Just IT

B2W group

Apprenticeships | Training | Recruitment

Data Technician

Name:

Course Date:

Table of contents

Day 1: Task 1

Please research and complete the below questions relating to key concepts of databases.

What is a primary key?	A primary key is a column (or a set of columns) in a database table that uniquely identifies each row in that table. It is a fundamental concept in relational database design.
How does this differ from a secondary key?	A primary key uniquely identifies each record in a table, is always unique, non-null, and one per table. A secondary key (like a foreign or candidate key) helps with data retrieval or establishing relationships between tables, can allow duplicates and nulls, and there can be multiple secondary keys in a table.
How are primary and foreign keys related?	Primary keys and foreign keys are closely related through database relationships. A primary key uniquely identifies records in its table, while a foreign key is a column in another table that references the primary key. This relationship ensures data integrity by linking the two tables, enabling consistent and reliable associations between them.
Provide a real-world example of a one-to-one relationship	National id and customer.
Provide a real-world example of a one-to-many relationship	Mother to children.

Provide a real-world example of a many-to-many relationship	Customer and product purchased.

Day 1: Task 2

Please research and complete the below questions relating to key concepts of databases.

What is the difference between a relational and non-relational database?	The key difference between relational and non-relational databases lies in structure and flexibility: • Relational Databases: Use structured tables with predefined schemas and relationships between tables. They are ideal for complex queries and data consistency (e.g., SQL databases like MySQL, PostgreSQL). • Non-Relational Databases: Use flexible data models like documents, key-value pairs, or graphs, without strict schemas. They are better for handling unstructured or rapidly changing data (e.g., NoSQL databases like MongoDB, Cassandra). Relational databases focus on structure and relationships, while non-relational ones emphasize flexibility and scalability.
What type of data would benefit off the non-relational model? **Why?**	Unstructured or semi-structured data benefits from non-relational databases because they handle flexible, dynamic, or varied formats efficiently. Non-relational databases are ideal for these due to their scalability, schema flexibility, and ability to manage diverse data types.

Day 3: Task 1

Please research the below 'JOIN' types, explain what they are and provide an example of the types of data it would be used on.

Self-join	A self join is when a table is joined with itself to compare rows within the same table. It is commonly used to find relationships between rows, such as hierarchical data (e.g., employees and their managers).
Right join	A right join (or right outer join) returns all rows from the right table and the matching rows from the left table. If there's no match, NULL is shown for the left table's columns.
Full join	A full join (or full outer join) returns all rows from both the left and right tables. If there is no match, NULL is shown for the columns from the table that doesn't have a matching row.
Inner join	An inner join returns only the rows that have matching values in both tables based on the specified condition. If there is no match, the row is excluded from the result.
Cross join	A cross join returns the Cartesian product of two tables, meaning it combines every row from the first table with every row from the second table. This can result in a large number of rows, especially if both tables have many records.
Left join	A left join (or left outer join) returns all rows from the left table and the matching rows from the right table. If there is no match, NULL is returned for the columns from the right table.

Day 4: Task 1: Written

In your groups, discuss and complete the below activity. You can either nominate one writer or split the elements between you. Everyone however must have the completed work below:

Imagine you have been hired by a small retail business that wants to streamline its operations by creating a new database system. This database will be used to manage inventory, sales, and customer information. The business is a small corner shop that sells a range of groceries and domestic products. It might help to picture your local convenience store and think of what they sell. They also have a loyalty program, which you will need to consider when deciding what tables to create.

Write a 500-word essay explaining the steps you would take to set up and create this database. Your essay should cover the following points:

1. ***Understanding the Business Requirements**:*
   a. *What kind of data will the database need to store?*
   b. *Who will be the users of the database, and what will they need to accomplish?*
2. ***Designing the Database Schema**:*
   a. *How would you structure the database tables to efficiently store inventory, sales, and customer information?*
   b. *What relationships between tables are necessary (e.g., how sales relate to inventory and customers)?*
3. ***Implementing the Database**:*
   a. *What SQL commands would you use to create the database and its tables?*
   b. *Provide examples of SQL statements for creating tables and defining relationships between them.*
4. ***Populating the Database**:*
   a. *How would you input initial data into the database? Give examples of SQL INSERT statements.*
5. ***Maintaining the Database**:*
   a. *What measures would you take to ensure the database remains accurate and up to date?*
   b. *How would you handle backups and data security?*

Your essay should include specific examples of SQL commands and explain why each step is necessary for creating a functional and efficient database for the retail business.

Pleas
e
write
your
500-
word
essay
here

A small retail business needs a comprehensive database to effectively manage its operations. This database should include detailed product information, customer data, sales records, supplier information and order details. This data allows for informed business decisions, efficient inventory management, and improved customer satisfaction. The database will be used by various personnel, including the owner for analysis and management, and staff for sales and inventory management. This collaborative use of the database will streamline operations and enhance the overall efficiency of the cornershop.

Database Schema

The database would consist of four tables: Customers, Loyalty, Products, and Sales.

The **Customers** table includes:

> **CustomerID** (unique and auto-incrementing), **FirstName**, **LastName**, **Email**, **Address**, and **LoyaltyPoints**.
> The primary key is **CustomerID.**

The **Loyalty** table includes:

> **CustomerID** (relating to **CustomerID** in the Customers table), **LoyaltyPoints**, **PointsEarned**, and **PointsUsed**.
> The primary key is **CustomerID**, tracking total, earned, and used points for analysis separate to other customer data.

The **Products** table includes:

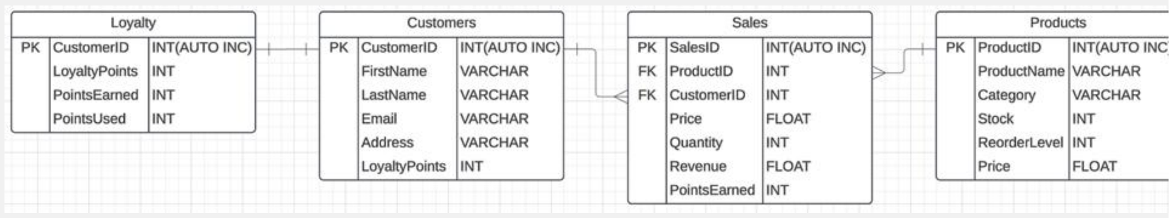
> **ProductID** (unique and auto-incrementing), **ProductName**, **Category**, **Stock**, **ReorderLevel**, and **Price**.
> The primary key is **ProductID**.

The **Sales** table includes:

> **SalesID** (unique and auto-incrementing), **ProductID**, **CustomerID**, **Price**, **Quantity**, **Revenue**, and **PointsEarned**.
>
> The primary key is **SalesID**, with **ProductID** and **CustomerID** as foreign keys that relate to the primary keys of the products and customers tables respectively.

Relationships:

> **Customers** and **Loyalty**: One-to-one, separating loyalty data for analysis.
>
> **Customers** and **Sales**: One-to-many, as one customer can make multiple purchases.
>
> **Products** and **Sales**: One-to-many, as a product can appear in multiple sales but each sale entry links to a single product.



3. Implementing the Database

To implement the database, we would use the following SQL commands to create our database.

Input ***'DROP SCHEMA IF EXISTS grocery_store'*** to prevent duplication, and input ***'CREATE SCHEMA grocery_store'*** to create a SQL schema, then input ***'USE grocery_store'*** to implement this schema.

And the following example commands to create our tables in the Database.

Input '***CREATE TABLE Loyalty(***

CustomerID INT NOT NULL AUTO_INCREMENT,

LoyaltyPoints INT NOT NULL,

PointsEarned INT NOT NULL,

PointsUsed INT NOT NULL,

PRIMARY KEY (CustomerID))' to create the table 'Loyalty'.

Input '***CREATE TABLE Customers(***

CustomerID INT NOT NULL AUTO_INCREMENT,

FirstName VARCHAR NOT NULL,

LastName VARCHAR NOT NULL,

Email VARCHAR NOT NULL,

Address VARCHAR NOT NULL,

LoyaltyPoints INT NOT NULL,

PRIMARY KEY (CustomerID))' to create the table 'Customers'.

Input '***CREATE TABLE Sales(***

SalesID INT NOT NULL AUTO_INCREMENT,

ProductID INT NOT NULL,

CustomerID INT NOT NULL,

Price FLOAT NOT NULL,

Quantity INT NOT NULL,

Revenue FLOAT NOT NULL,

PointsEarned INT NOT NULL,

PRIMARY KEY (SalesID),

FOREIGN KEY (ProductID),

FOREIGN KEY (CustomerID))' to create the table 'Sales'.

Input '***CREATE TABLE Products(***

ProductID INT NOT NULL AUTO_INCREMENT,

ProductName VARCHAR NOT NULL,

Category VARCHAR NOT NULL,

Stock INT NOT NULL,

ReorderLevel INT NOT NULL,

Price FLOAT NOT NULL,

PRIMARY KEY (ProductID))' to create the table 'Products'.

4. Populating the Database

Create a Google Forms document for clients to input customer details and purchases. Another form manages stock items, quantities, and prices, linked to the database. Customers earn loyalty points for purchases, and all data is connected to the Sales Table.

If data needs to be entered manually we would do so as follows:

INSERT INTO Products (ProductID,ProductName, Category, Stock, ReorderLevel, Price)

VALUES

(1, 'Milk', 'Dairy', 15, 5, £1.00),

(2, ‘Bread’, ‘Bakery’, 20, 5, £1.20),

(3, ‘Peanuts’, ‘Snacks’, 10, 3, £0.75)

(4, ‘Eggs’, ‘Poultry’, 25, 7, £1.50)

(5, ‘Chocolate Bar’, ‘Snacks’, 40, 10, £0.50);

INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Address)

VALUES

(1,’Donald’, ‘Trumpet’, ‘d.trumpet@yahoo.com’, ‘25 Avenue London’, 100)

(2, ‘Elon’, ‘Tusk’, ‘e.tusk@yahoo.com’, ‘12 Avenue London’, 55)

(3, ‘Billy’, ‘Gate’, ‘b.gated@yahoo.com’, ‘16 Avenue London’, 250)

(4, ‘John’, ‘Doe’, ‘j.doe@yahoo.com’, ‘5 Avenue London’, 20)

(5, ‘Jelly’, ‘Jane’, ‘j.jane@yahoo.com’, ‘33 Avenue London’, 5);

INSERT INTO SALES (SalesID, ProductID, CustomerID, Price, Quantity, Revenue, PointsEarned)

VALUES

(1, 3, 2, £0.75, 2, £1.50, 2)

(2, 5, 2, £0.50, 4, £2.00, 2)

(3, 4, 1, £1.50, 3, £4.50, 5)

(4, 4, 2, £1.50, 2, £3.00, 3)

(5, 1, 2, £1.00, 4, £4.00, 4);

5. Maintaining the Database

In order to ensure the database remains accurate, up-to-date and secure we would following these procedures:

Validation Rules: Enforce data integrity using constraints like NOT NULL and CHECK.

Regular Backups: Automate daily full and incremental backups, stored offsite, and regularly test restores.

Access Control: Limit access based on roles (e.g., admins, cashiers).

Data Encryption: Use SSL/TLS for data in transit and encryption for sensitive data at rest.

Software Updates: Keep systems updated to protect against vulnerabilities.

Monitoring & Auditing: Track activities and unusual queries.

Data Masking: Protect sensitive data in non-production environments.

Network Security: Use firewalls and private networks to restrict access.

Security Audits: Regularly review access policies and security protocols.

Day 4: Task 2: SQL Practical

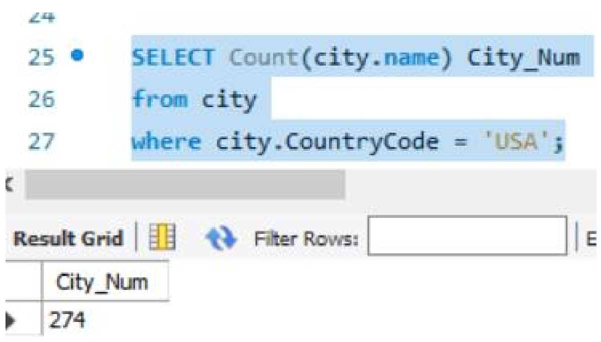
In your groups, work together to answer the below questions. It may be of benefit if one of you shares your screen with the group and as a team answer / take screen shots from there.

<u>Setting up the database:</u>

1. **Download world_db(1)**
2. **Follow each step to create your database**

For each question I would like to see both the syntax used and the output.

1. **Count Cities in USA:** *Scenario:* You've been tasked with conducting a demographic analysis of cities in the United States. Your first step is to determine the total number of cities within the country to provide a baseline for further analysis.

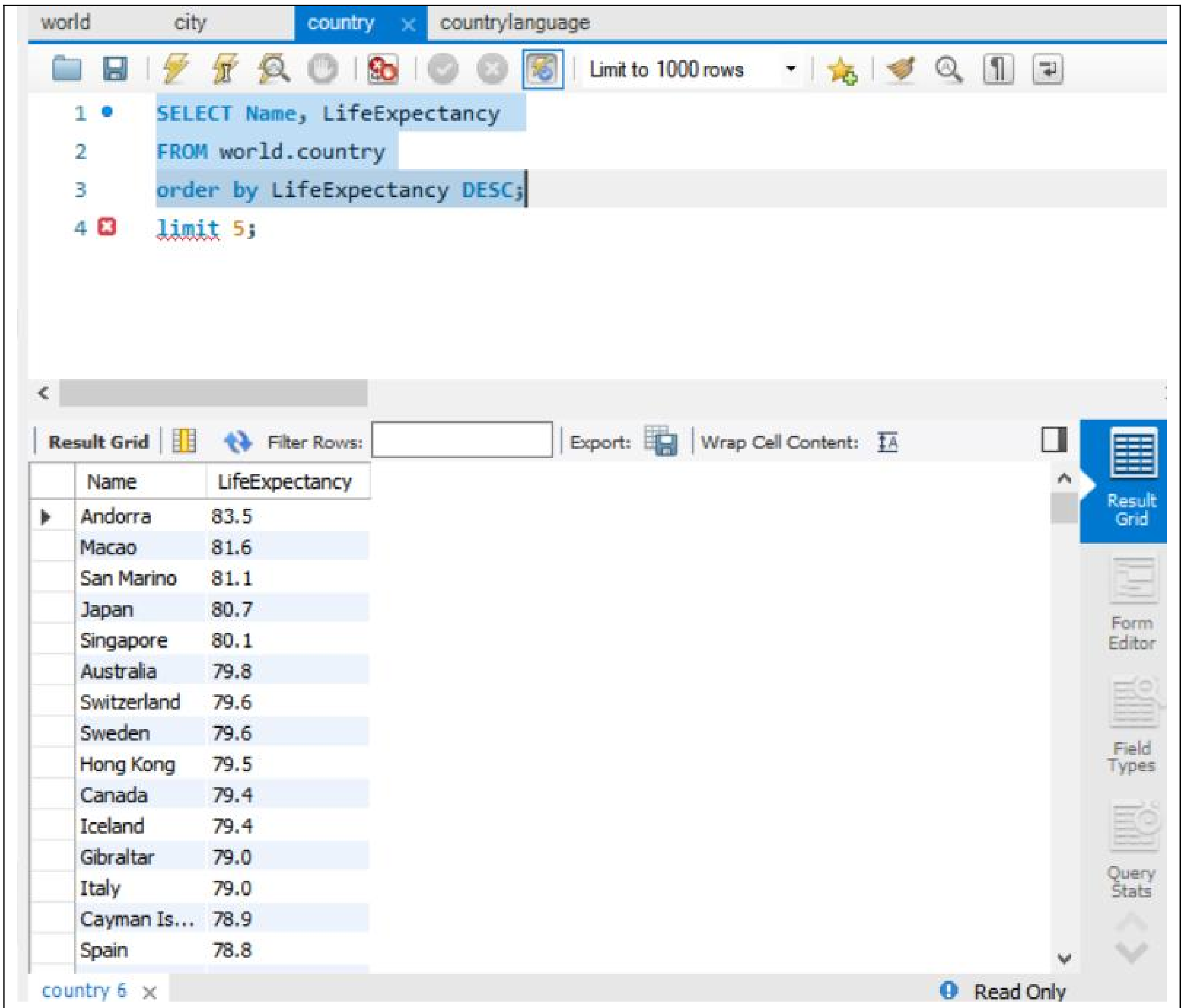


```sql
SELECT Count(city.name) City_Num
from city
where city.CountryCode = 'USA';
```

```
City_Num
274
```

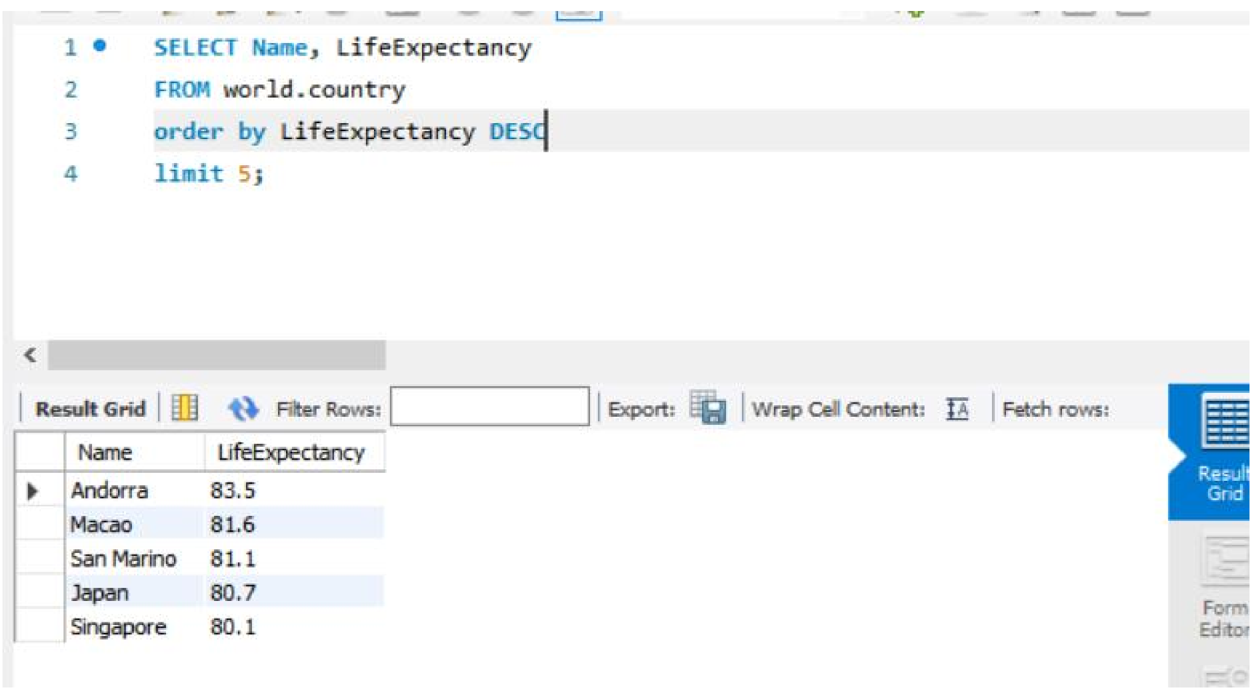
There are 274 cities in USA.

2. **Country with Highest Life Expectancy:** *Scenario:* As part of a global health initiative, you've been assigned to identify the country with the highest life expectancy. This information will be crucial for prioritising healthcare resources and interventions.



```
SELECT Name, LifeExpectancy
FROM world.country
order by LifeExpectancy DESC;
limit 5;
```

Name	LifeExpectancy
Andorra	83.5
Macao	81.6
San Marino	81.1
Japan	80.7
Singapore	80.1
Australia	79.8
Switzerland	79.6
Sweden	79.6
Hong Kong	79.5
Canada	79.4
Iceland	79.4
Gibraltar	79.0
Italy	79.0
Cayman Is...	78.9
Spain	78.8

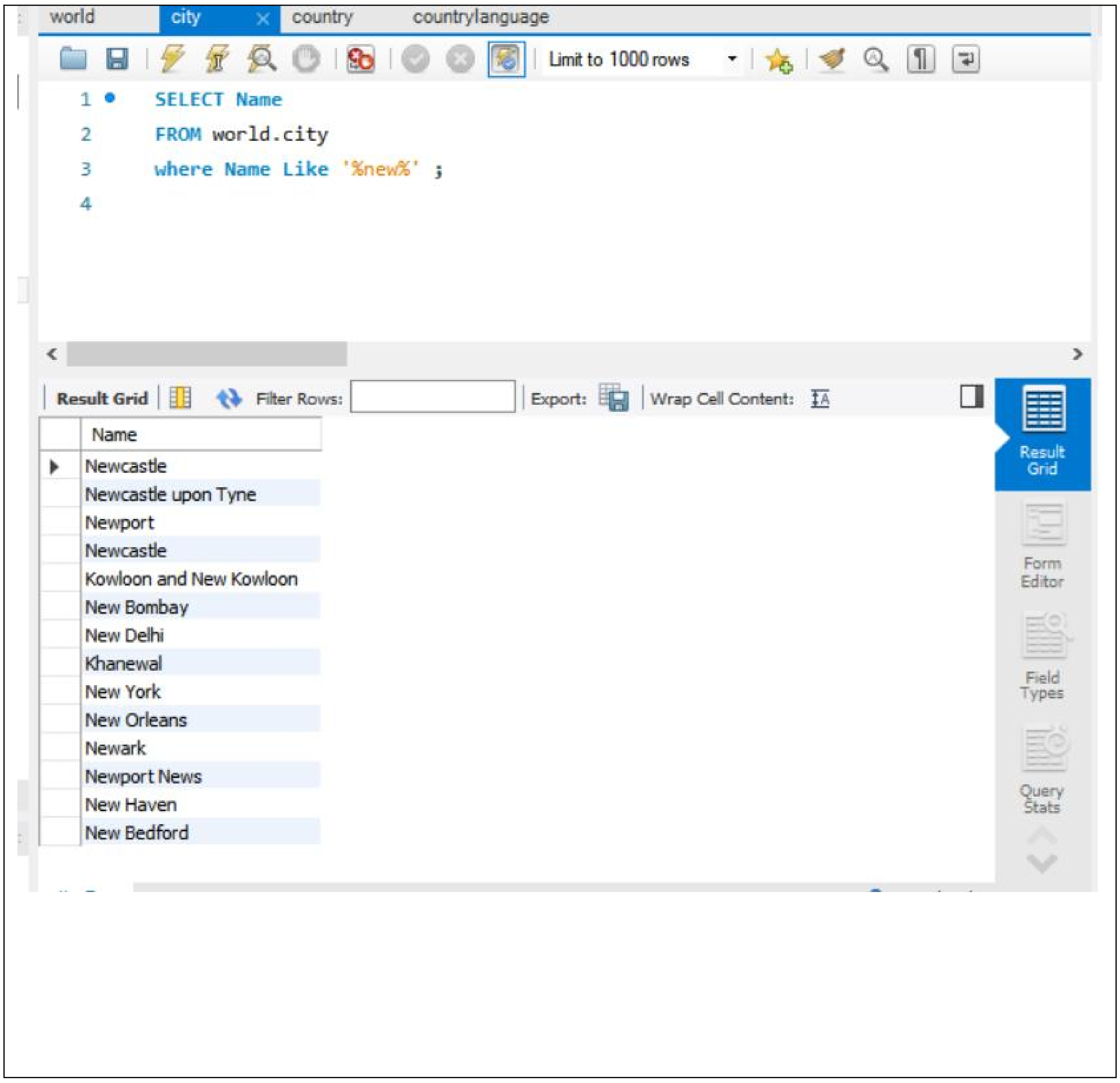


```
SELECT Name, LifeExpectancy
FROM world.country
order by LifeExpectancy DESC
limit 5;
```

Name	LifeExpectancy
Andorra	83.5
Macao	81.6
San Marino	81.1
Japan	80.7
Singapore	80.1

Above two screenshots show the LifeExpectancy rank and the top 5 LifeExpectancy countries. The country with highest LifeExpectancy is Andorra.

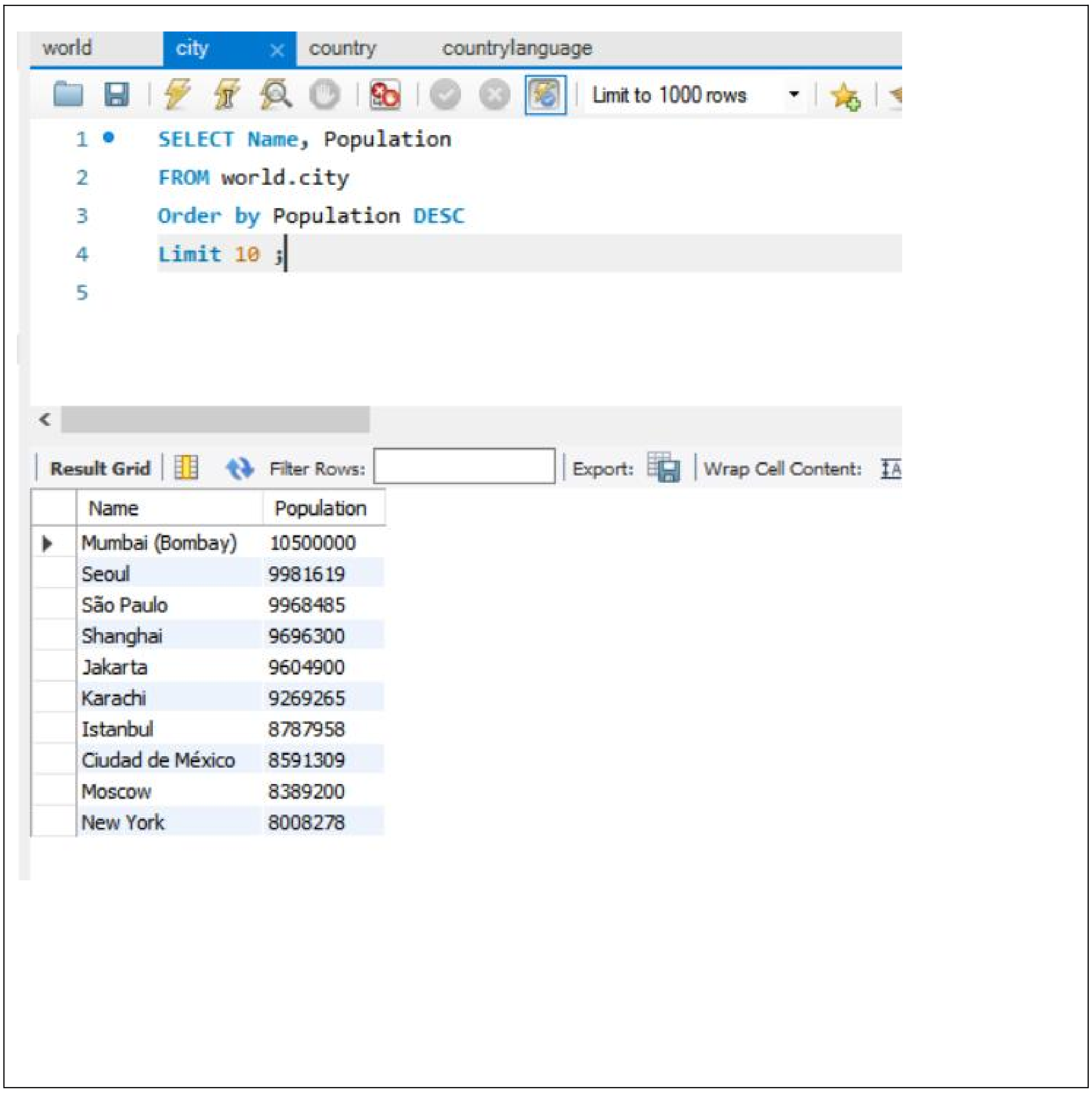
3. **"New Year Promotion: Featuring Cities with 'New :** *Scenario:* In anticipation of the upcoming New Year, your travel agency is gearing up for a special promotion featuring cities with names including the word 'New'. You're tasked with swiftly compiling a list of all cities from around the world. This curated selection will be essential in creating promotional materials and enticing travellers with exciting destinations to kick off the New Year in style.



```
SELECT Name
FROM world.city
where Name Like '%new%' ;
```

Name
Newcastle
Newcastle upon Tyne
Newport
Newcastle
Kowloon and New Kowloon
New Bombay
New Delhi
Khanewal
New York
New Orleans
Newark
Newport News
New Haven
New Bedford

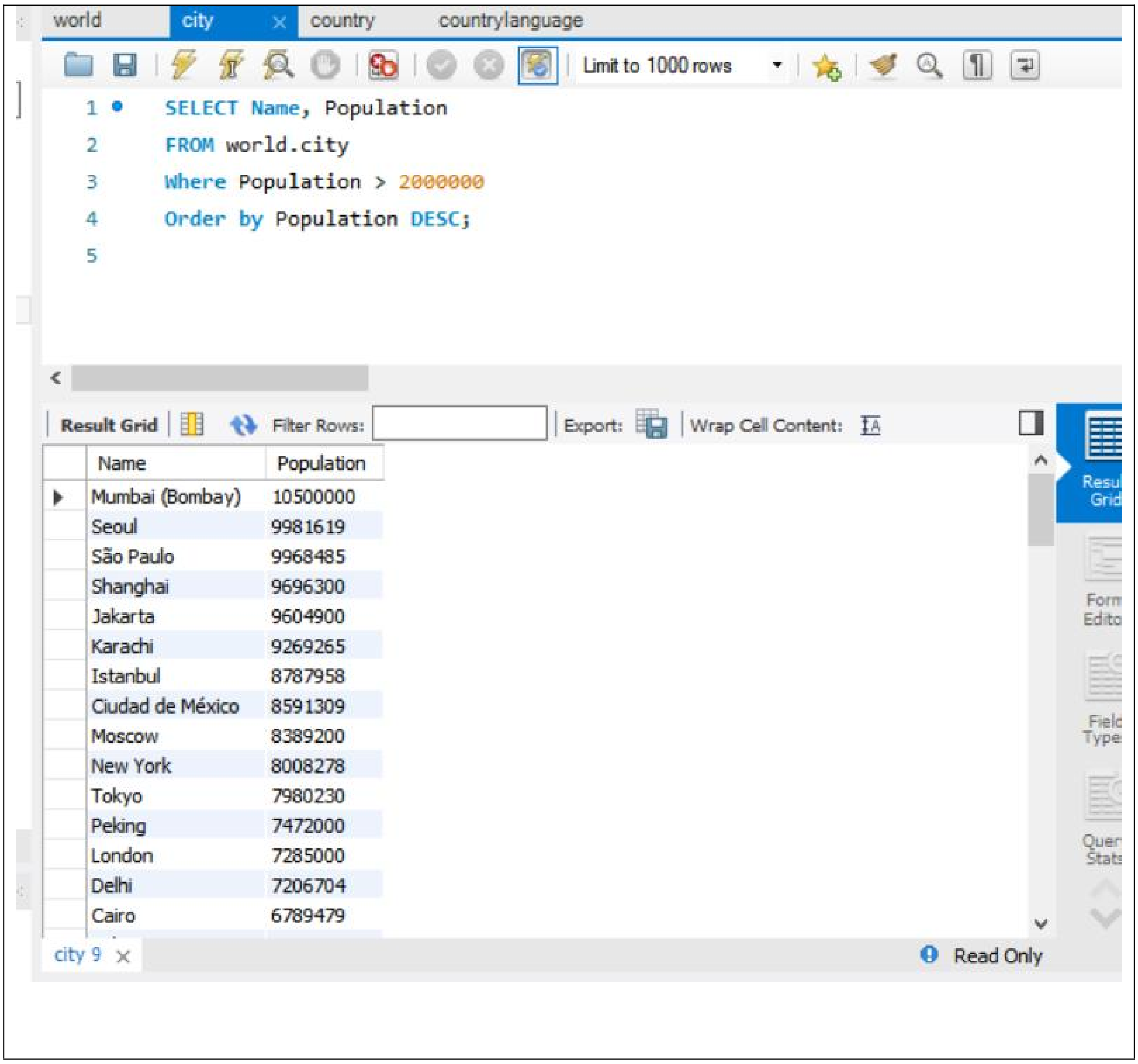
4. **Display Columns with Limit (First 10 Rows):** *Scenario:* You're tasked with providing a brief overview of the most populous cities in the world. To keep the report concise, you're instructed to list only the first 10 cities by population from the database.



```sql
SELECT Name, Population
FROM world.city
Order by Population DESC
Limit 10 ;
```

Name	Population
Mumbai (Bombay)	10500000
Seoul	9981619
São Paulo	9968485
Shanghai	9696300
Jakarta	9604900
Karachi	9269265
Istanbul	8787958
Ciudad de México	8591309
Moscow	8389200
New York	8008278

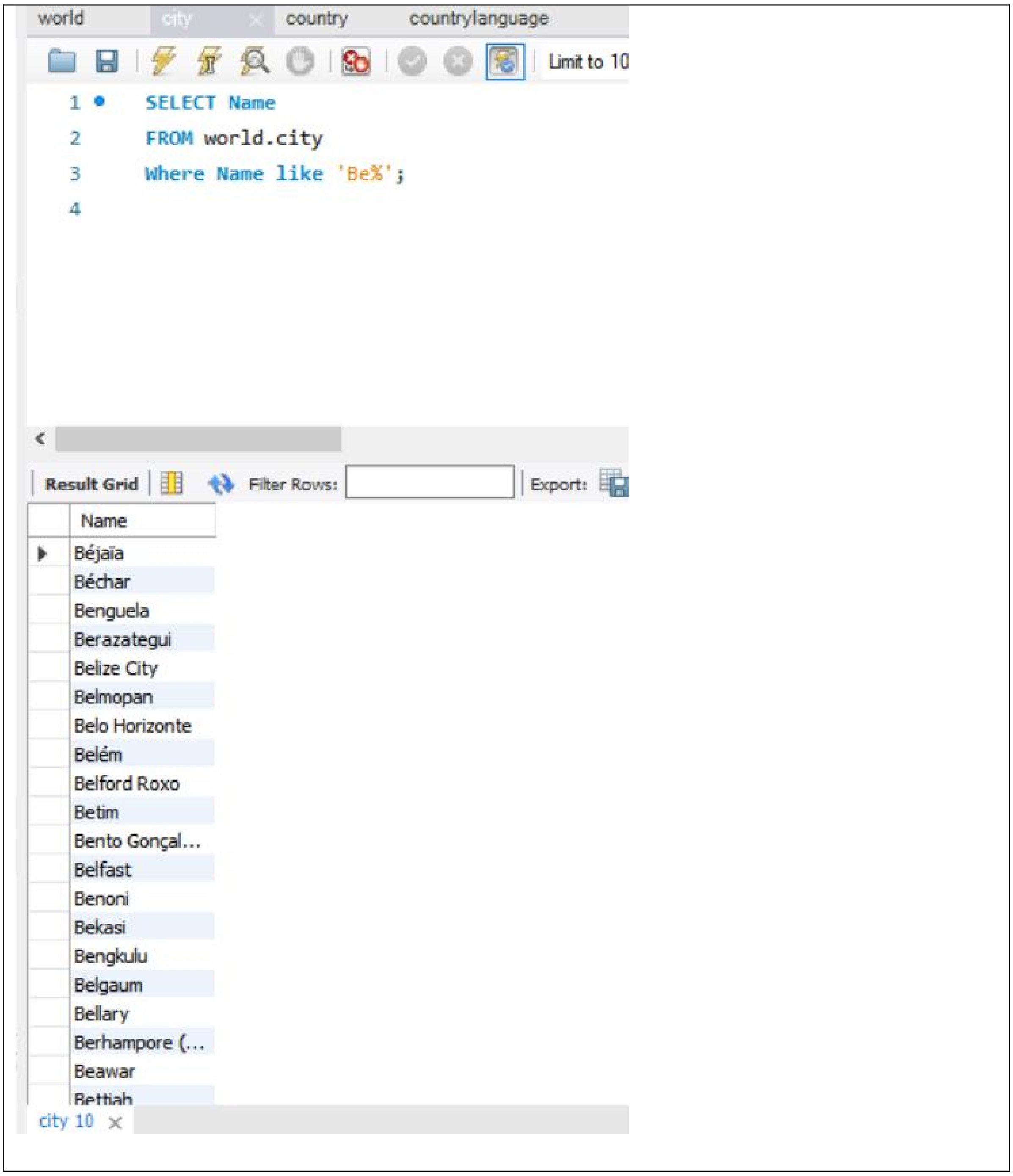
5. **Cities with Population Larger than 2,000,000:** *Scenario:* A real estate developer is interested in cities with substantial population sizes for potential investment opportunities. You're tasked with identifying cities from the database with populations exceeding 2 million to focus their research efforts.



```sql
SELECT Name, Population
FROM world.city
Where Population > 2000000
Order by Population DESC;
```

Name	Population
Mumbai (Bombay)	10500000
Seoul	9981619
São Paulo	9968485
Shanghai	9696300
Jakarta	9604900
Karachi	9269265
Istanbul	8787958
Ciudad de México	8591309
Moscow	8389200
New York	8008278
Tokyo	7980230
Peking	7472000
London	7285000
Delhi	7206704
Cairo	6789479

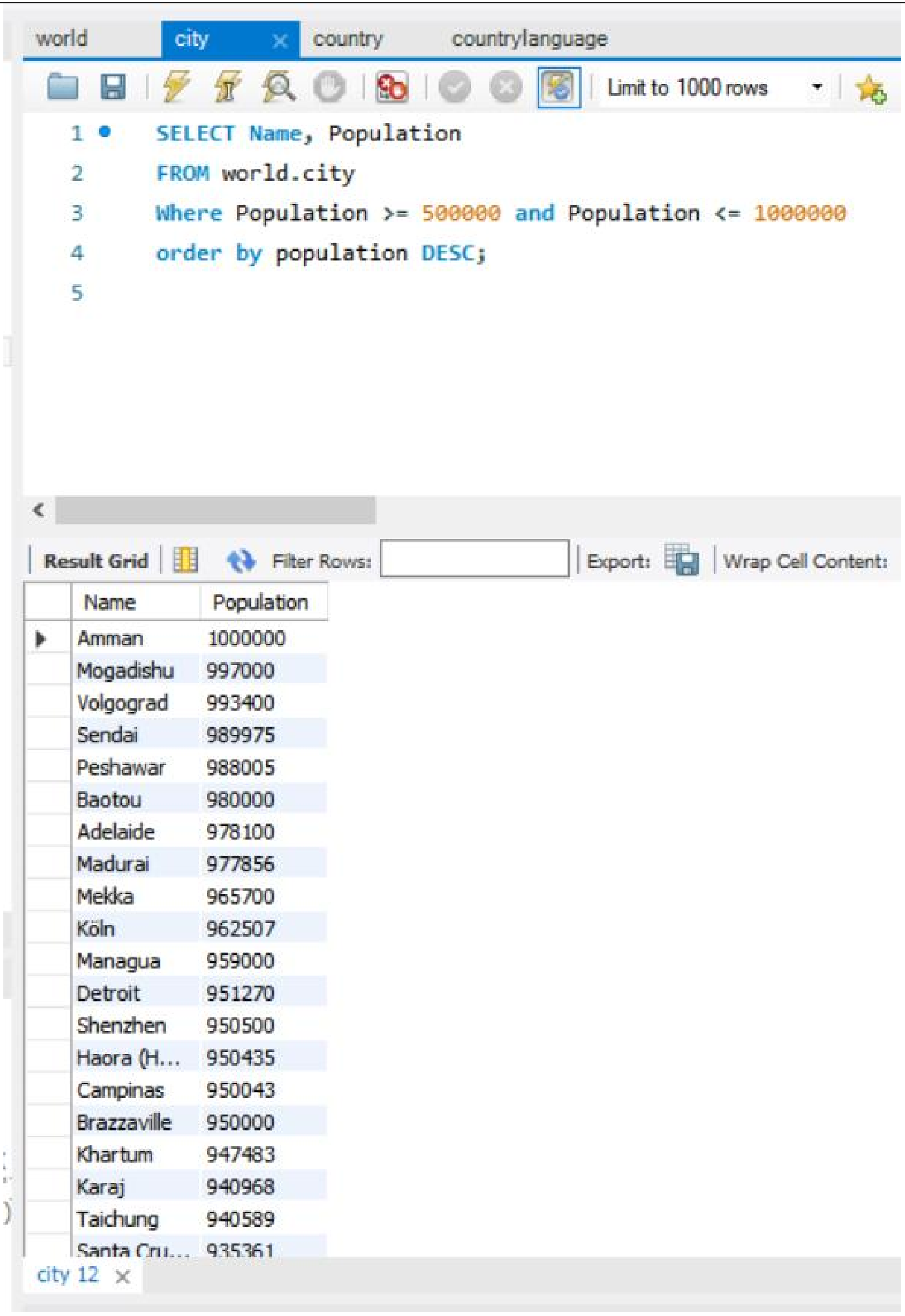
6. **Cities Beginning with 'Be' Prefix:** *Scenario:* A travel blogger is planning a series of articles featuring cities with unique names. You're tasked with compiling a list of cities from the database that start with the prefix 'Be' to assist in the blogger's content creation process.



```
SELECT Name
FROM world.city
Where Name like 'Be%';
```

```
Name
Béjaïa
Béchar
Benguela
Berazategui
Belize City
Belmopan
Belo Horizonte
Belém
Belford Roxo
Betim
Bento Gonçal...
Belfast
Benoni
Bekasi
Bengkulu
Belgaum
Bellary
Berhampore (...
Beawar
Bettiah
```

7. **Cities with Population Between 500,000-1,000,000:** *Scenario:* An urban planning committee needs to identify mid-sized cities suitable for infrastructure development projects. You're tasked with identifying cities with populations ranging between 500,000 and 1 million to inform their decision-making process.

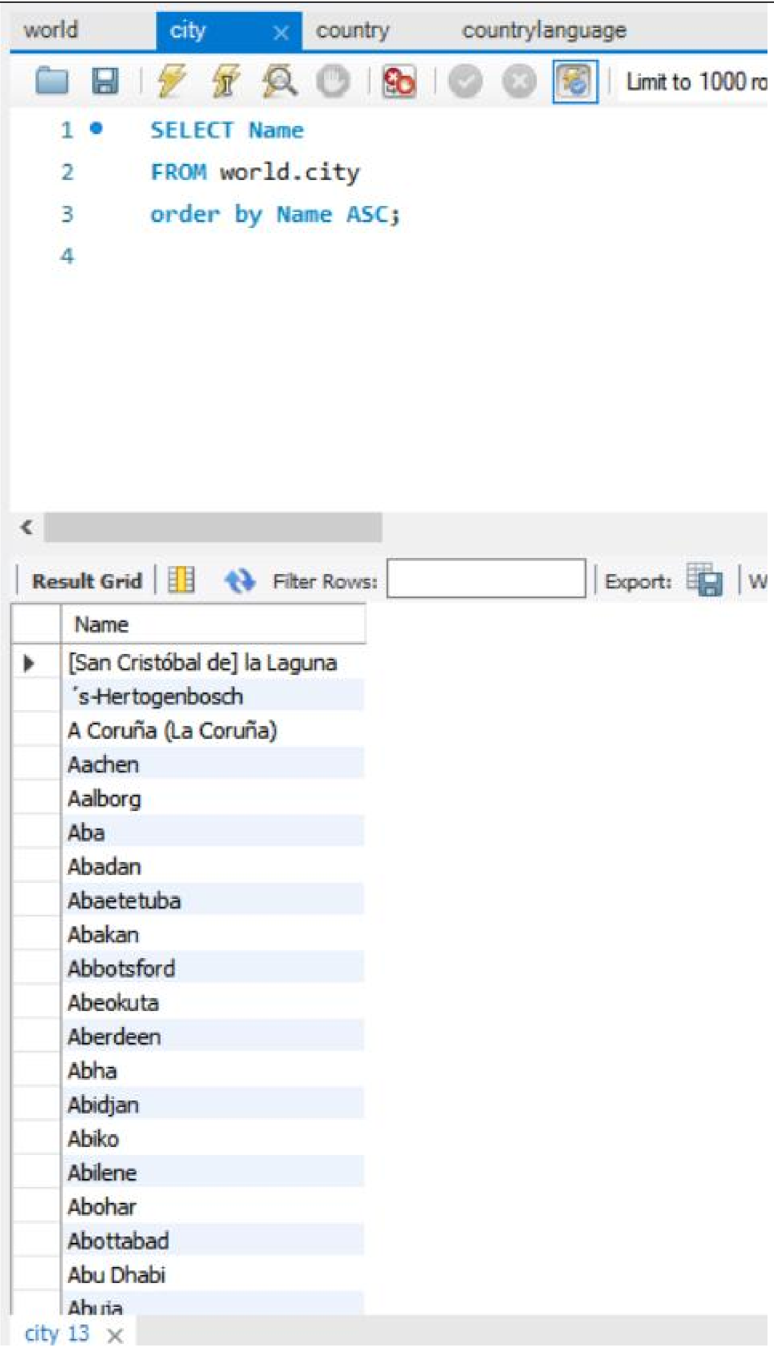


```sql
SELECT Name, Population
FROM world.city
Where Population >= 500000 and Population <= 1000000
order by population DESC;
```

Name	Population
Amman	1000000
Mogadishu	997000
Volgograd	993400
Sendai	989975
Peshawar	988005
Baotou	980000
Adelaide	978100
Madurai	977856
Mekka	965700
Köln	962507
Managua	959000
Detroit	951270
Shenzhen	950500
Haora (H...	950435
Campinas	950043
Brazzaville	950000
Khartum	947483
Karaj	940968
Taichung	940589
Santa Cru...	935361

8. **Display Cities Sorted by Name in Ascending Order:** *Scenario:* A geography teacher is preparing a lesson on alphabetical order using city names. You're tasked

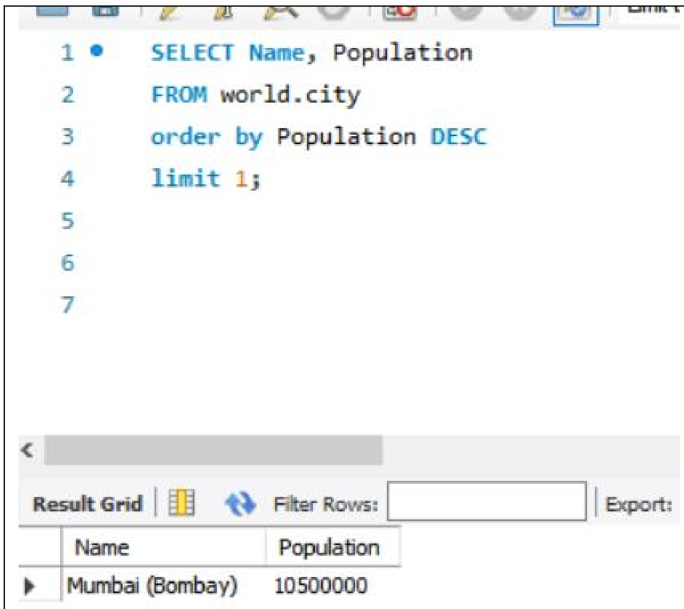
with providing a sorted list of cities from the database in ascending order by name to support the lesson plan.



```
SELECT Name
FROM world.city
order by Name ASC;
```

```
Name
[San Cristóbal de] la Laguna
´s-Hertogenbosch
A Coruña (La Coruña)
Aachen
Aalborg
Aba
Abadan
Abaetetuba
Abakan
Abbotsford
Abeokuta
Aberdeen
Abha
Abidjan
Abiko
Abilene
Abohar
Abottabad
Abu Dhabi
Abuja
```

9. **Most Populated City:** *Scenario:* A real estate investment firm is interested in cities with significant population densities for potential development projects. You're tasked with identifying the most populated city from the database to guide their investment decisions and strategic planning.



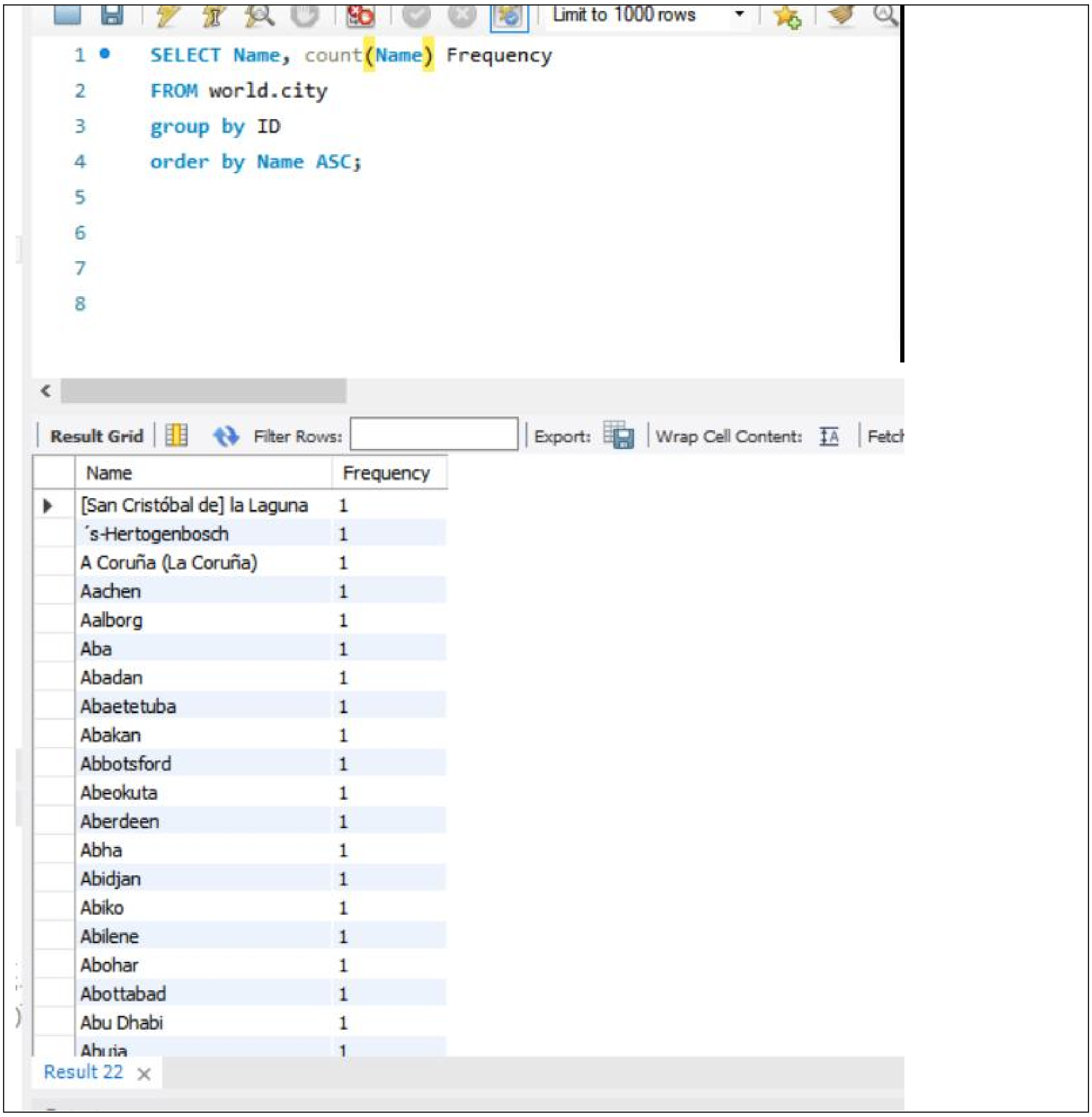
```
SELECT Name, Population
FROM world.city
order by Population DESC
limit 1;
```

Result Grid | Filter Rows: | Export:

Name	Population
Mumbai (Bombay)	10500000

Mumbai is the most populated city.

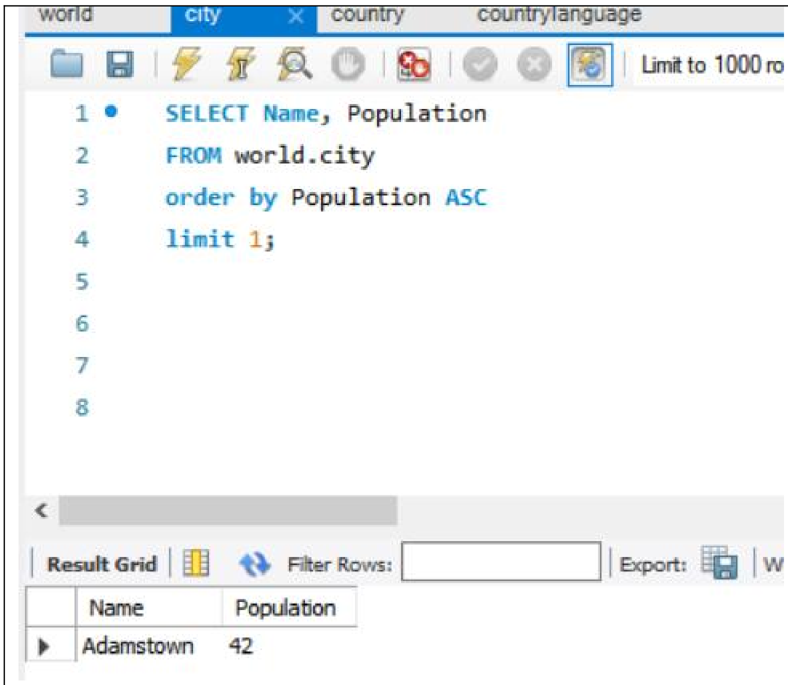
10. **City Name Frequency Analysis: Supporting Geography Education** *Scenario*: In a geography class, students are learning about the distribution of city names around the world. The teacher, in preparation for a lesson on city name frequencies, wants to provide students with a list of unique city names sorted alphabetically, along with their respective counts of occurrences in the database. You're tasked with this sorted list to support the geography teacher.



```
SELECT Name, count(Name) Frequency
FROM world.city
group by ID
order by Name ASC;
```

Name	Frequency
[San Cristóbal de] la Laguna	1
´s-Hertogenbosch	1
A Coruña (La Coruña)	1
Aachen	1
Aalborg	1
Aba	1
Abadan	1
Abaetetuba	1
Abakan	1
Abbotsford	1
Abeokuta	1
Aberdeen	1
Abha	1
Abidjan	1
Abiko	1
Abilene	1
Abohar	1
Abottabad	1
Abu Dhabi	1
Abuja	1

11. **City with the Lowest Population:** *Scenario:* A census bureau is conducting an analysis of urban population distribution. You're tasked with identifying the city with the lowest population from the database to provide a comprehensive overview of demographic trends.

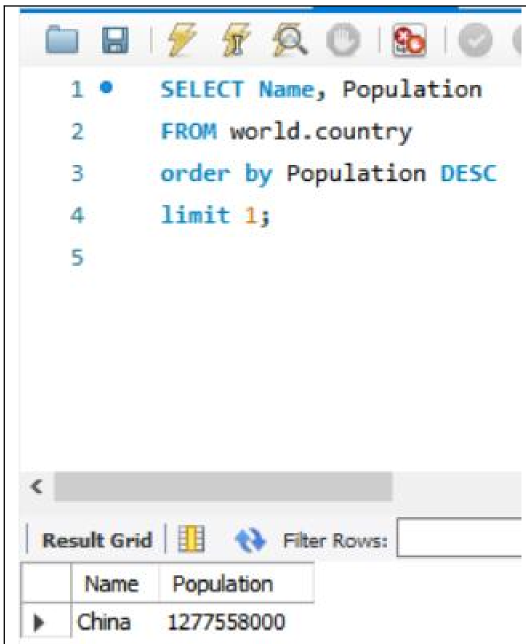


```sql
SELECT Name, Population
FROM world.city
order by Population ASC
limit 1;
```

```
Name       Population
Adamstown  42
```

The city with the lowest population is Adamstown.

12. **Country with Largest Population:** *Scenario:* A global economic research institute requires data on countries with the largest populations for a comprehensive analysis. You're tasked with identifying the country with the highest population from the database to provide valuable insights into demographic trends.



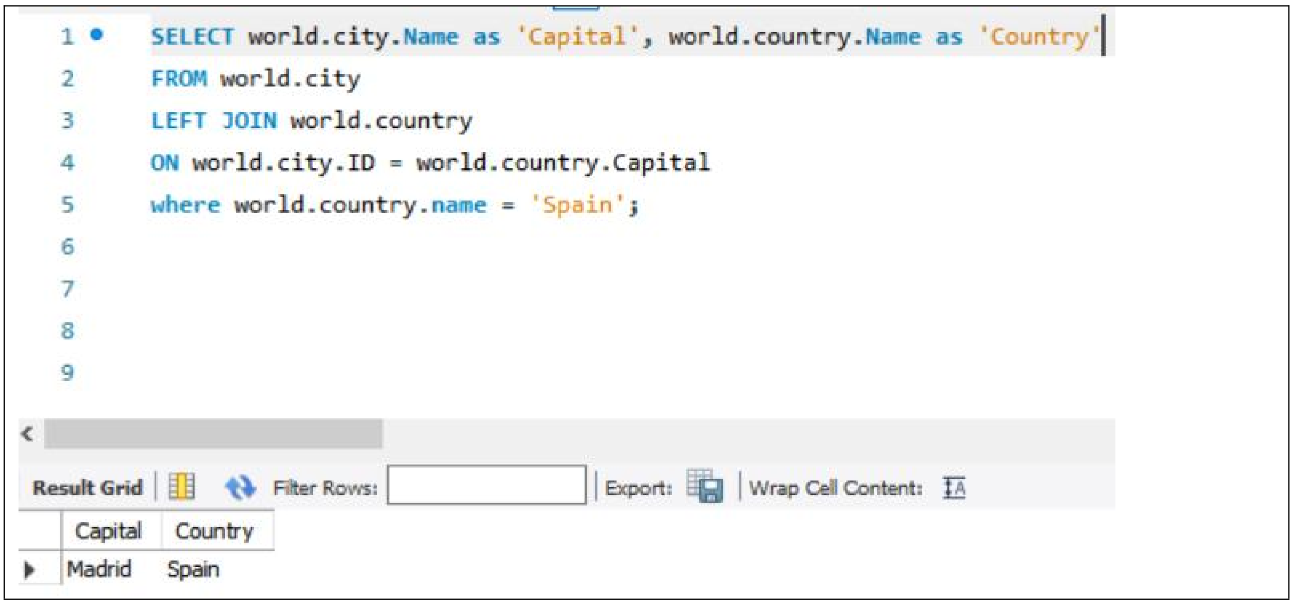
```sql
SELECT Name, Population
FROM world.country
order by Population DESC
limit 1;
```

```
Name   Population
China  1277558000
```

China is the country with the largest population.

13. **Capital of Spain:** *Scenario:* A travel agency is organising tours across Europe and needs accurate information on capital cities. You're tasked with identifying the

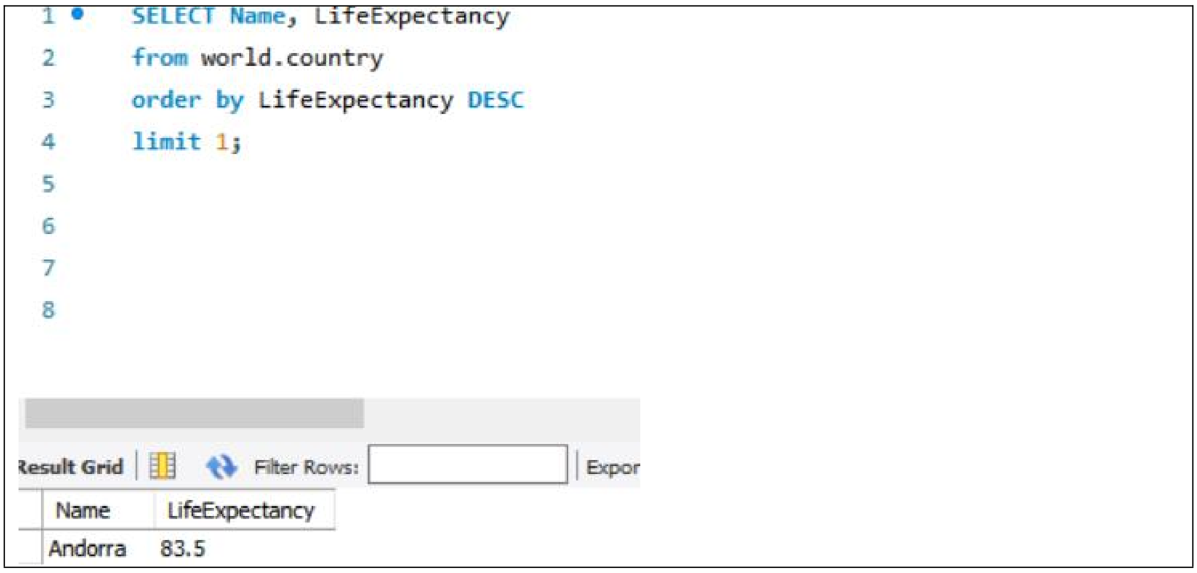
capital of Spain from the database to ensure itinerary accuracy and provide travellers with essential destination information.



```
SELECT world.city.Name as 'Capital', world.country.Name as 'Country'
FROM world.city
LEFT JOIN world.country
ON world.city.ID = world.country.Capital
where world.country.name = 'Spain';
```

Capital	Country
Madrid	Spain

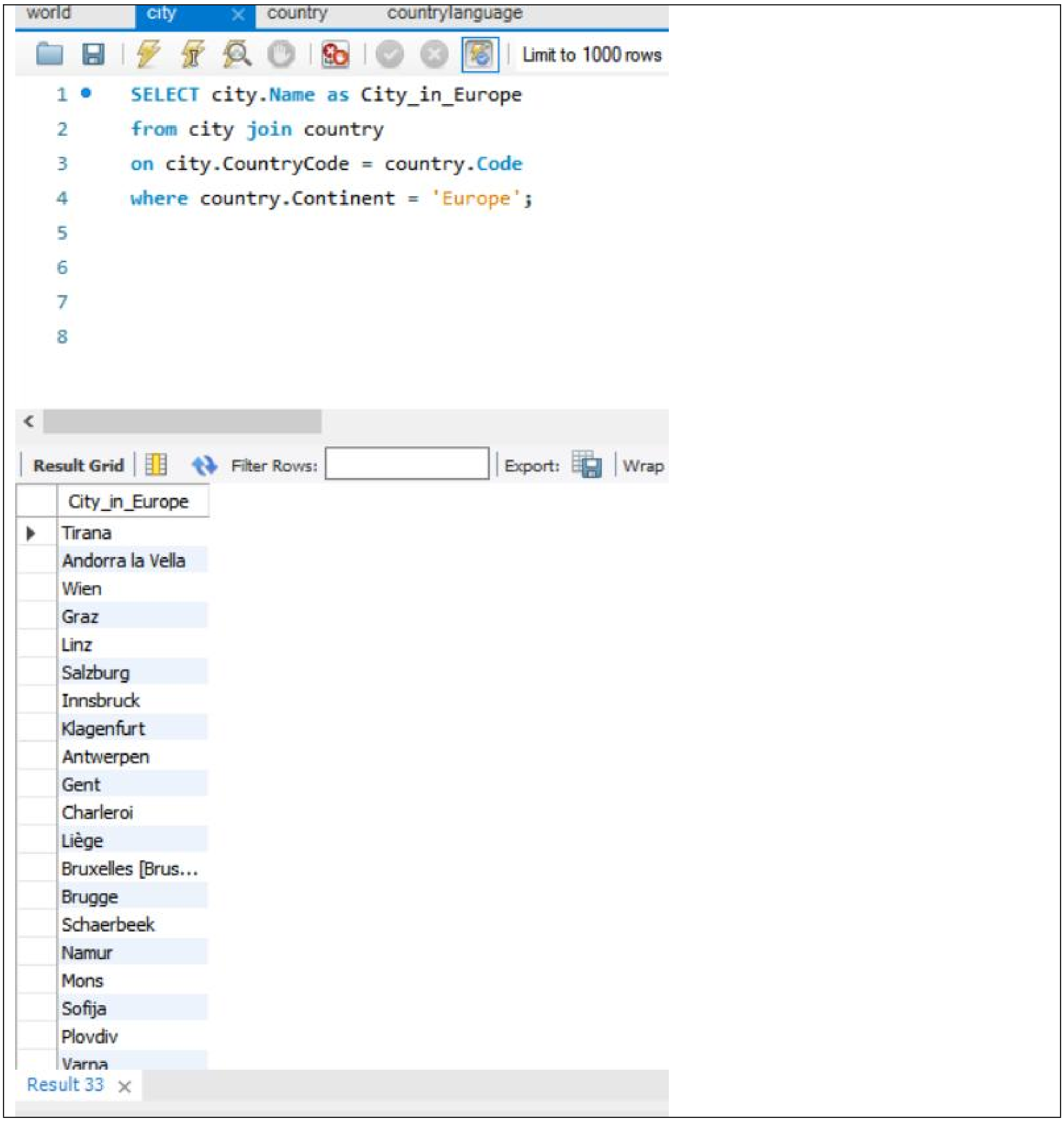
14. **Country with Highest Life Expectancy:** *Scenario:* A healthcare foundation is conducting research on global health indicators. You're tasked with identifying the country with the highest life expectancy from the database to inform their efforts in improving healthcare systems and policies.



```
SELECT Name, LifeExpectancy
from world.country
order by LifeExpectancy DESC
limit 1;
```

Name	LifeExpectancy
Andorra	83.5

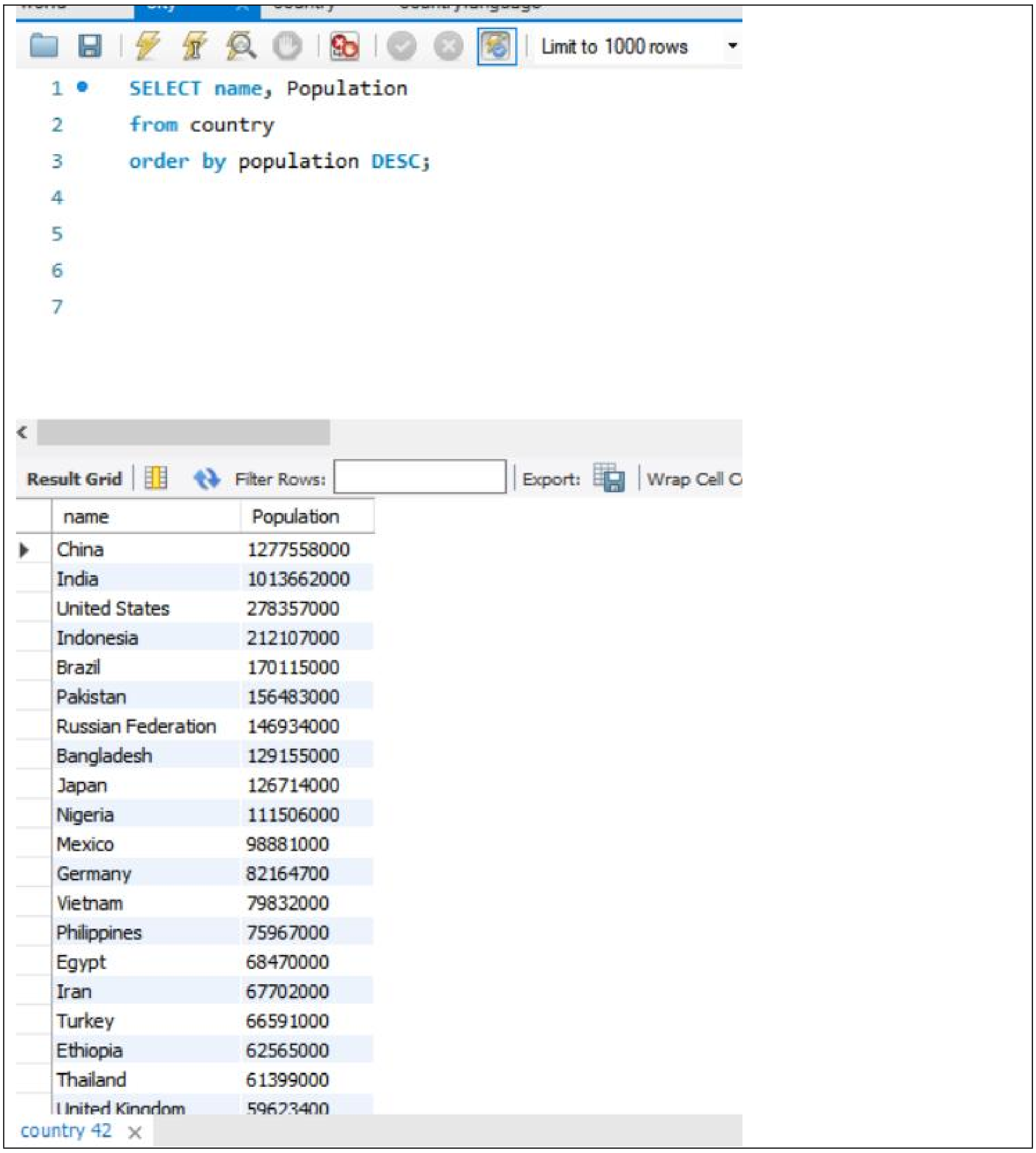
15. **Cities in Europe:** *Scenario:* A European cultural exchange program is seeking to connect students with cities across the continent. You're tasked with compiling a list of cities located in Europe from the database to facilitate program planning and student engagement.



```sql
SELECT city.Name as City_in_Europe
from city join country
on city.CountryCode = country.Code
where country.Continent = 'Europe';
```

City_in_Europe
Tirana
Andorra la Vella
Wien
Graz
Linz
Salzburg
Innsbruck
Klagenfurt
Antwerpen
Gent
Charleroi
Liège
Bruxelles [Brus...
Brugge
Schaerbeek
Namur
Mons
Sofija
Plovdiv
Varna

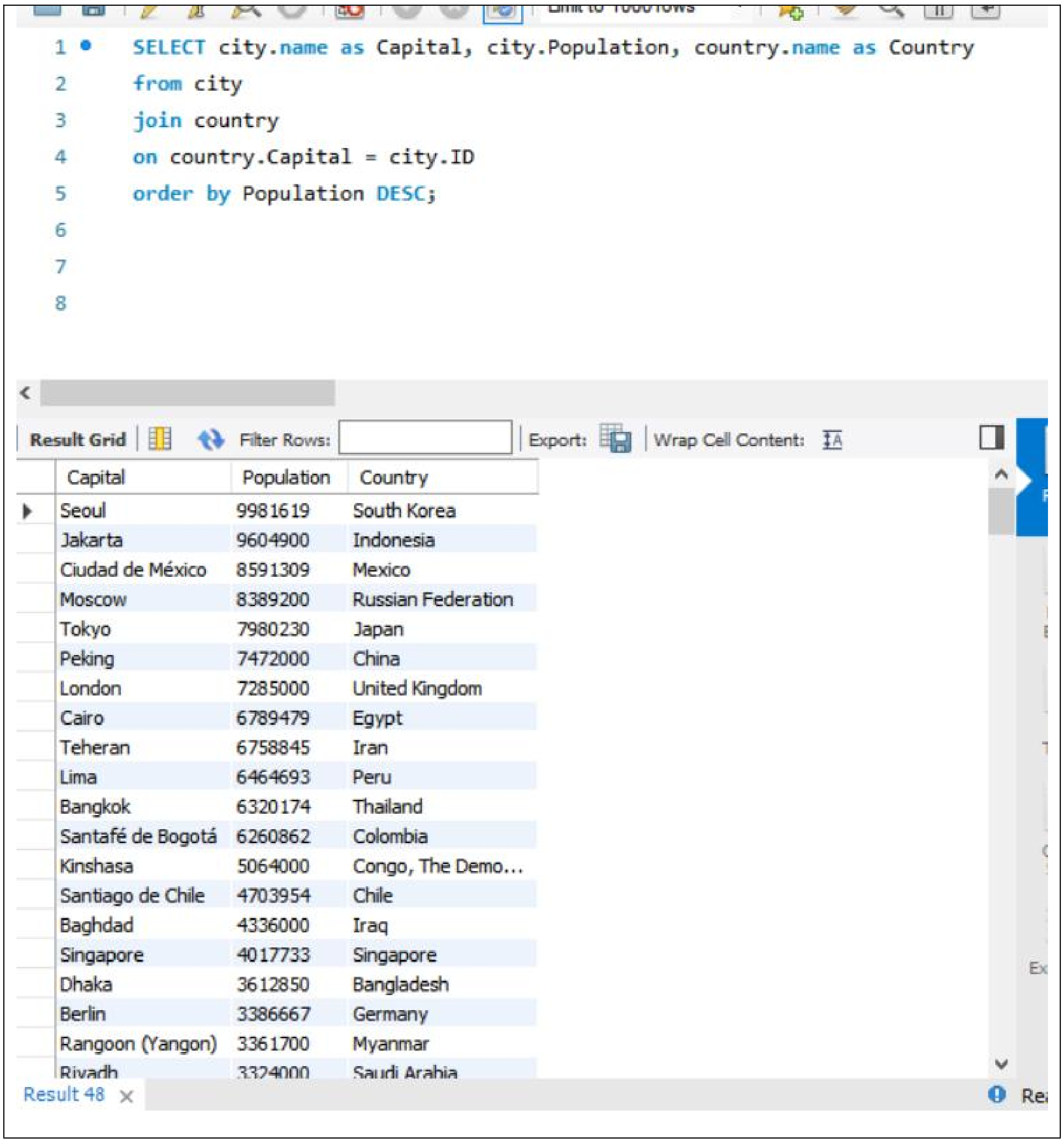
16. **Average Population by Country:** *Scenario:* A demographic research team is conducting a comparative analysis of population distributions across countries. You're tasked with calculating the average population for each country from the database to provide valuable insights into global population trends.



```
SELECT name, Population
from country
order by population DESC;
```

name	Population
China	1277558000
India	1013662000
United States	278357000
Indonesia	212107000
Brazil	170115000
Pakistan	156483000
Russian Federation	146934000
Bangladesh	129155000
Japan	126714000
Nigeria	111506000
Mexico	98881000
Germany	82164700
Vietnam	79832000
Philippines	75967000
Egypt	68470000
Iran	67702000
Turkey	66591000
Ethiopia	62565000
Thailand	61399000
United Kingdom	59623400

17. **Capital Cities Population Comparison:** *Scenario:* A statistical analysis firm is examining population distributions between capital cities worldwide. You're tasked with comparing the populations of capital cities from different countries to identify trends and patterns in urban demographics.

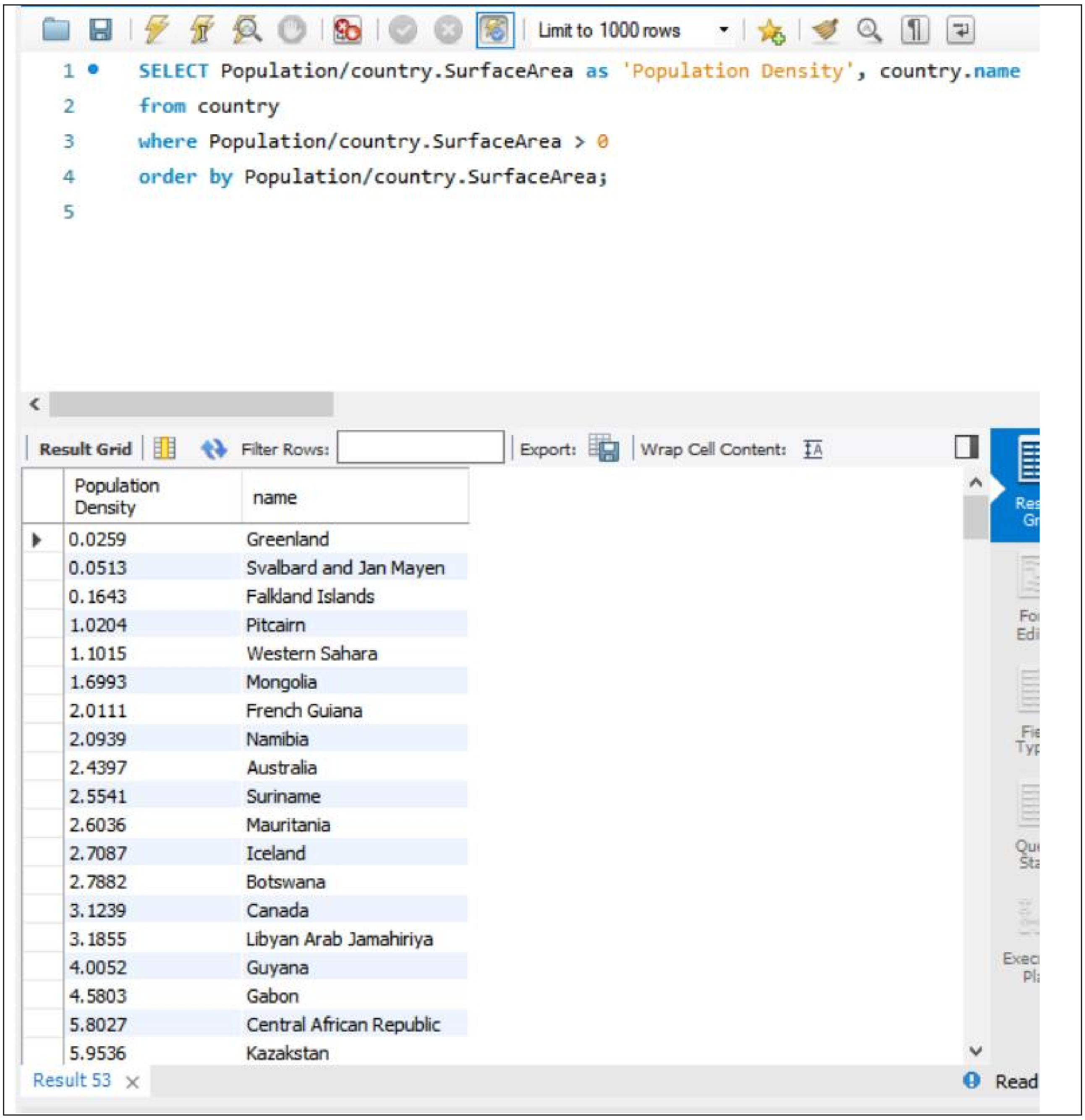


```sql
SELECT city.name as Capital, city.Population, country.name as Country
from city
join country
on country.Capital = city.ID
order by Population DESC;
```

Capital	Population	Country
Seoul	9981619	South Korea
Jakarta	9604900	Indonesia
Ciudad de México	8591309	Mexico
Moscow	8389200	Russian Federation
Tokyo	7980230	Japan
Peking	7472000	China
London	7285000	United Kingdom
Cairo	6789479	Egypt
Teheran	6758845	Iran
Lima	6464693	Peru
Bangkok	6320174	Thailand
Santafé de Bogotá	6260862	Colombia
Kinshasa	5064000	Congo, The Demo...
Santiago de Chile	4703954	Chile
Baghdad	4336000	Iraq
Singapore	4017733	Singapore
Dhaka	3612850	Bangladesh
Berlin	3386667	Germany
Rangoon (Yangon)	3361700	Myanmar
Riyadh	3324000	Saudi Arabia

Result 48

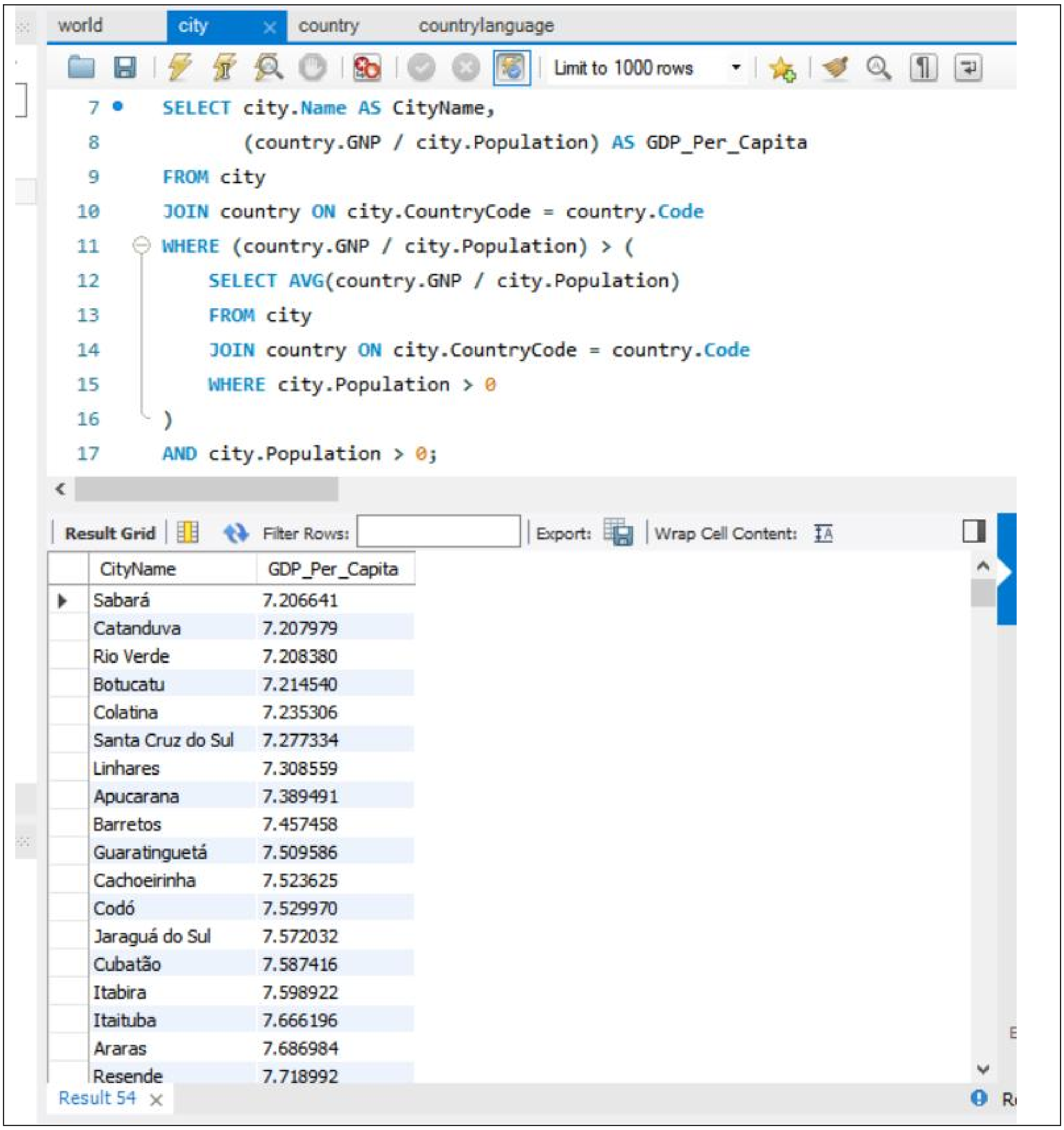
18. **Countries with Low Population Density:** *Scenario:* An agricultural research institute is studying countries with low population densities for potential agricultural development projects. You're tasked with identifying countries with sparse populations from the database to support the institute's research efforts.



```sql
SELECT Population/country.SurfaceArea as 'Population Density', country.name
from country
where Population/country.SurfaceArea > 0
order by Population/country.SurfaceArea;
```

Population Density	name
0.0259	Greenland
0.0513	Svalbard and Jan Mayen
0.1643	Falkland Islands
1.0204	Pitcairn
1.1015	Western Sahara
1.6993	Mongolia
2.0111	French Guiana
2.0939	Namibia
2.4397	Australia
2.5541	Suriname
2.6036	Mauritania
2.7087	Iceland
2.7882	Botswana
3.1239	Canada
3.1855	Libyan Arab Jamahiriya
4.0052	Guyana
4.5803	Gabon
5.8027	Central African Republic
5.9536	Kazakstan

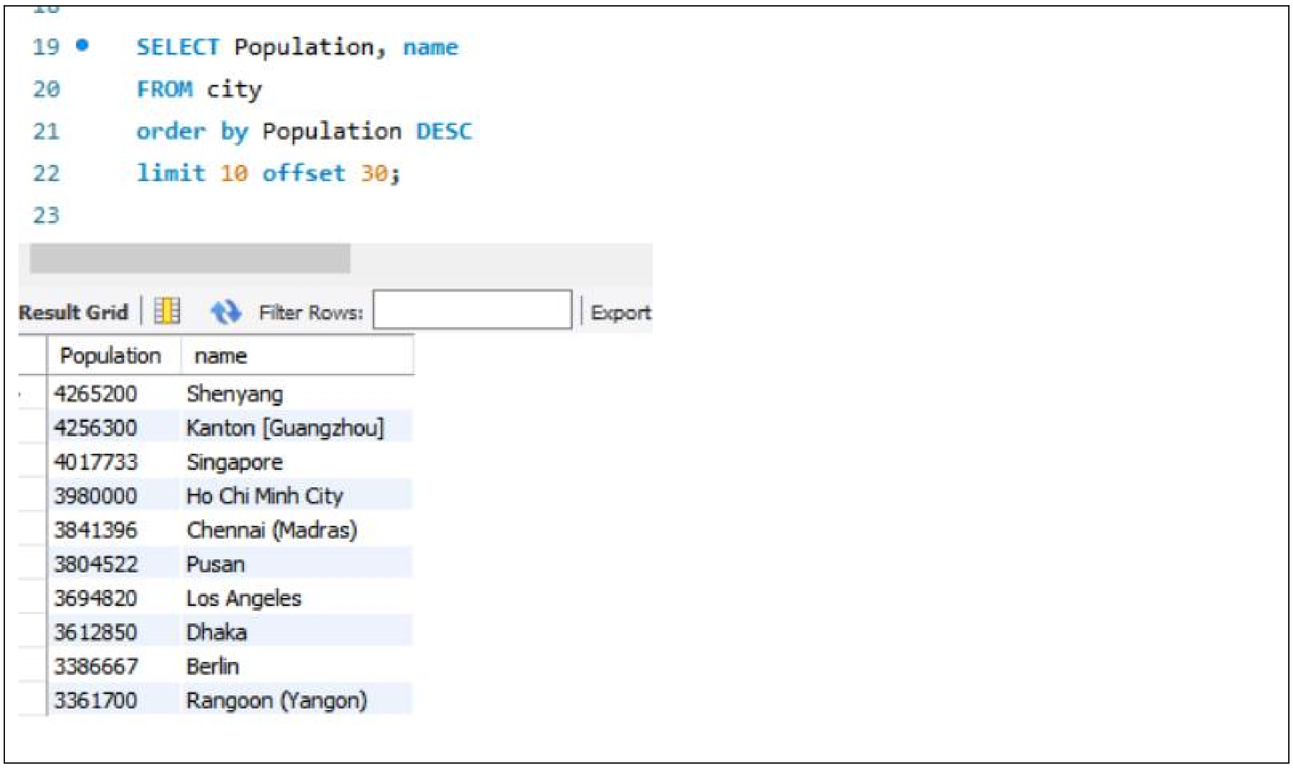
19. **Cities with High GDP per Capita:** *Scenario:* An economic consulting firm is analysing cities with high GDP per capita for investment opportunities. You're tasked with identifying cities with above-average GDP per capita from the database to assist the firm in identifying potential investment destinations.



```sql
SELECT city.Name AS CityName,
       (country.GNP / city.Population) AS GDP_Per_Capita
FROM city
JOIN country ON city.CountryCode = country.Code
WHERE (country.GNP / city.Population) > (
    SELECT AVG(country.GNP / city.Population)
    FROM city
    JOIN country ON city.CountryCode = country.Code
    WHERE city.Population > 0
)
AND city.Population > 0;
```

CityName	GDP_Per_Capita
Sabará	7.206641
Catanduva	7.207979
Rio Verde	7.208380
Botucatu	7.214540
Colatina	7.235306
Santa Cruz do Sul	7.277334
Linhares	7.308559
Apucarana	7.389491
Barretos	7.457458
Guaratinguetá	7.509586
Cachoeirinha	7.523625
Codó	7.529970
Jaraguá do Sul	7.572032
Cubatão	7.587416
Itabira	7.598922
Itaituba	7.666196
Araras	7.686984
Resende	7.718992

20. **Display Columns with Limit (Rows 31-40):** *Scenario:* A market research firm requires detailed information on cities beyond the top rankings for a comprehensive analysis. You're tasked with providing data on cities ranked between 31st and 40th by population to ensure a thorough understanding of urban demographics.



```sql
SELECT Population, name
FROM city
order by Population DESC
limit 10 offset 30;
```

Population	name
4265200	Shenyang
4256300	Kanton [Guangzhou]
4017733	Singapore
3980000	Ho Chi Minh City
3841396	Chennai (Madras)
3804522	Pusan
3694820	Los Angeles
3612850	Dhaka
3386667	Berlin
3361700	Rangoon (Yangon)

Course Notes

It is recommended to take notes from the course, use the space below to do so, or use the revision guide shared with the class:

We have included a range of additional links to further resources and information that you may find useful, these can be found within your revision guide.

END OF WORKBOOK

Please check through your work thoroughly before submitting and update the table of contents if required.

Please send your completed work booklet to your trainer.