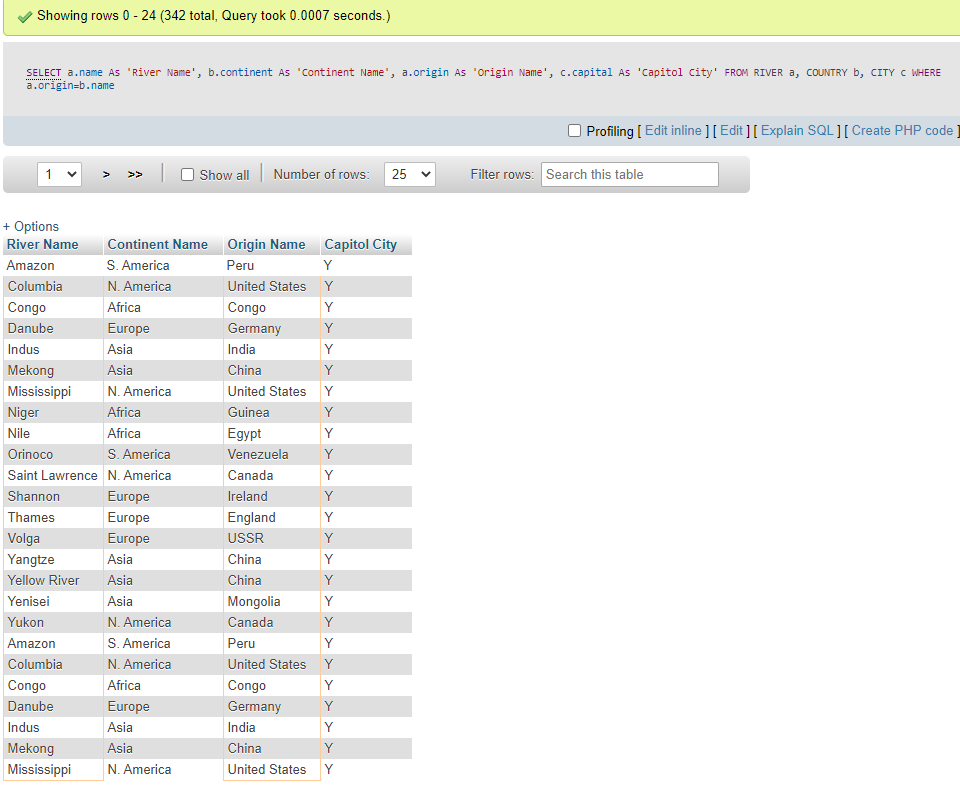
|  |  |  |
| --- | --- | --- |
| City Name | City Population | Continent Name |



1. **List all rivers in the “mini\_world” by their names, the continents they are in and the capital cities of the countries they originate from. Use the following heading.**

**Query Used “SELECT a.name As 'River Name', b.continent As 'Continent Name', a.origin As 'Origin Name', c.capital As 'Capitol City' FROM RIVER a, COUNTRY b, CITY c WHERE a.origin=b.name”**

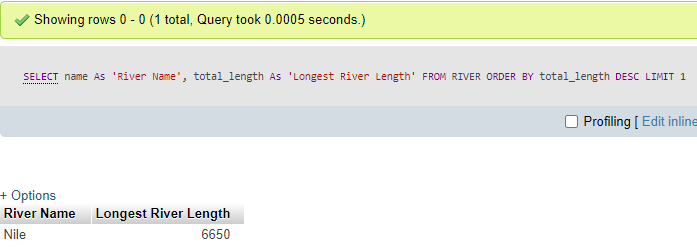
|  |  |  |  |
| --- | --- | --- | --- |
| River Name | Continent Name | Country of Origin | Capital City |



1. **Write a SQL command to display the name and country of origin of the longest river and its length.**

**Query Used “SELECT name As 'River Name', total\_length As 'Longest River Length' FROM RIVER ORDER BY total\_length DESC LIMIT 1”**

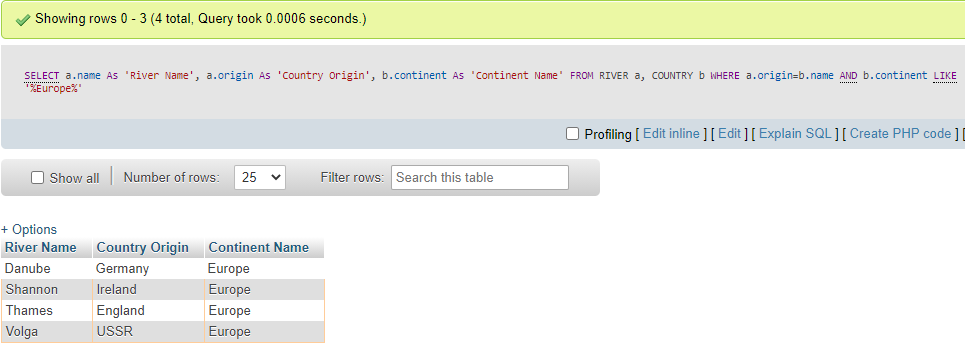
|  |  |
| --- | --- |
| Longest River Name | Longest River Length |



1. **List all rivers that originate from European countries.**

**Query Used “SELECT a.name As 'River Name', a.origin As 'Country Origin', b.continent As 'Continent Name' FROM RIVER a, COUNTRY b WHERE a.origin=b.name AND b.continent LIKE '%Europe%' ”**

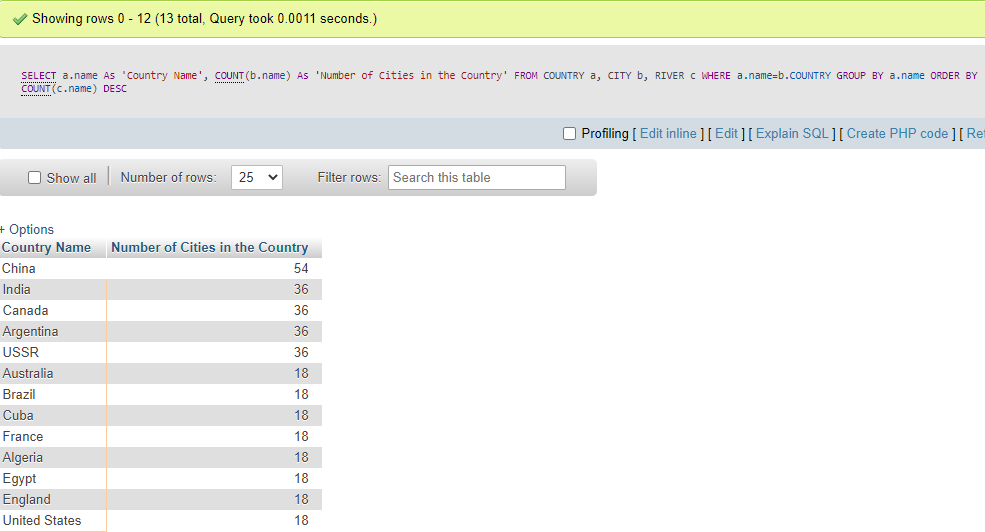
|  |  |  |
| --- | --- | --- |
| River Name | Country Origin | Continent Name |



1. **List all countries and the number of cities they have. List them in the ascending order of the number of rivers.**

**Query Used” SELECT a.name As 'Country Name', COUNT(b.name) As 'Number of Cities in the Country' FROM COUNTRY a, CITY b, RIVER c WHERE a.name=b.COUNTRY GROUP BY a.name ORDER BY COUNT(c.name) DESC”**

|  |  |
| --- | --- |
| Country Name | Number of Cities in the Country |



1. **List those countries that have 2 or more rivers originated from them. List them in the descending order of the total number rivers originated from.**

**Query Used “SELECT a.name As 'Country Name', COUNT(b.origin) As 'Total number of Rivers Originate from The Country' FROM COUNTRY a, RIVER b WHERE b.origin =a.name GROUP BY a.name HAVING count(b.origin) >= 2 ORDER BY COUNT(b.origin) DESC ”**

|  |  |
| --- | --- |
| Country Name | Total number of Rivers Originate from The Country |



1. **Write a SQL query to find out which country has the most of number of rivers originate from.**

**Query Used “SELECT a.name As 'Country Name', COUNT(b.origin) As 'Total number of Rivers Originate from The Country' FROM COUNTRY a, RIVER b WHERE b.origin =a.name GROUP BY a.name ORDER BY COUNT(b.origin) DESC LIMIT 1”**

|  |
| --- |
| Country that has the most number of Rivers Originate from |

