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 field (or combination of fields) that uniquely identifies each record in a table. • It must be unique and cannot be missing (i.e. NULL). • Every table should have one. E.g. In a Students table, Student_ID can be the primary key – no two students can share the same ID.
How does this differ from a secondary key?	A **secondary key** (also called an alternate key) is any field (or set of fields) that can also uniquely identify a record but is not the primary key. • It is used for indexing or searching but not as the main identifier. E.g. In the same Students table, Email might be a secondary key – it should also be unique, but it's not the official primary key.
How are primary and foreign keys related?	• A **foreign key** is a field in one table that **references the primary key** in another table. • It **creates a link** between two tables and enforces referential integrity. **Example:** • Students.Guardian_ID is a **foreign key** referencing Guardians.Guardian_ID (the primary key). • This ensures every student is linked to a valid guardian.
Provide a real-world example of a one-to-one relationship	Each passport is assigned to one person, and each person can hold only one valid passport.
Provide a real-world example of a	One teacher can teach many students, but each student has only one main teacher.

one-to-many relationship	
Provide a real-world example of a many-to-many relationship	A student can enroll in many subjects, and each subject can have many students.

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?

Feature	Relational DB	Non-Relational DB
Structure	Tables (rows & columns)	Documents, key-value, graphs, wide-columns
Schema	Fixed schema (strict)	Dynamic schema (flexible)
Relationships	Strong use of joins and foreign keys	Usually denormalized, embeds related data
Scalability	Vertical scaling (bigger server)	Horizontal scaling (more servers)
Examples	MySQL, PostgreSQL, SQL Server	MongoDB, Redis, Cassandra, Neo4j
Best For	Structured, consistent data	Semi-structured/unstructured or fast-evolving data

What type of data would benefit off the non-relational model?

Why?

Non-relational databases are ideal when:

1. **The data is unstructured or semi-structured**
   Example: JSON, multimedia, logs.
   Relational databases require structured data with predefined columns and types.

   Unstructured data like images, videos, or JSON objects doesn't fit cleanly into rows and columns.

   Non-relational databases (like MongoDB or Couchbase) let you store documents or binary data without forcing a rigid schema.

2. **Rapid development with frequent schema changes**
   You don't need to constantly migrate tables.

In fast-paced projects (startups, MVPs), the data model evolves constantly — new features might require adding or removing fields.

Relational databases require schema migrations for every structural change, which can be time-consuming and error-prone.

Document databases allow you to add new fields to documents without changing a global table definition.

3. **High-speed read/write is more important than strict consistency**
   Use cases like caching don't need guaranteed consistency — it's okay if some nodes are slightly behind.

   Relational databases enforce strict ACID compliance (especially consistency), which can slow down performance.

   Non-relational databases (like Redis or DynamoDB) offer eventual consistency, prioritizing speed and availability.

4. **Massive scale-out applications**
   Like IoT data ingestion, social media feeds, or event logs. These systems generate huge volumes of data from many sources in real time.

   Relational databases struggle to scale horizontally — they scale better vertically (which hits hardware limits quickly).

   Non-relational systems are designed for horizontal scaling, distributing data across multiple nodes.

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	Joins a table to itself to find relationships within the same table. E.g. Finding employees who report to the same manager in an employee table. SELECT A.name AS employee, B.name AS manager FROM Employees A JOIN Employees B ON A.manager_id = B.employee_id; $R \bowtie R = \{ (r1, r2) \mid r1, r2 \in R \text{ and } r1.key = r2.foreign\_key \}$
Right outer join	Returns all records from the right table, and matched records from the left table. E.g. Getting all customers when the customer table is on the right. SELECT Orders.order_id, Customers.name FROM Orders RIGHT JOIN Customers ON Orders.customer_id = Customers.customer_id; $A \bowtie B = \{ (a, b) \mid a \in A, b \in B, a.key = b.key \} \cup \{ (NULL, b) \mid b \in B \text{ and no } a \in A \text{ such that } a.key = b.key \}$
Full outer join	Returns all records from both tables, matching where possible. Fills in NULL where there is no match. E.g. Listing all products and orders, including products never ordered and orders with missing product info. SELECT Products.name, Orders.order_id FROM Products FULL JOIN Orders ON Products.product_id = Orders.product_id; $A \bowtie\!\!\!\!\bowtie B = (A \bowtie B) \cup (A \bowtie B)$
Inner join	Returns only the rows with matching values in both tables.

	I.e. Intersection of 2 sets E.g. Finding customers who have placed orders. SELECT Customers.name, Orders.order_id FROM Customers INNER JOIN Orders ON Customers.customer_id = Orders.customer_id; $A \cap B = \{ x \in A \text{ and } y \in B \mid x.key = y.key \}$
Cross join	Returns the Cartesian product of both tables. Every row of A combined with every row of B. E.g. Generating every possible combination of colours and sizes for a product. SELECT Colours.name, Sizes.size FROM Colours CROSS JOIN Sizes; $A \times B = \{ (a, b) \mid a \in A, b \in B \}$
Left outer join	Returns all records from the left table and matched records from the right. E.g. Listing all customers and any orders they may have, including customers without orders. SELECT Customers.name, Orders.order_id FROM Customers LEFT JOIN Orders ON Customers.customer_id = Orders.customer_id; $A \bowtie B = \{ (a, b) \mid a \in A, b \in B, a.key = b.key \} \cup \{ (a, NULL) \mid a \in A \text{ and no } b \in B \text{ such that } a.key = b.key \}$

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.

Please write your 500-word essay here	The data stored will include Inventory data (product names, categories, quantities, suppliers, and prices), Sales data (transaction dates, items sold, quantities, customer and totals), Customer info (names, contact details, and loyalty program participation) & Loyalty program info (points earned, redeemed, and membership status) The primary users will be store's staff and manager. Staff need to add and retrieve sales and inventory data quickly, while the manager needs reports on sales trends, inventory levels, and customer engagement. To structure the database tables efficiently, I would create the tables: • Products – Stores all product info • Customers – Stores customer profiles and loyalty data • Sales – Records each sale transaction • Sale_Items – Detailed item descriptions in each sale • Loyalty – Tracks the points, status and redemption history of customers One Sale is linked to One Customer One Sale can include Many Sale_Items, which are tied to One Product One Customer is linked to One Loyalty profile E.g. Creating the tables: CREATE TABLE Products (product_id SERIAL PRIMARY KEY, name VARCHAR(100), category VARCHAR(50), price DECIMAL(10,2), stock_quantity INT); CREATE TABLE Customers (customer_id SERIAL PRIMARY KEY, name VARCHAR(100),

```
    email VARCHAR(100),
    phone VARCHAR(20)
);

CREATE TABLE Loyalty (
    customer_id INT PRIMARY KEY REFERENCES
Customers(customer_id),
    points INT DEFAULT 0
);

CREATE TABLE Sales (
    sale_id SERIAL PRIMARY KEY,
    customer_id INT REFERENCES Customers(customer_id),
    sale_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    total_amount DECIMAL(10,2)
);

CREATE TABLE SaleItems (
    sale_item_id SERIAL PRIMARY KEY,
    sale_id INT REFERENCES Sales(sale_id),
    product_id INT REFERENCES Products(product_id),
    quantity INT,
    item_price DECIMAL(10,2)
);
```

E.g. Inputting initial records:

```
INSERT INTO Products (name, category, price, stock_quantity)
VALUES ('Milk', 'Dairy', 1.99, 50);

INSERT INTO Customers (name, email, phone)
VALUES ('John Smith', 'john98@example.com', '0121 555 5555');

INSERT INTO Loyalty (customer_id, points)
VALUES (1, 20);

INSERT INTO Sales (customer_id, total_amount)
VALUES (1, 5.97);
```

INSERT INTO SaleItems (sale_id, product_id, quantity, item_price) VALUES (1, 1, 3, 1.99);

To ensure the database remains accurate and up to date, I would add constraints and validation, i.e. using NOT NULL, UNIQUE, CHECK, etc. to prevent the wrong data inputted

E.g. To prevent using negative values for prices:

ALTER TABLE Products
ADD CONSTRAINT check_stock_positive CHECK (stock_quantity >= 0);

I would also add automated triggers – which can update stock levels or loyalty points when a sale is recorded – and regularly running reports to detect anomalies. E.g. products with stock but no sales, or duplicate customer emails.

To handle backups and data security, I would set up daily backups of the entire database – either hourly (incremental), daily (full) or weekly (off-site) – and test the restore process to ensure the backups work as intended. Also, keeping track with version control and changelogs (e.g. of schema changes and SQL scripts) can help look back at previous queries if something goes wrong.

For data security, user roles and permissions can be created for cashiers and administrators:

CREATE ROLE cashier NOINHERIT;
GRANT SELECT, INSERT ON Sales, SaleItems TO cashier;

Aswell as encrypting/masking sensitive data (E.g. customer data) to limit access, and regularly updating the database system with security patches.

Day 4: Task 2: SQL Practical

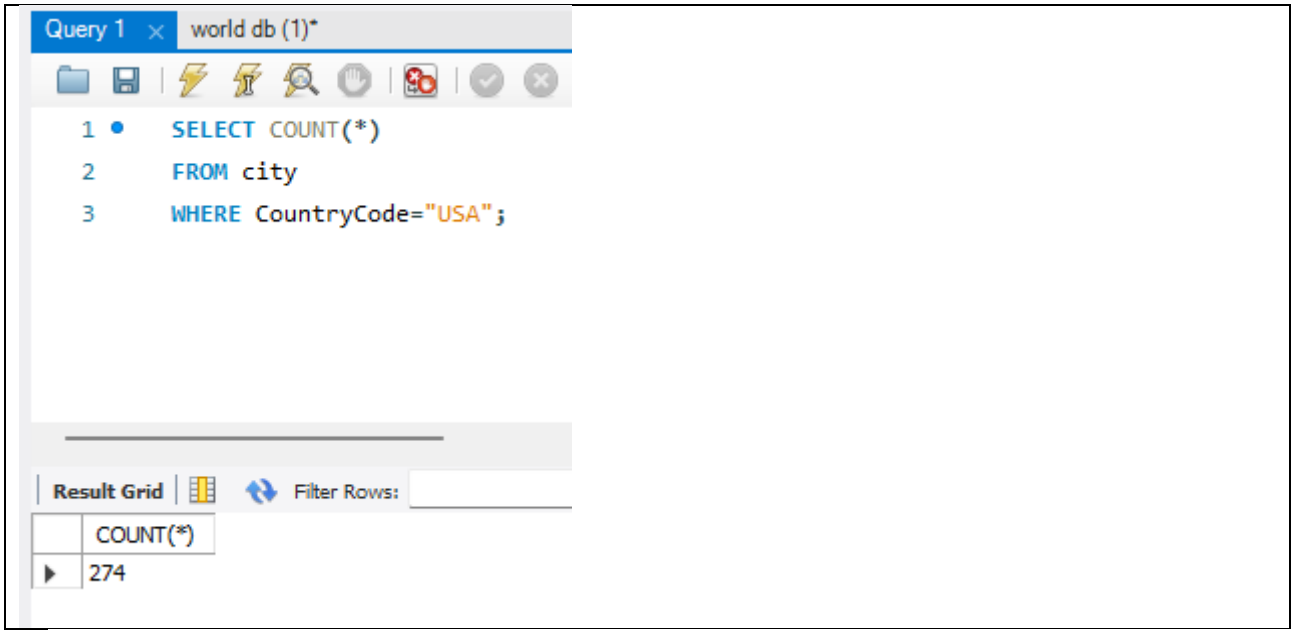
In your groups, work together to answer the below questions. It may be of benefit if one of you shares your screen with the group and as a team answer / take screen shots from there.

Setting up the database:

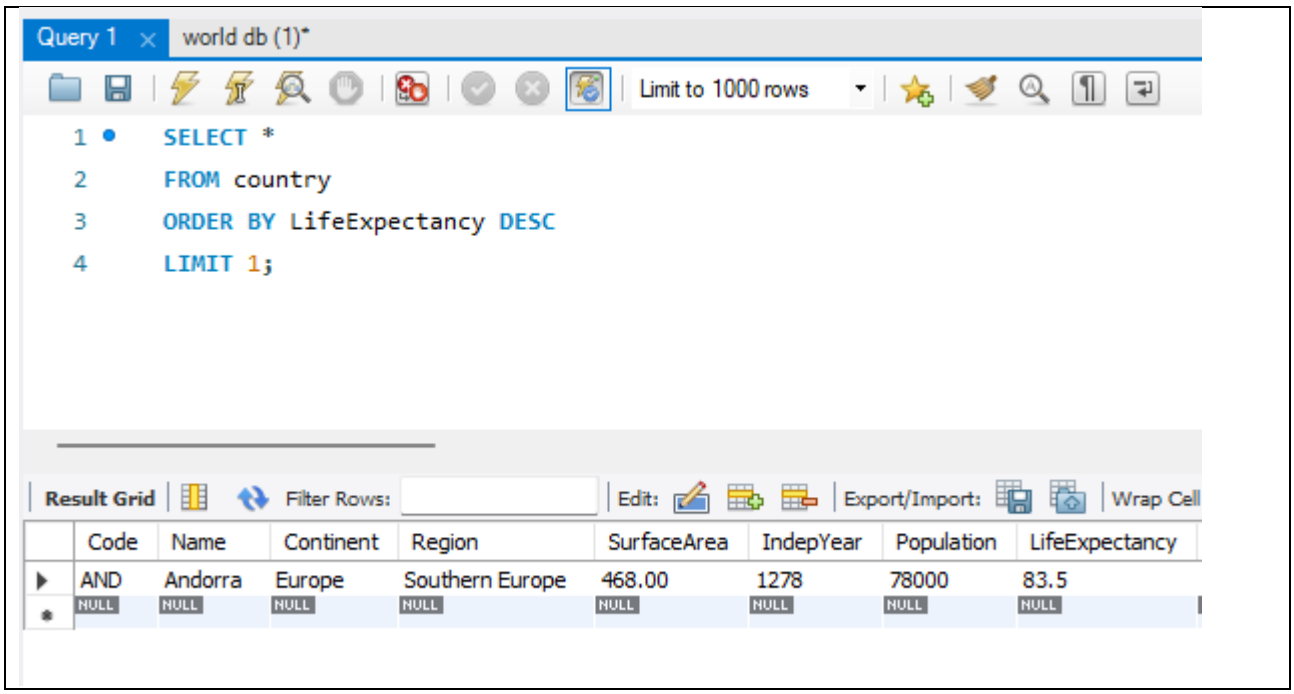
1. **Download world_db(1) here**
2. **Follow each step to create your database here**

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

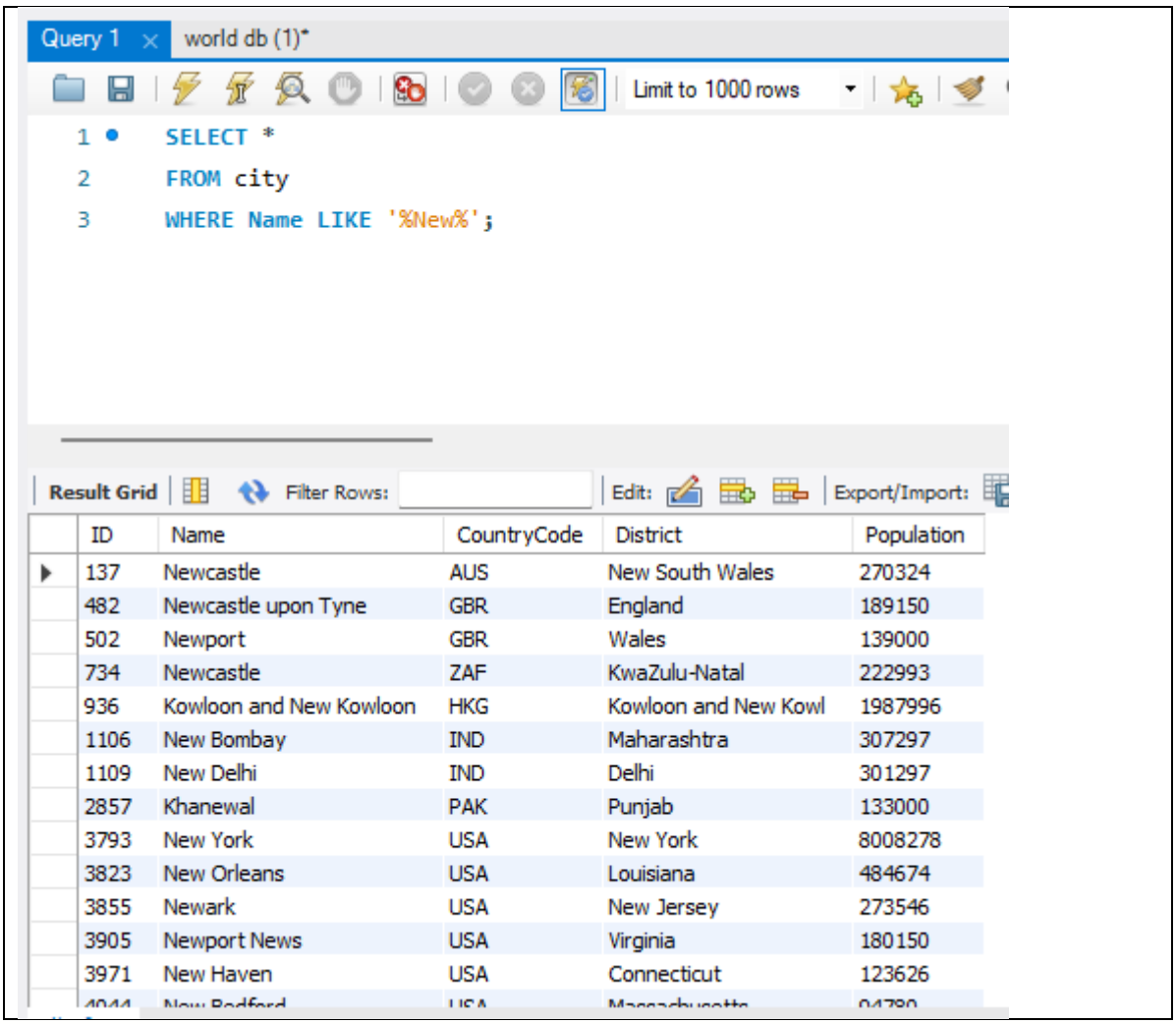
1. **Count Cities in USA:** *Scenario:* You've been tasked with conducting a demographic analysis of cities in the United States. Your first step is to determine the total number of cities within the country to provide a baseline for further analysis.



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.

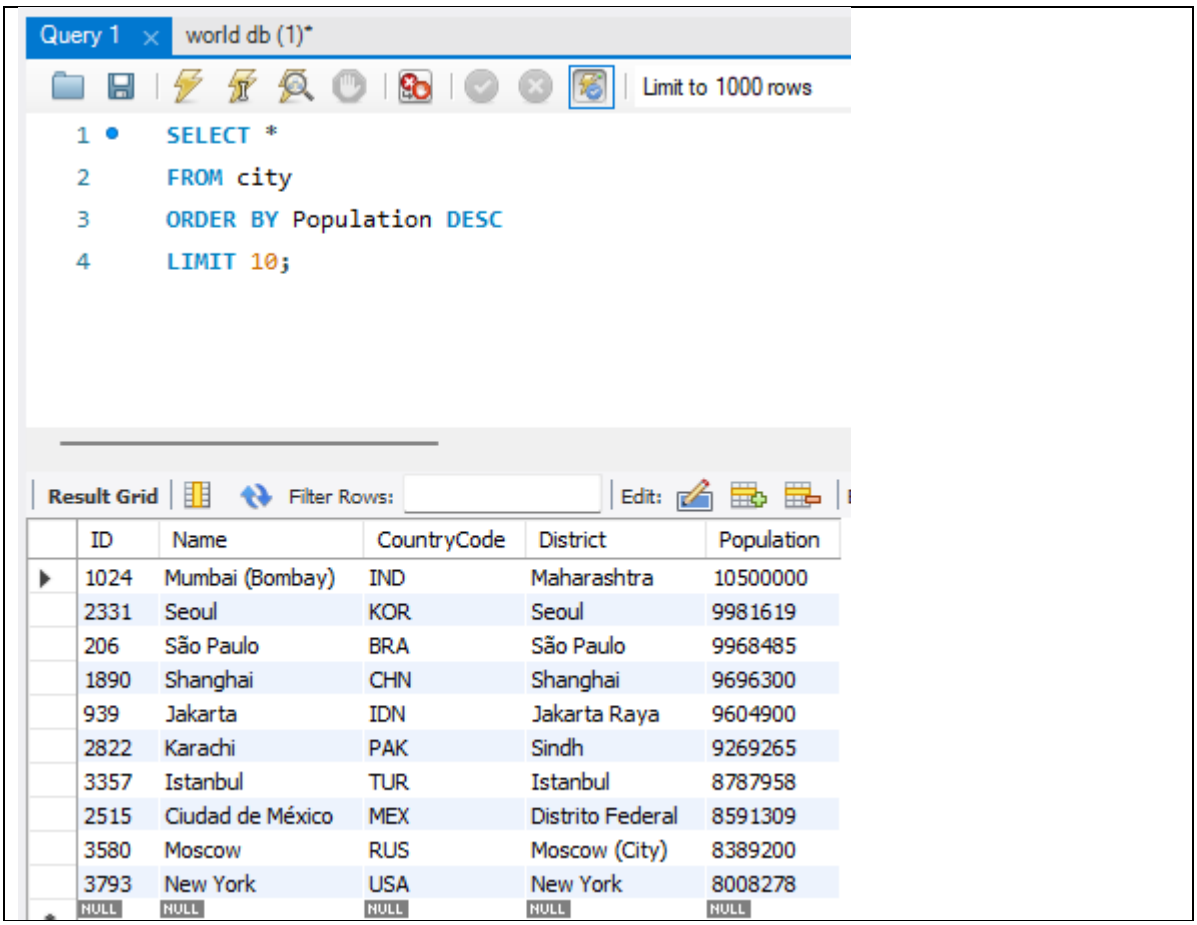


3. **"New Year Promotion: Featuring Cities with 'New :** *Scenario:* In anticipation of the upcoming New Year, your travel agency is gearing up for a special promotion featuring cities with names including the word 'New'. You're tasked with swiftly compiling a list of all cities from around the world.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

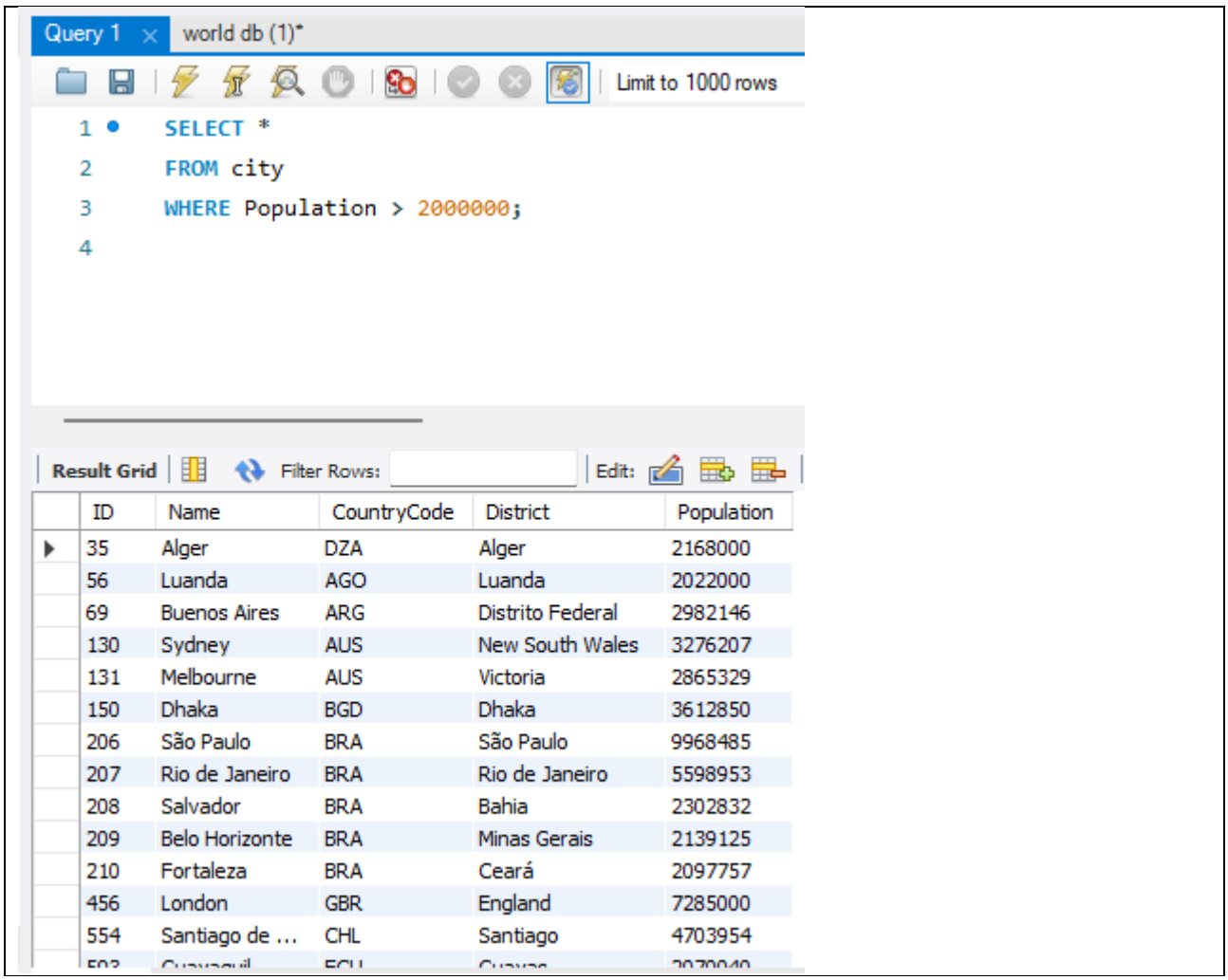


ID	Name	CountryCode	District	Population
137	Newcastle	AUS	New South Wales	270324
482	Newcastle upon Tyne	GBR	England	189150
502	Newport	GBR	Wales	139000
734	Newcastle	ZAF	KwaZulu-Natal	222993
936	Kowloon and New Kowloon	HKG	Kowloon and New Kowl	1987996
1106	New Bombay	IND	Maharashtra	307297
1109	New Delhi	IND	Delhi	301297
2857	Khanewal	PAK	Punjab	133000
3793	New York	USA	New York	8008278
3823	New Orleans	USA	Louisiana	484674
3855	Newark	USA	New Jersey	273546
3905	Newport News	USA	Virginia	180150
3971	New Haven	USA	Connecticut	123626

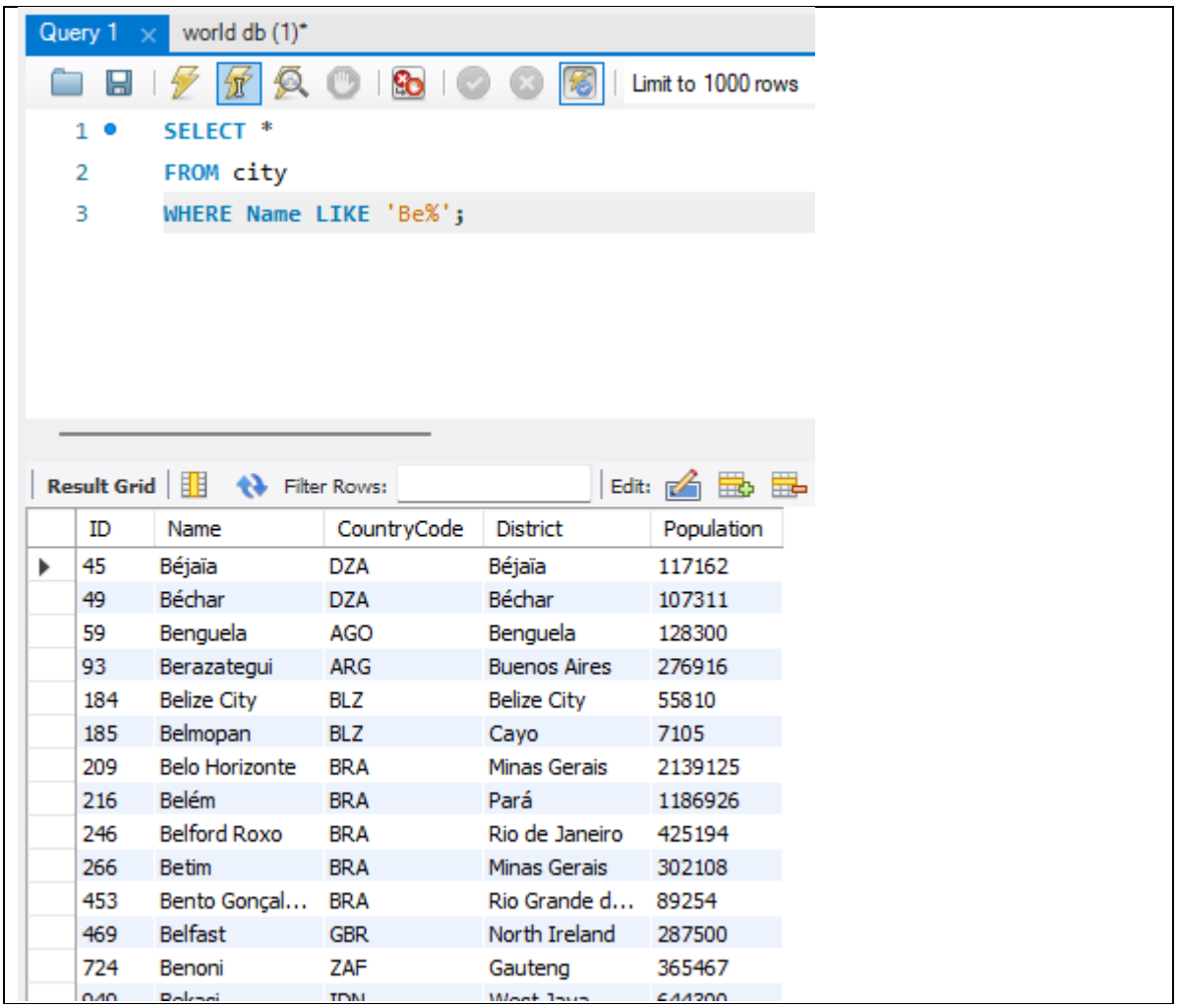
4. **Display Columns with Limit (First 10 Rows):** *Scenario:* You're tasked with providing a brief overview of the most populous cities in the world. To keep the report concise, you're instructed to list only the first 10 cities by population from the database.



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.

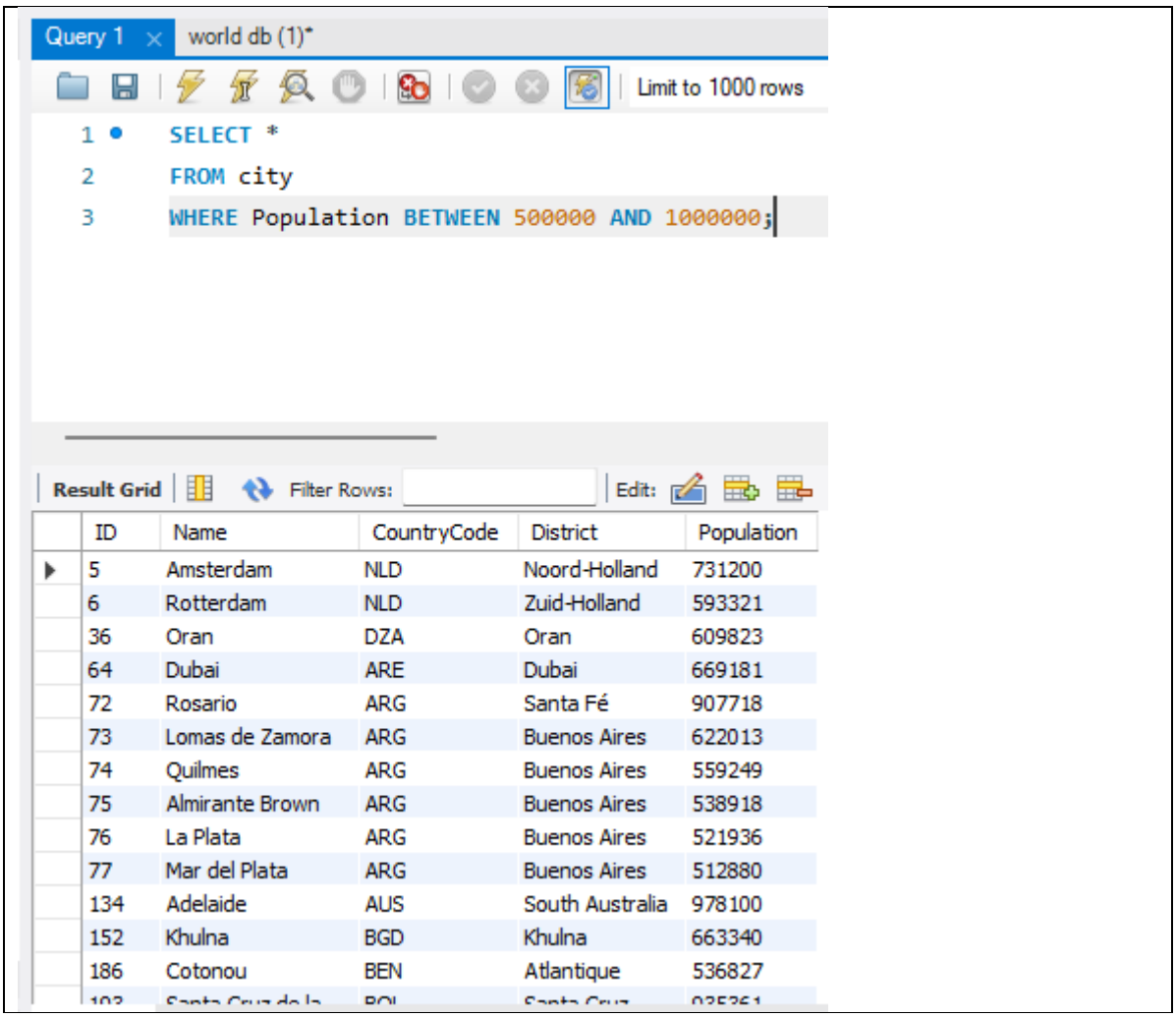


6. **Cities Beginning with 'Be' Prefix:** *Scenario:* A travel blogger is planning a series of articles featuring cities with unique names. You're tasked with compiling a list of cities from the database that start with the prefix 'Be' to assist in the blogger's content creation process.

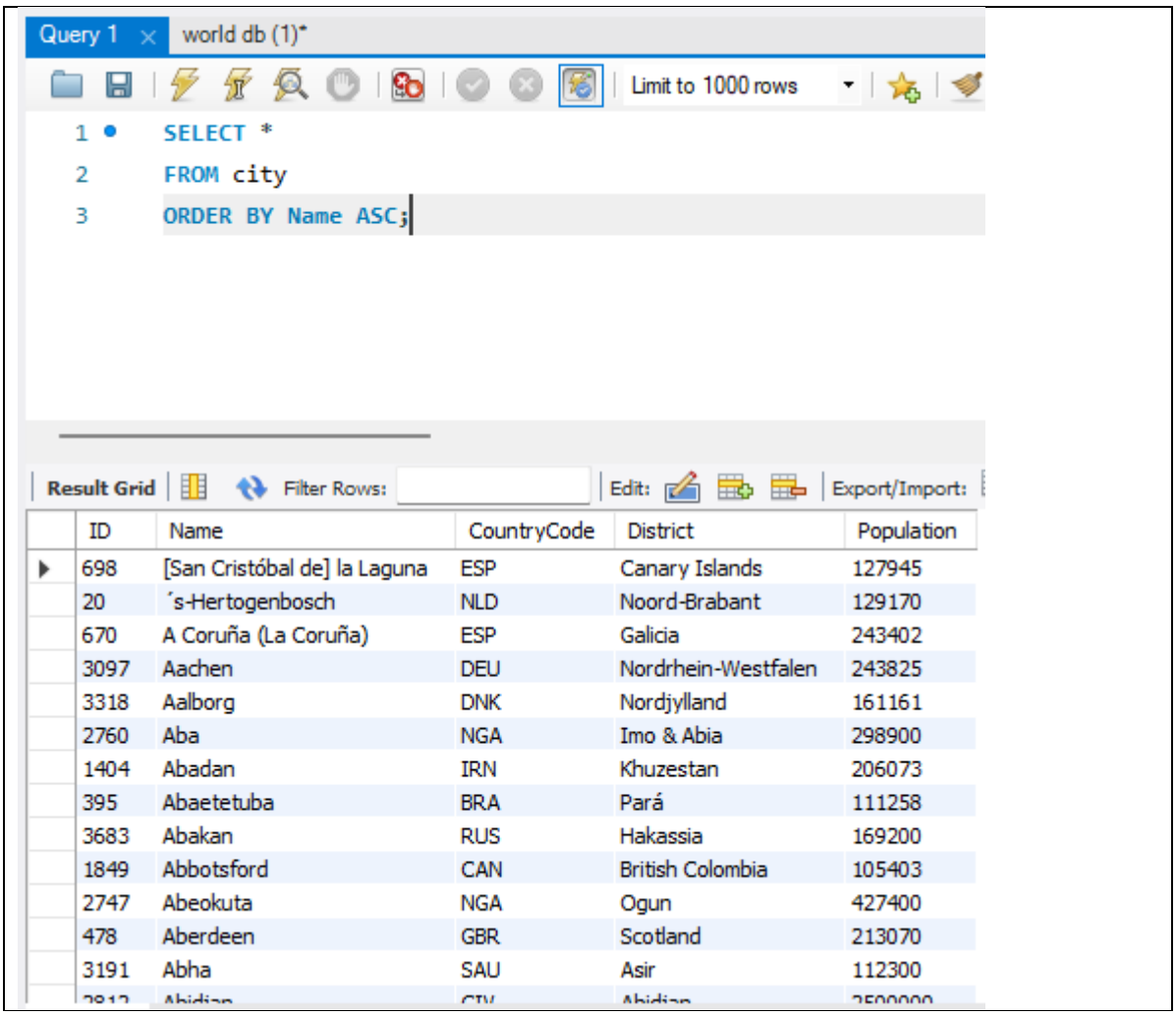


ID	Name	CountryCode	District	Population
45	Béjaïa	DZA	Béjaïa	117162
49	Béchar	DZA	Béchar	107311
59	Benguela	AGO	Benguela	128300
93	Berazategui	ARG	Buenos Aires	276916
184	Belize City	BLZ	Belize City	55810
185	Belmopan	BLZ	Cayo	7105
209	Belo Horizonte	BRA	Minas Gerais	2139125
216	Belém	BRA	Pará	1186926
246	Belford Roxo	BRA	Rio de Janeiro	425194
266	Betim	BRA	Minas Gerais	302108
453	Bento Gonçal...	BRA	Rio Grande d...	89254
469	Belfast	GBR	North Ireland	287500
724	Benoni	ZAF	Gauteng	365467

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.

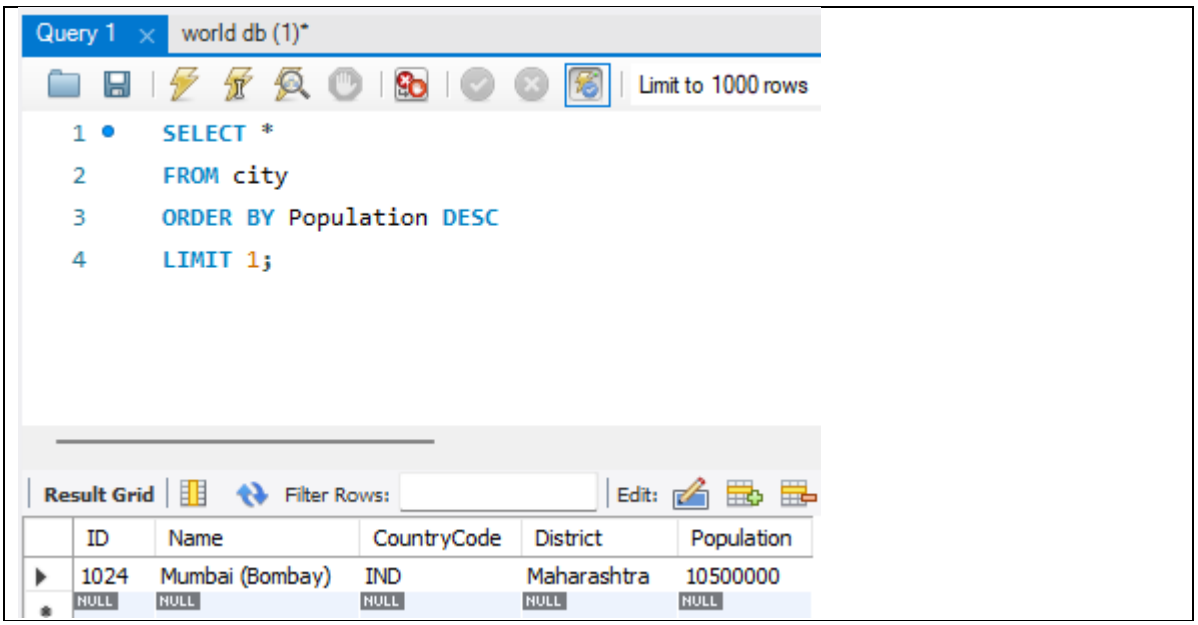


8. **Display Cities Sorted by Name in Ascending Order:** *Scenario:* A geography teacher is preparing a lesson on alphabetical order using city names. You're tasked with providing a sorted list of cities from the database in ascending order by name to support the lesson plan.

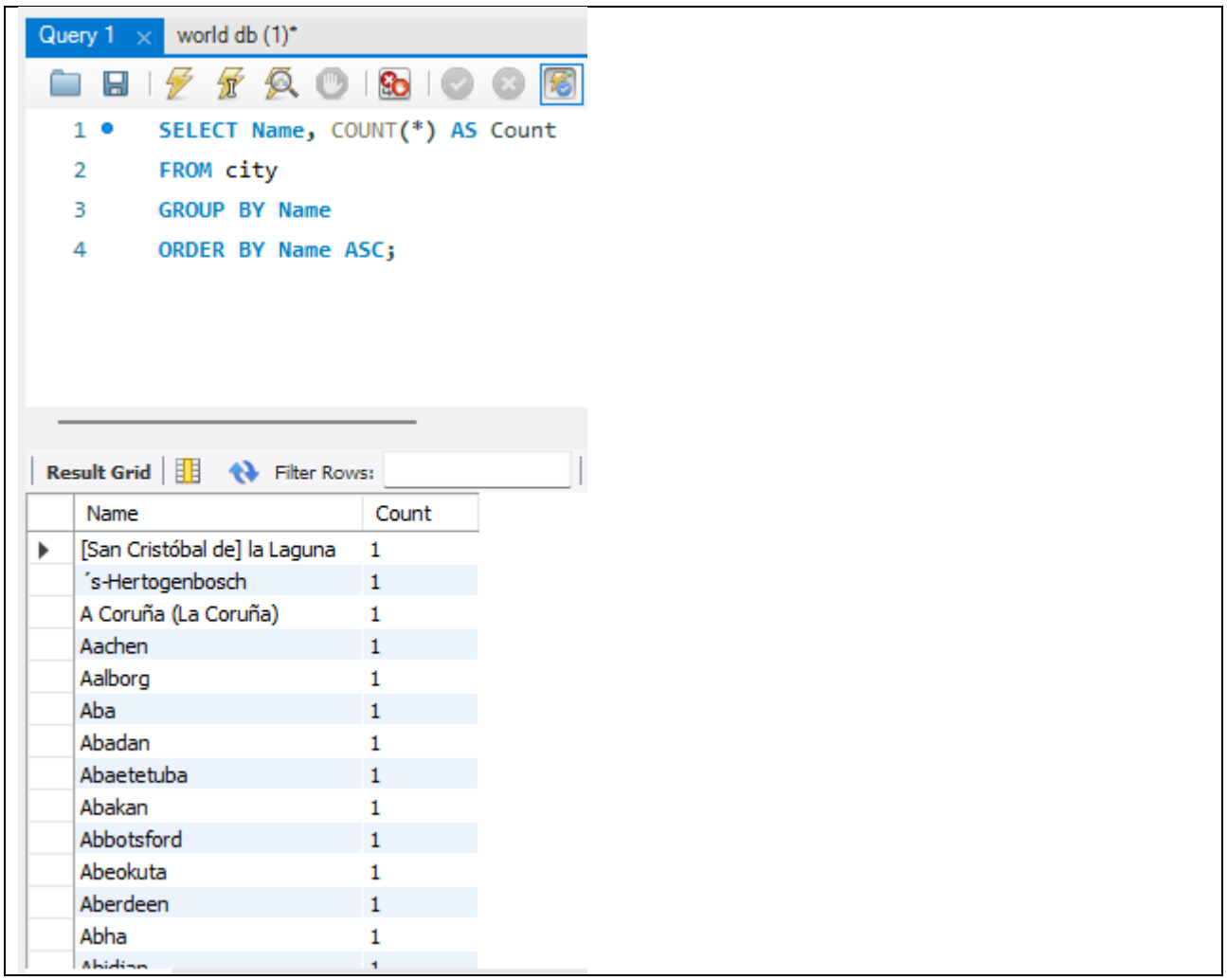


ID	Name	CountryCode	District	Population
698	[San Cristóbal de] la Laguna	ESP	Canary Islands	127945
20	´s-Hertogenbosch	NLD	Noord-Brabant	129170
670	A Coruña (La Coruña)	ESP	Galicia	243402
3097	Aachen	DEU	Nordrhein-Westfalen	243825
3318	Aalborg	DNK	Nordjylland	161161
2760	Aba	NGA	Imo & Abia	298900
1404	Abadan	IRN	Khuzestan	206073
395	Abaetetuba	BRA	Pará	111258
3683	Abakan	RUS	Hakassia	169200
1849	Abbotsford	CAN	British Colombia	105403
2747	Abeokuta	NGA	Ogun	427400
478	Aberdeen	GBR	Scotland	213070
3191	Abha	SAU	Asir	112300

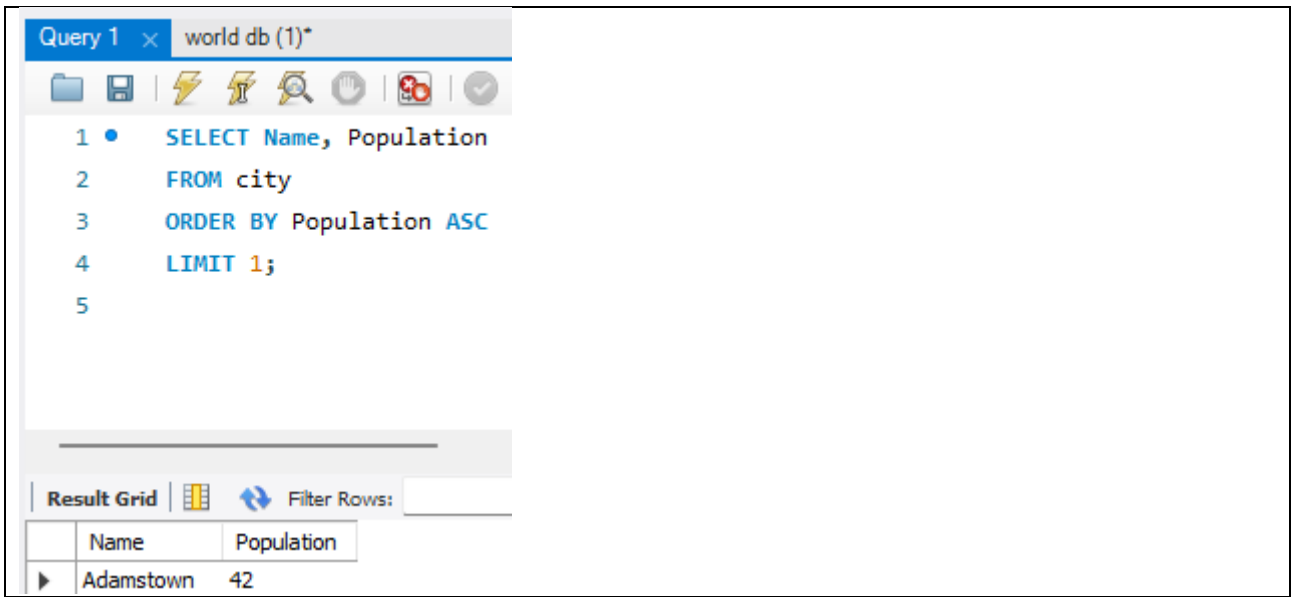
9. **Most Populated City:** *Scenario:* A real estate investment firm is interested in cities with significant population densities for potential development projects. You're tasked with identifying the most populated city from the database to guide their investment decisions and strategic planning.



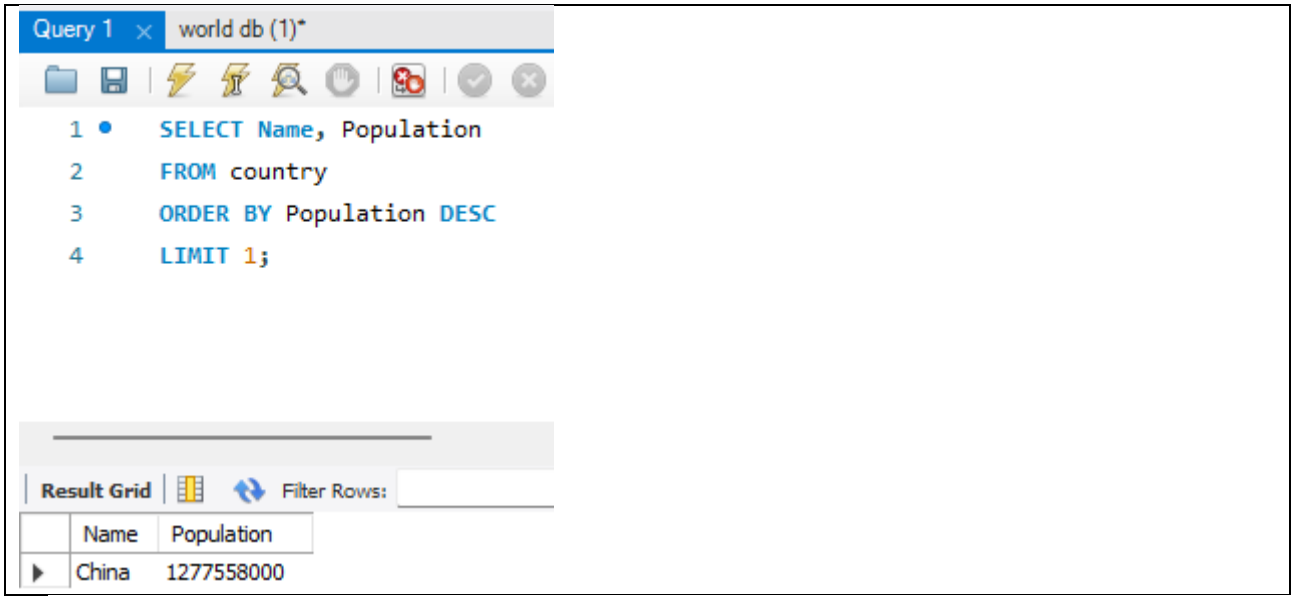
10. **City Name Frequency Analysis: Supporting Geography Education** *Scenario*: In a geography class, students are learning about the distribution of city names around the world. The teacher, in preparation for a lesson on city name frequencies, wants to provide students with a list of unique city names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



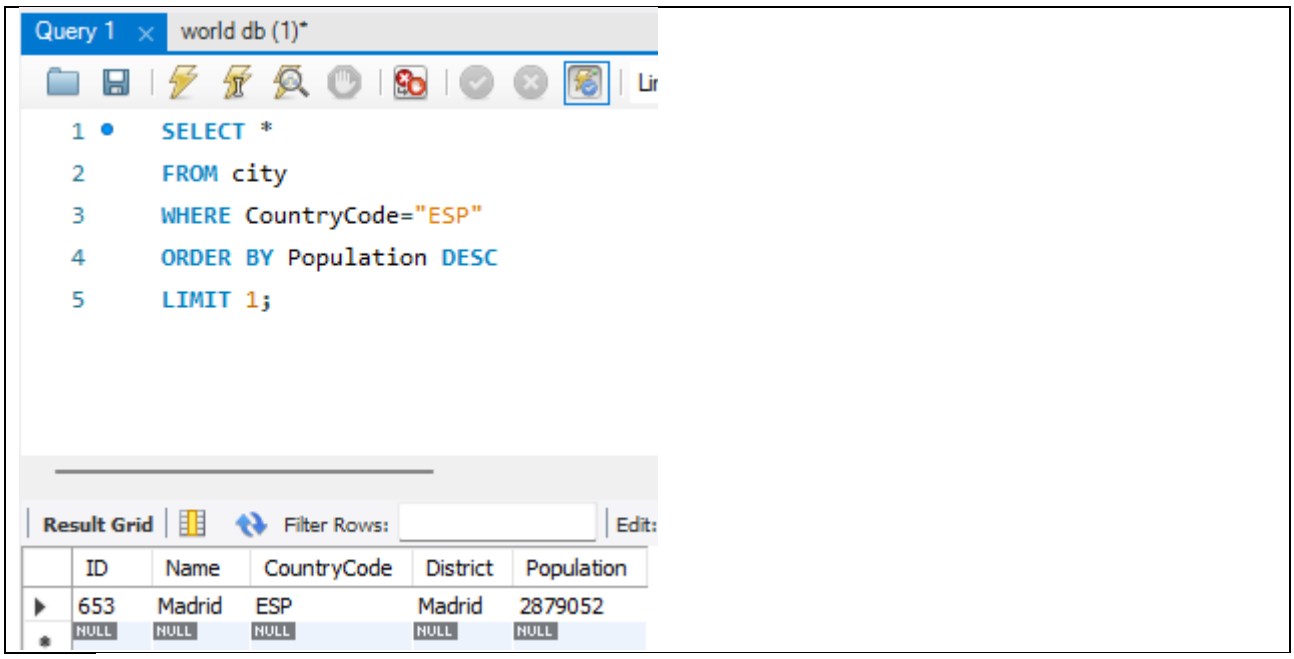
11. **City with the Lowest Population:** *Scenario:* A census bureau is conducting an analysis of urban population distribution. You're tasked with identifying the city with the lowest population from the database to provide a comprehensive overview of demographic trends.



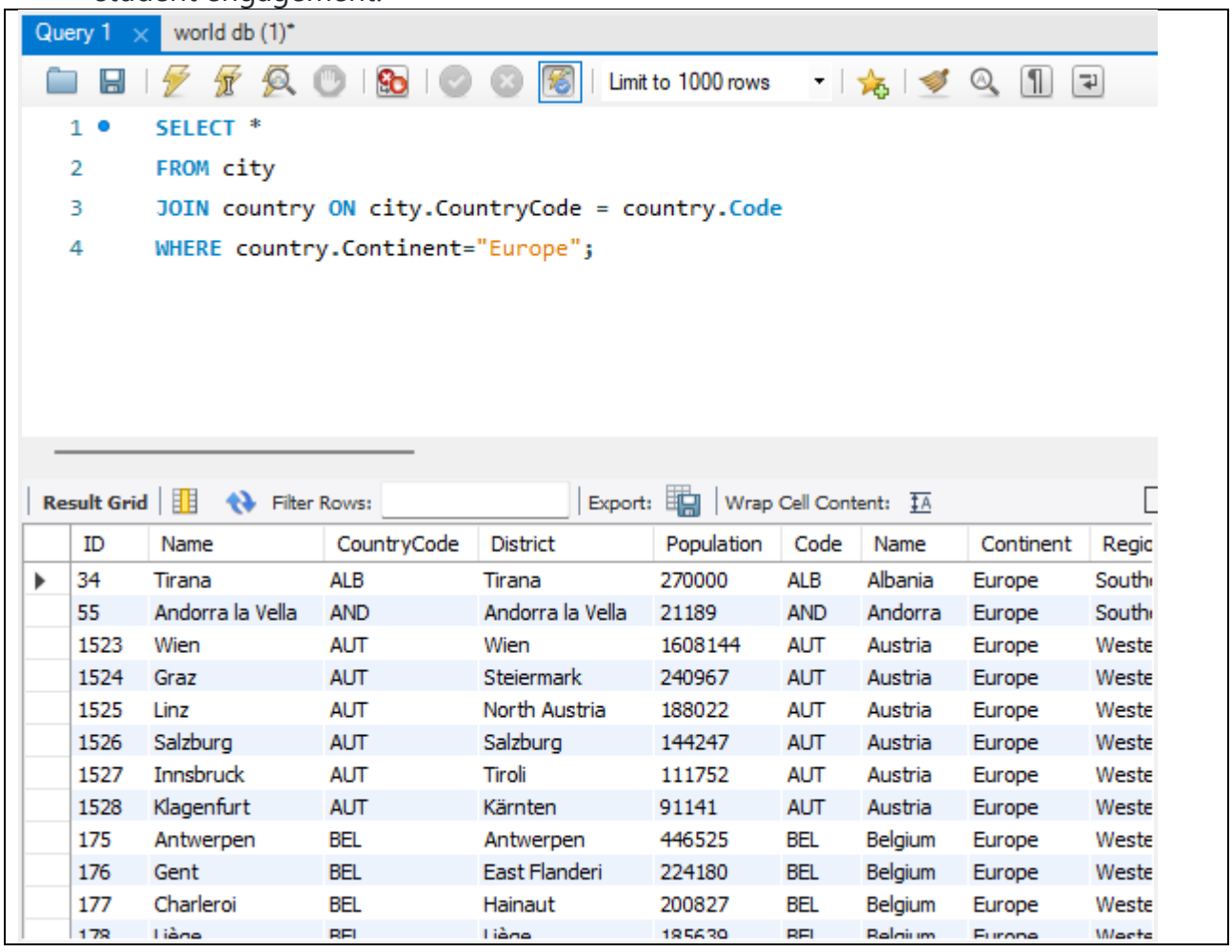
12. **Country with Largest Population:** *Scenario:* A global economic research institute requires data on countries with the largest populations for a comprehensive analysis. You're tasked with identifying the country with the highest population from the database to provide valuable insights into demographic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



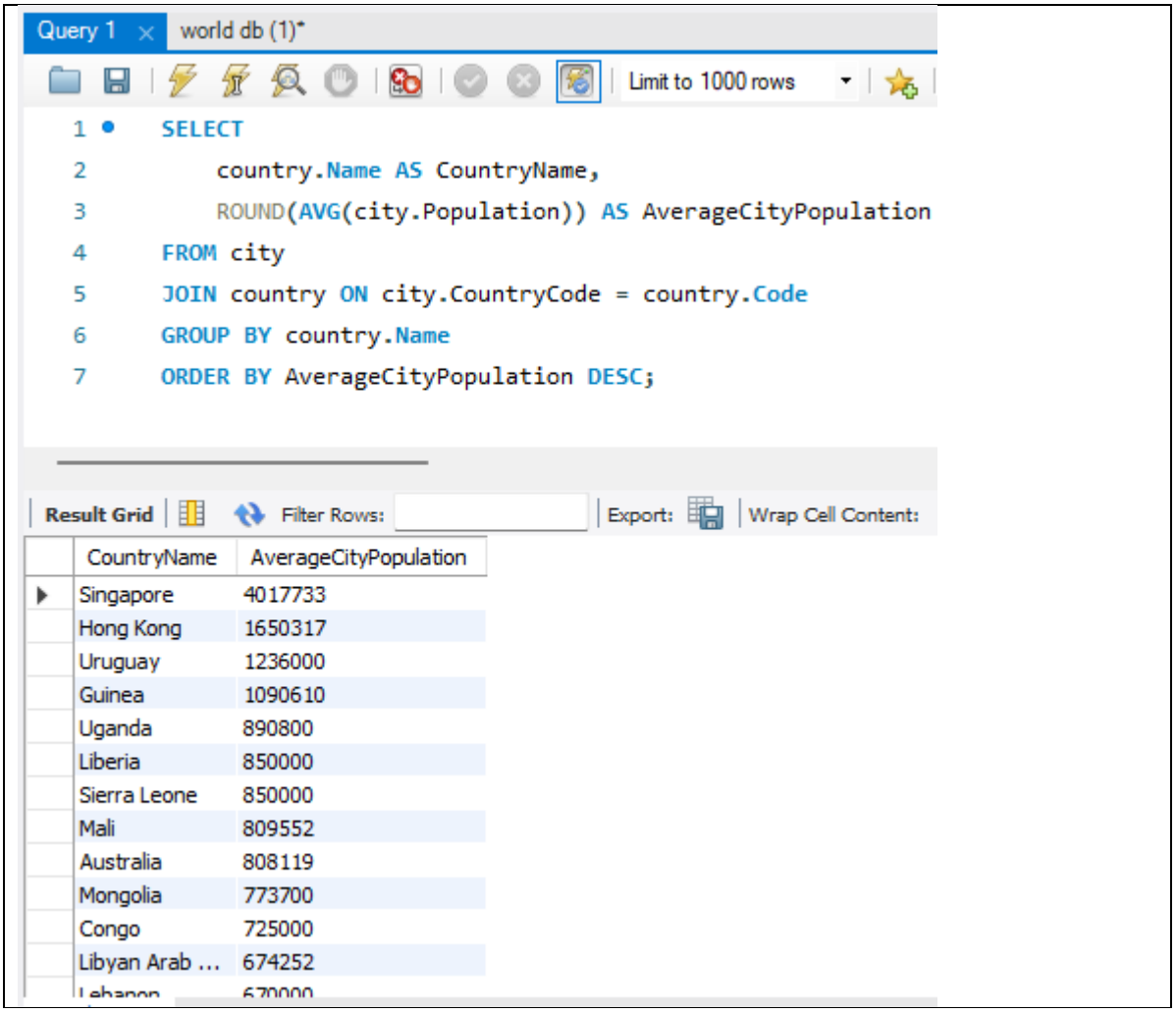
13. **Capital of Spain:** *Scenario:* A travel agency is organising tours across Europe and needs accurate information on capital cities. You're tasked with identifying the capital of Spain from the database to ensure itinerary accuracy and provide travellers with essential destination information.



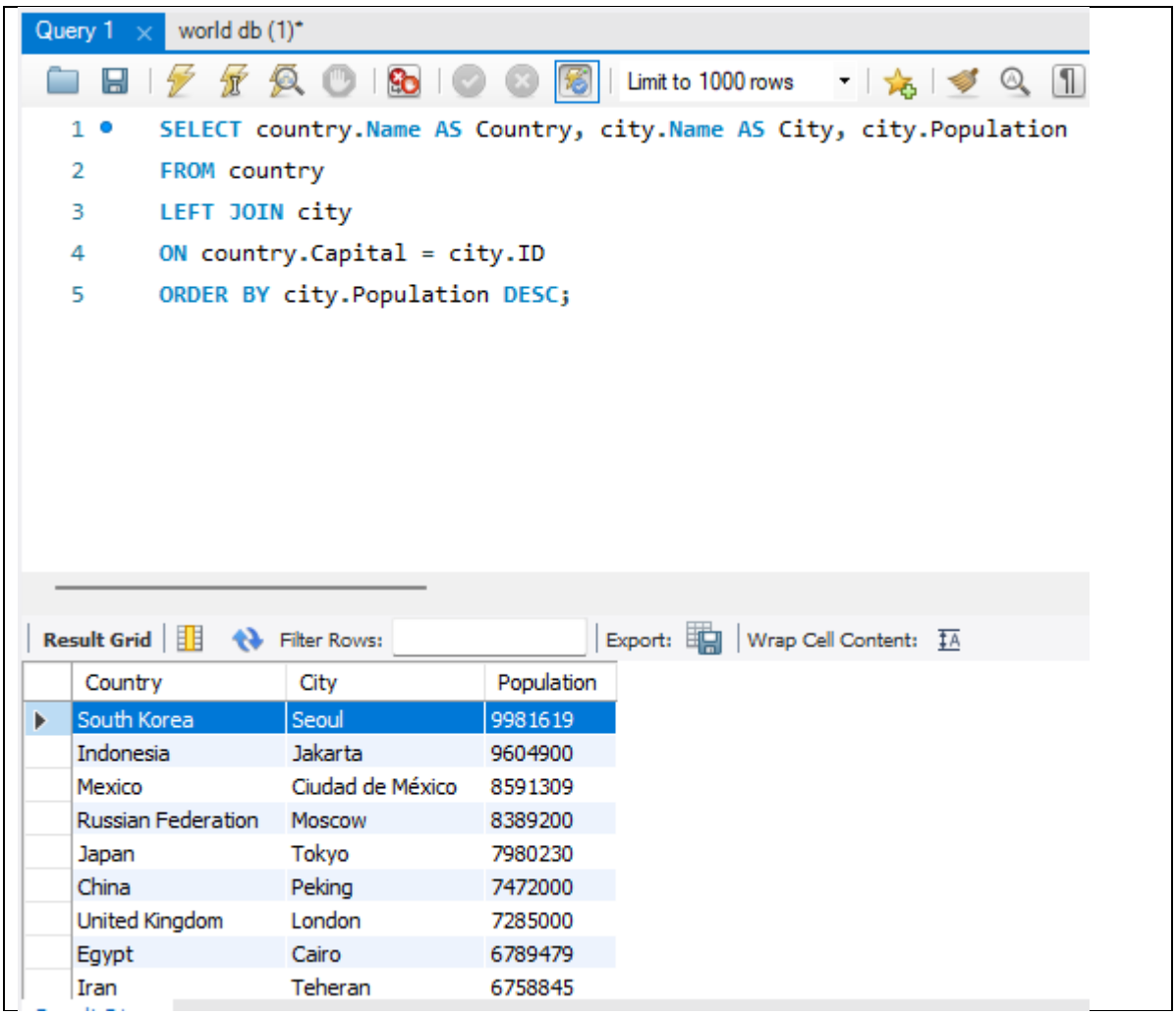
14. **Cities in Europe:** *Scenario:* A European cultural exchange program is seeking to connect students with cities across the continent. You're tasked with compiling a list of cities located in Europe from the database to facilitate program planning and student engagement.



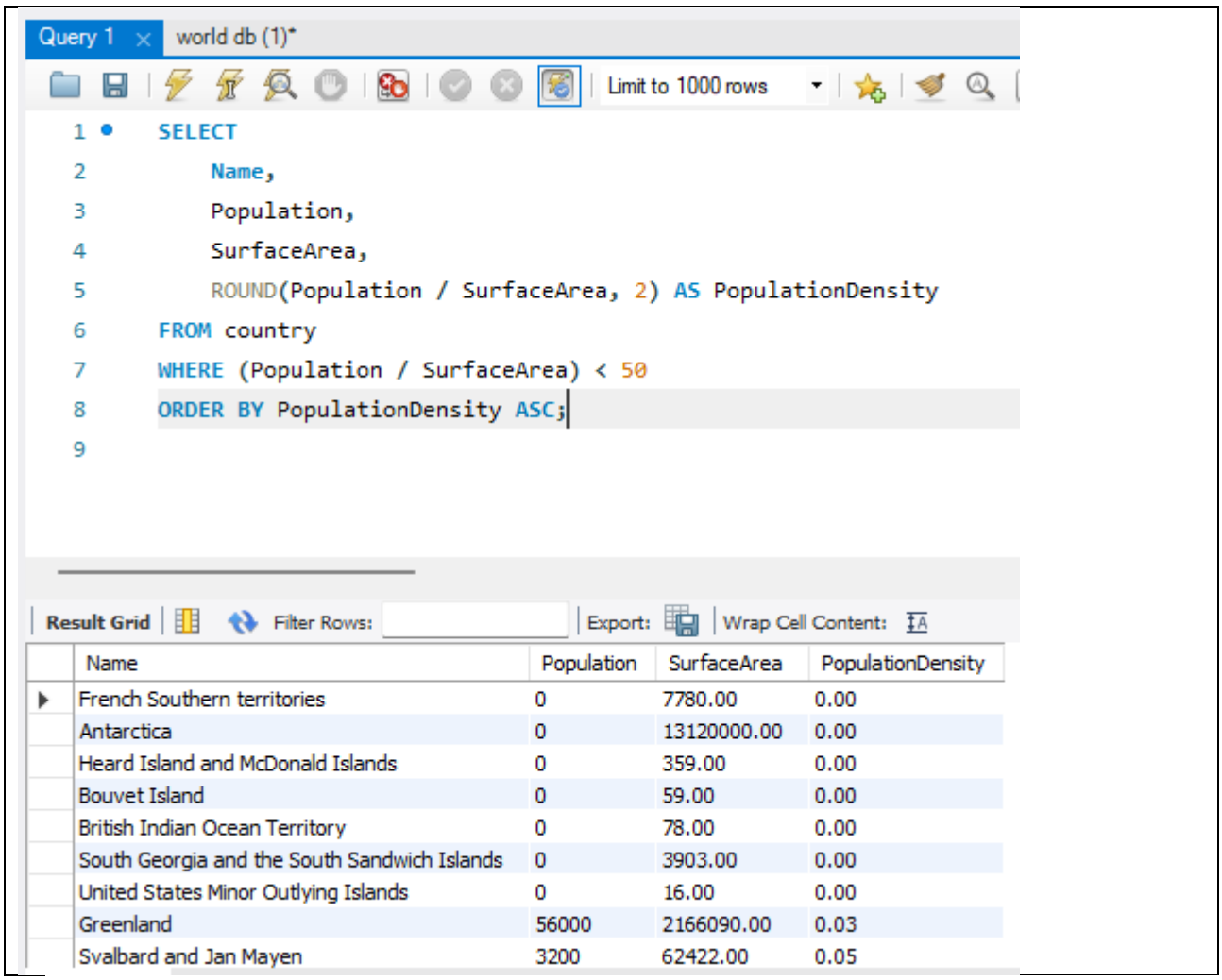
15. **Average Population by Country:** *Scenario:* A demographic research team is conducting a comparative analysis of population distributions across countries. You're tasked with calculating the average population for each country from the database to provide valuable insights into global population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



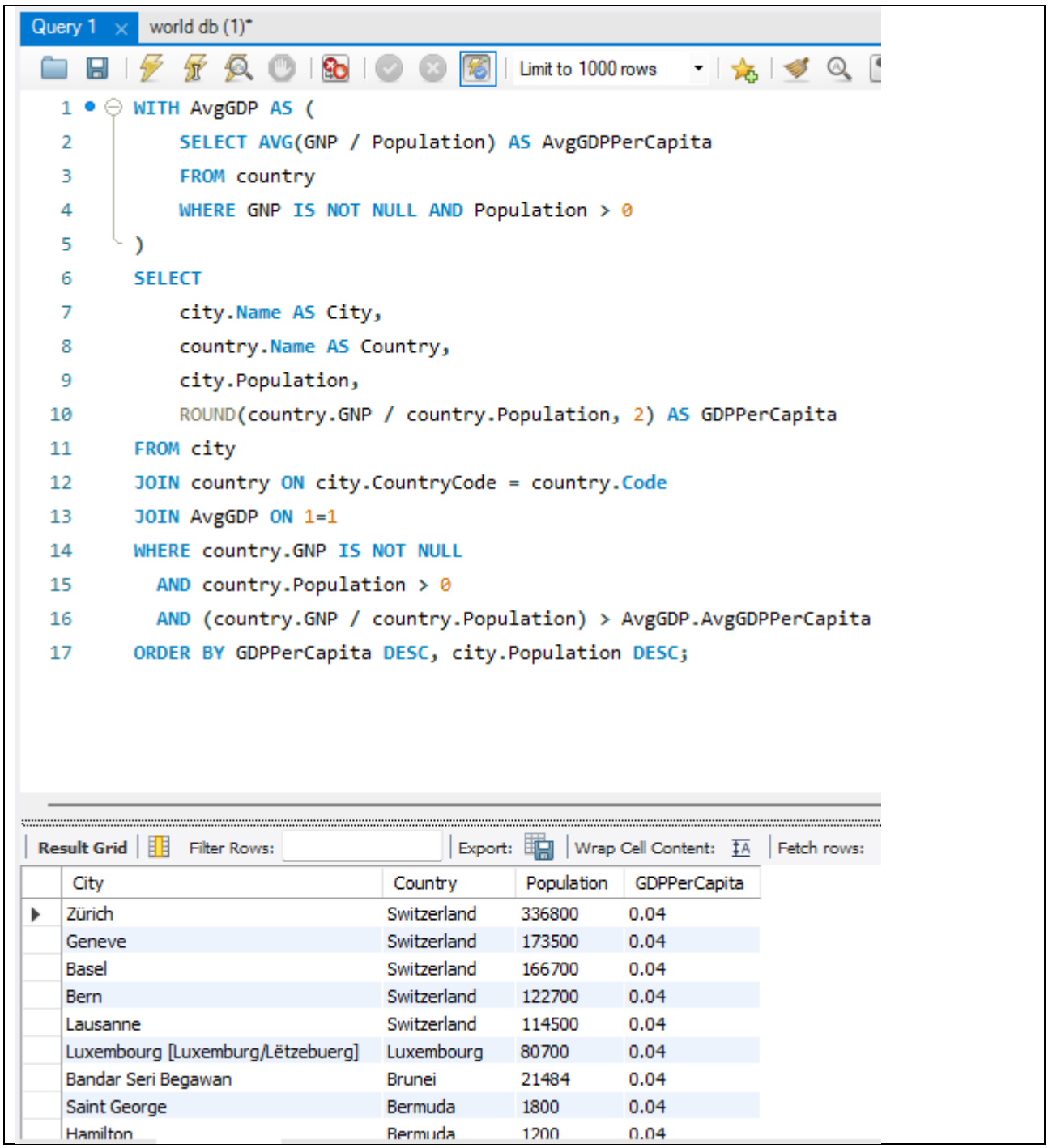
16. **Capital Cities Population Comparison:** *Scenario:* A statistical analysis firm is examining population distributions between capital cities worldwide. You're tasked with comparing the populations of capital cities from different countries to identify trends and patterns in urban demographics.



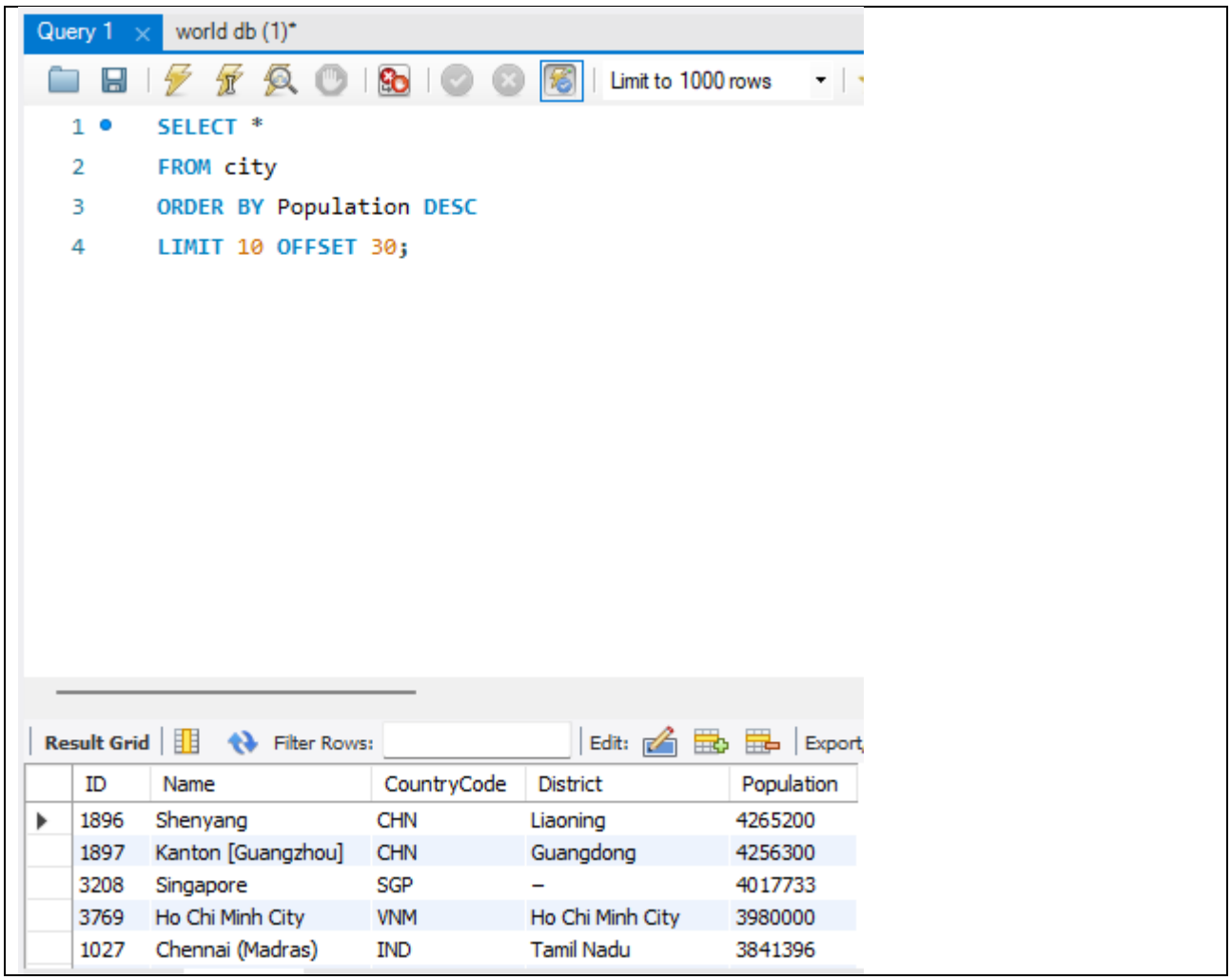
17. **Countries with Low Population Density:** *Scenario:* An agricultural research institute is studying countries with low population densities for potential agricultural development projects. You're tasked with identifying countries with sparse populations from the database to support the institute's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



18. **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.



19. **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.


Query 1
world db (1)*
Limit to 1000 rows
1 SELECT *
2 FROM city
3 ORDER BY Population DESC
4 LIMIT 10 OFFSET 30;
Result Grid
Filter Rows:
Edit:
Export
ID Name CountryCode District Population
1896 Shenyang CHN Liaoning 4265200
1897 Kanton [Guangzhou] CHN Guangdong 4256300
3208 Singapore SGP – 4017733
3769 Ho Chi Minh City VNM Ho Chi Minh City 3980000
1027 Chennai (Madras) IND Tamil Nadu 3841396

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.