

SQL Assignment

Portfolio



November 15, 2024

Suzan Yemane

Contents

[Task1 – MySQL Part 1: 1](#_Toc171687281)

[Task 2 – My SQL Part 2: 1](#_Toc171687282)

[Tasks 2](#_Toc171687283)

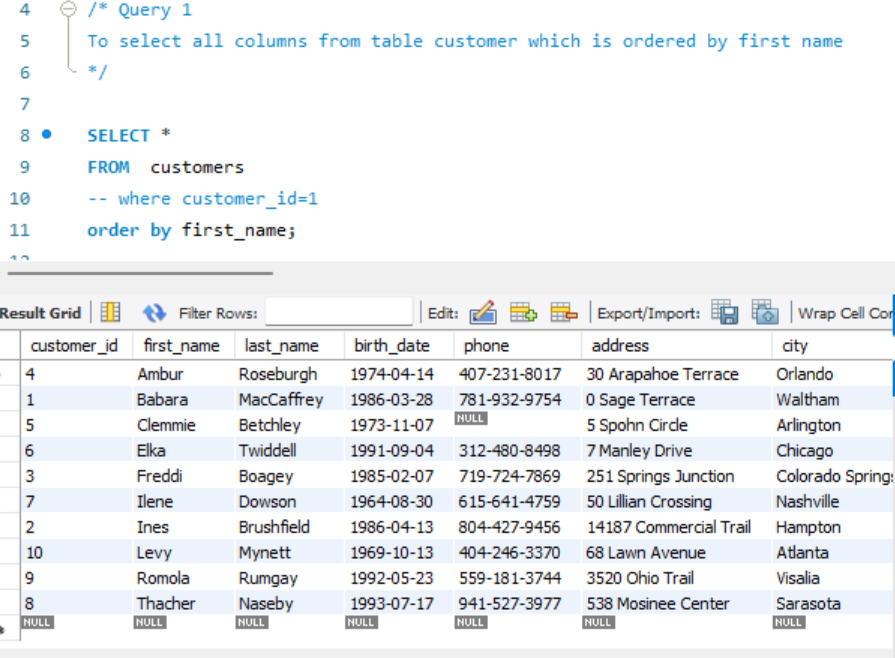
[Task 3 – Interview Part 1: 5](#_Toc171687284)

[Task 4 – Interview questions Part 2: 6](#_Toc171687285)

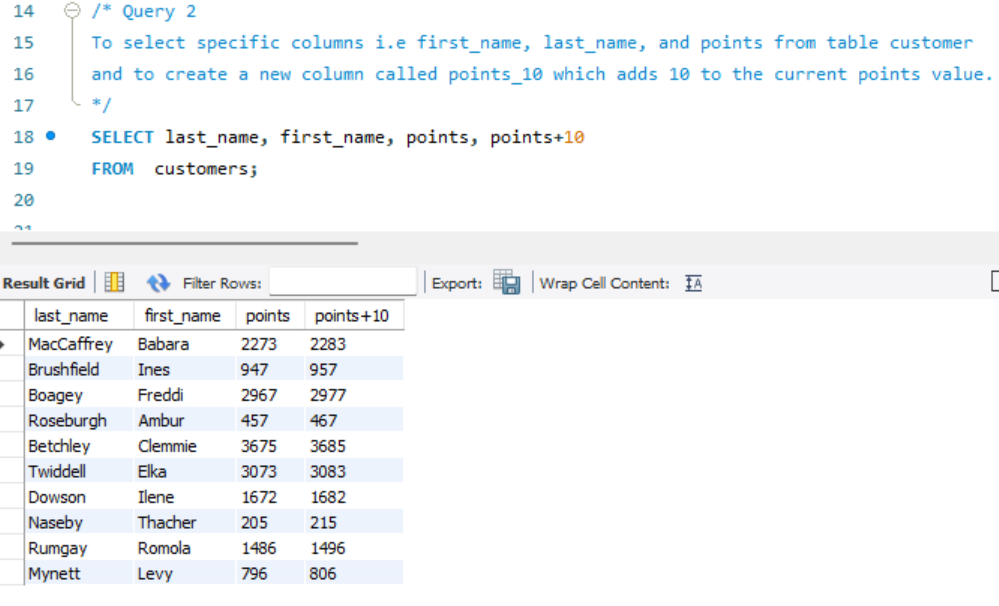
# Task1 – MySQL Part 1:

Task from PPT:

**Query 1**



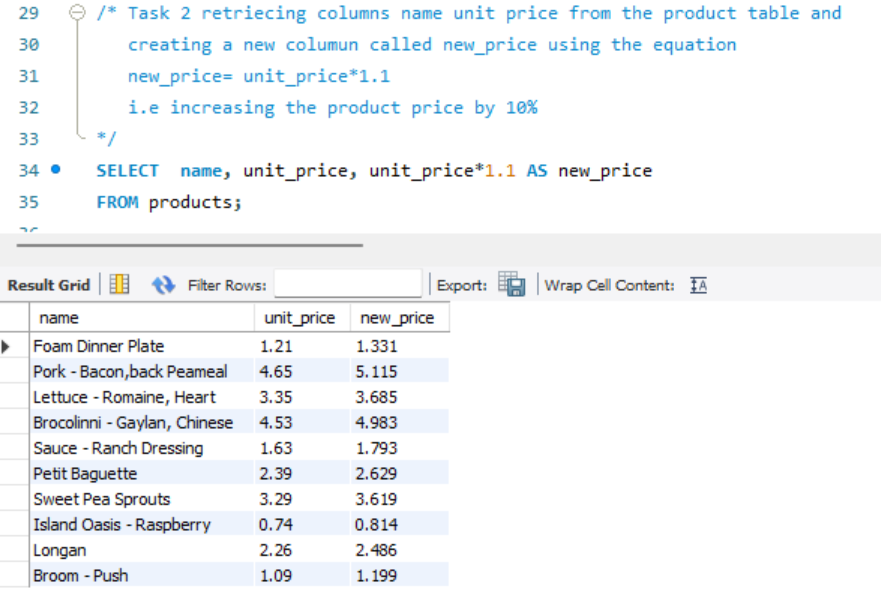
**Query 2**



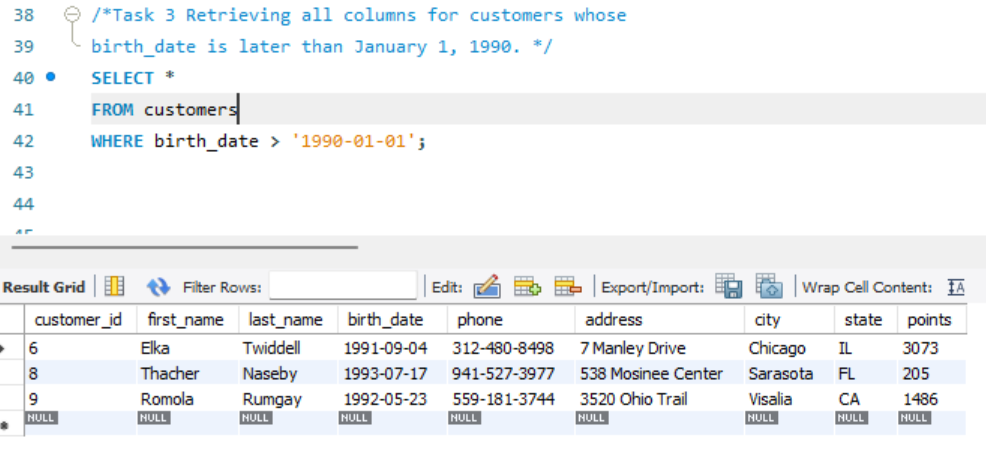
**TASK 1**

****

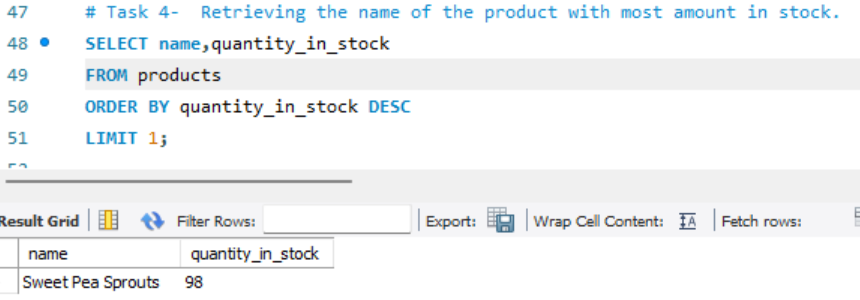
**TASK 2**

****

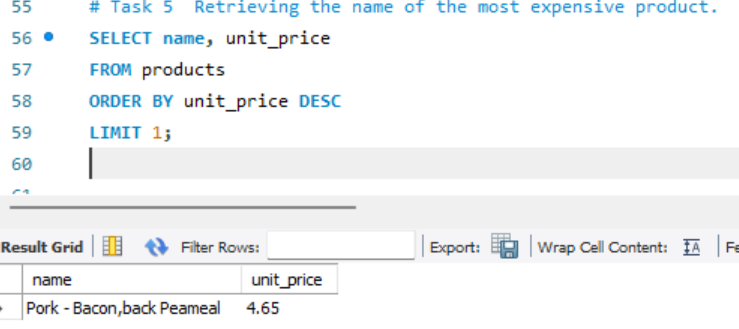
**TASK 3**

****

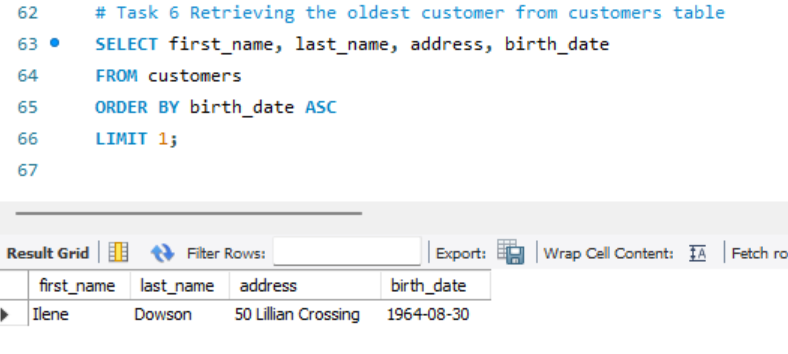
**TASK 4**

****

**TASK 5**

****

**TASK 6**

****

**EER Diagram for ‘sql\_store’ Database**

# 

|  |  |  |
| --- | --- | --- |
| **TABLE** | **Primary\_ Key/s** | **Foreign Key/s** |
| **Customers** | Customer\_id |  |
| **Orders** | Order\_id | Customer\_id, shippers\_id, status |
| **Order\_status** | Order\_status |  |
| **Order\_items** | Order\_id, Product\_id | Product\_id |
| **Shippers** | Shipper\_id |  |
| **Products** | Product\_id |  |

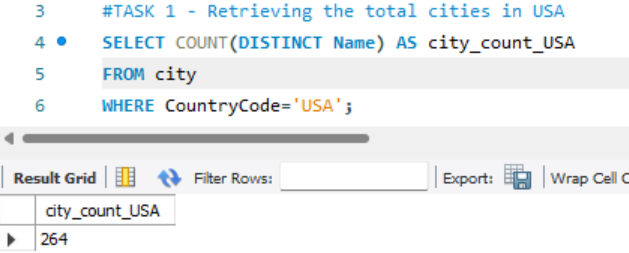
**Table Relationships:**

1. **Customers and Orders**: One customer can have multiple orders. This is a **1-to-many relationship** between the **Customers** table and the **Orders** table, with Customer\_id as the foreign key in the **Orders** table.
2. **Orders and Shippers**: Each order is associated with one shipping company. This is also a **1-to-many relationship**, with Shipper\_id as the foreign key in the **Orders** table.
3. **Orders and Order\_status**: Each order has one status (e.g., Pending, Shipped). This is a **1-to-many relationship** between **Orders** and **Order\_status**, with Order\_status\_id as the foreign key in the **Orders** table.
4. **Orders and Order\_items**: Each order can contain multiple items. This is a **1-to-many relationship** between **Orders** and **Order\_items**, with Order\_id as the foreign key in the **Order\_items** table.
5. **Order\_items and Products**: Each order item refers to one product. This is a **many-to-1 relationship** between **Order\_items** and **Products**, with Product\_id as the foreign key in the **Order\_items** table.

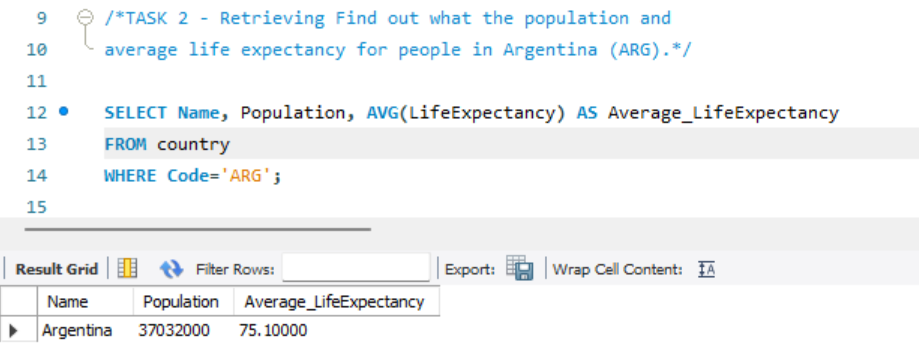
# Task 2 – My SQL Part 2:

Task from PPT:

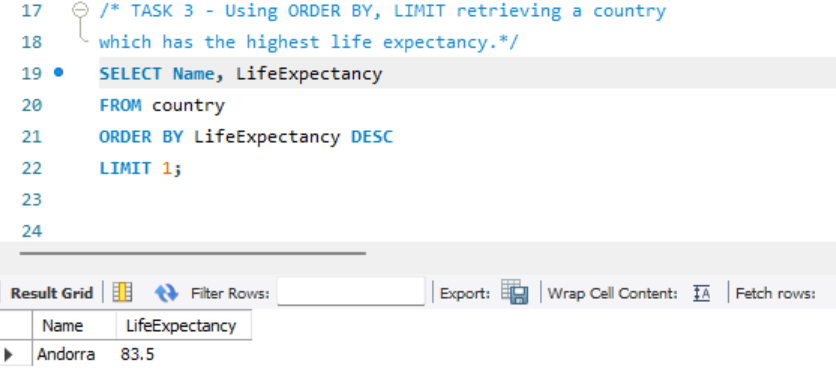
**TASK 1**



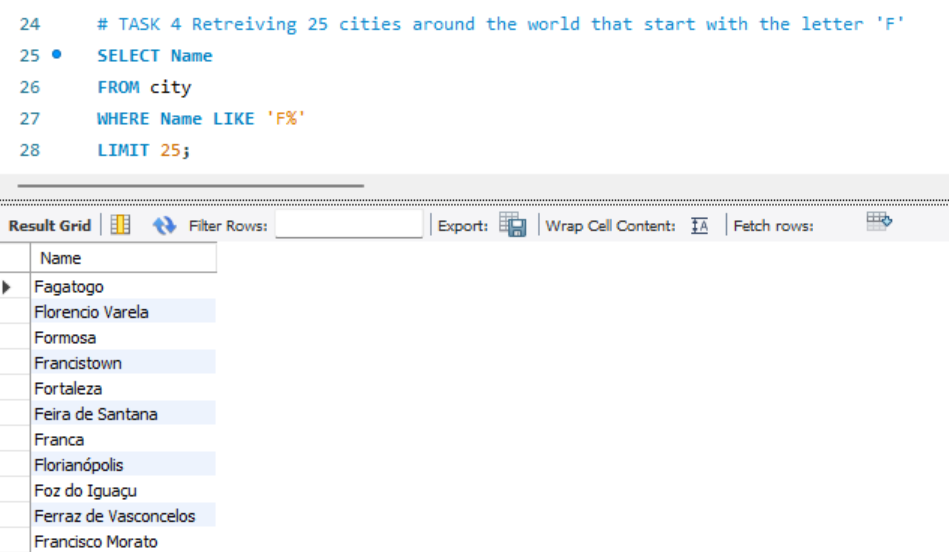
**TASK 2**

****

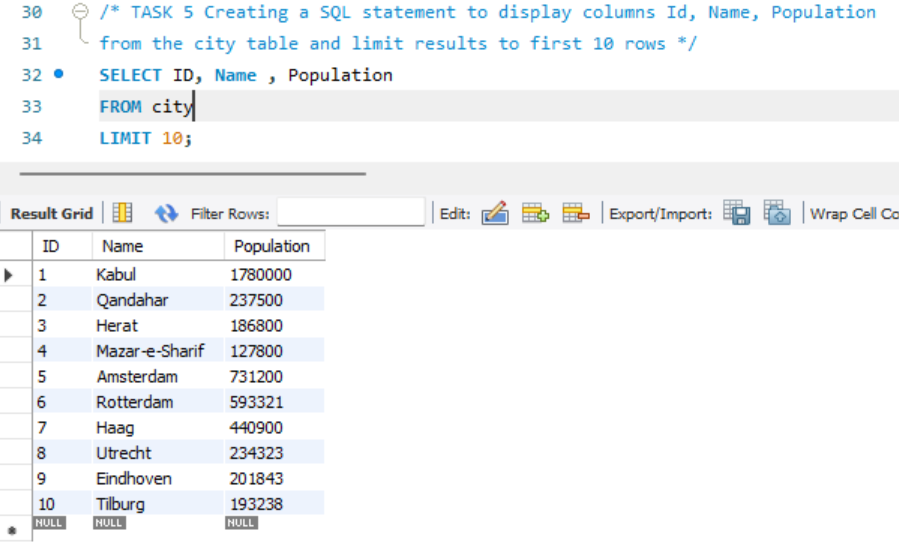
**TASK 3**

****

**TASK 4**

****

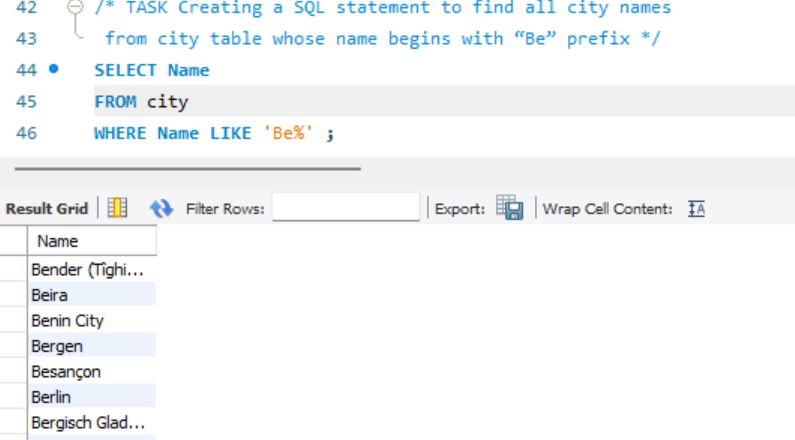
**TASK 5**

****

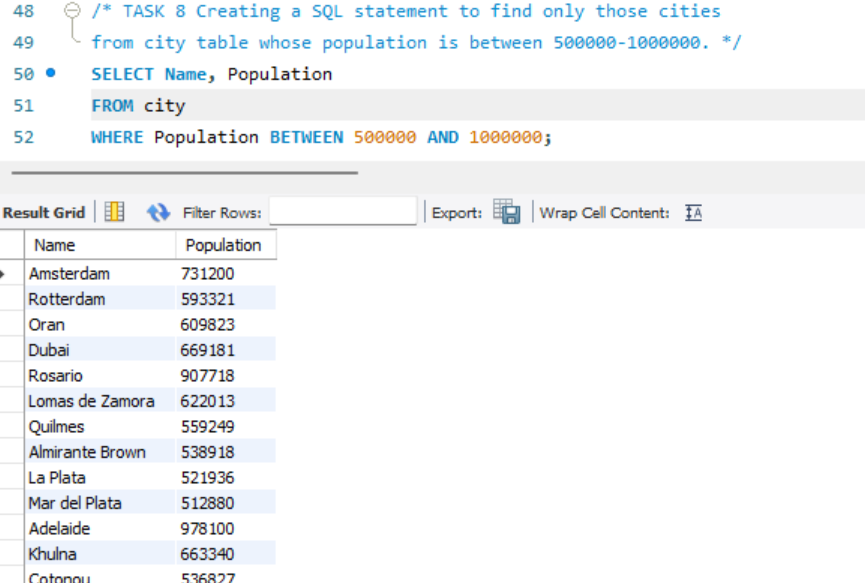
**TASK 6**

****

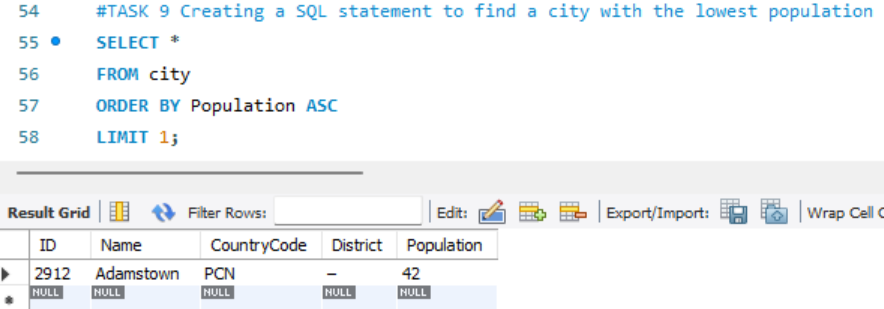
**TASK 7**

****

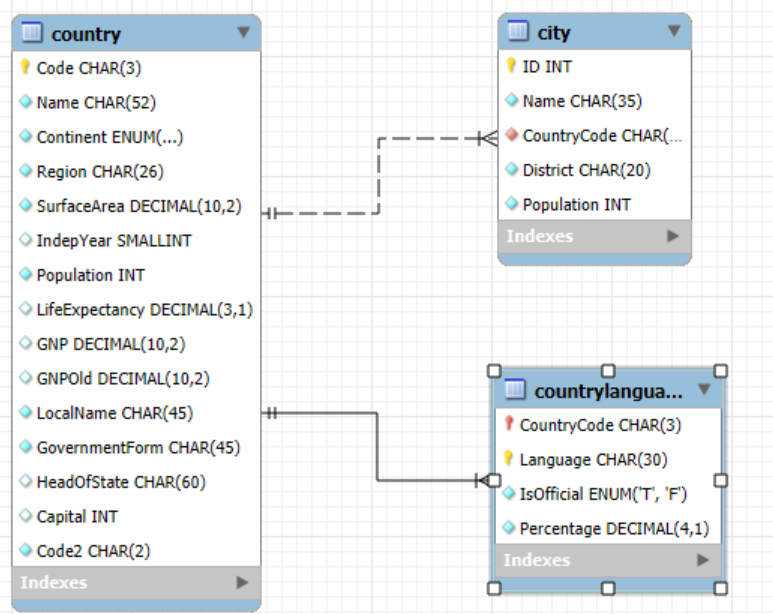
**TASK 8**

****

**TASK 9**

****

**EER Diagram for the database ‘world'**

****

* Identify the primary key in country table.

**Ans: Code CHAR(3)**

* Identify the primary key in city table.

**Ans: ID INT**

* Identify the primary key in countrylanguage table.

**Ans: CountryCode CHAR(3)**

**Language CHAR(30)**

* Identify the foreign key in city table.

**Ans: CountryCode CHAR(3)**

* Identify the foreign key in countrylanguage table.

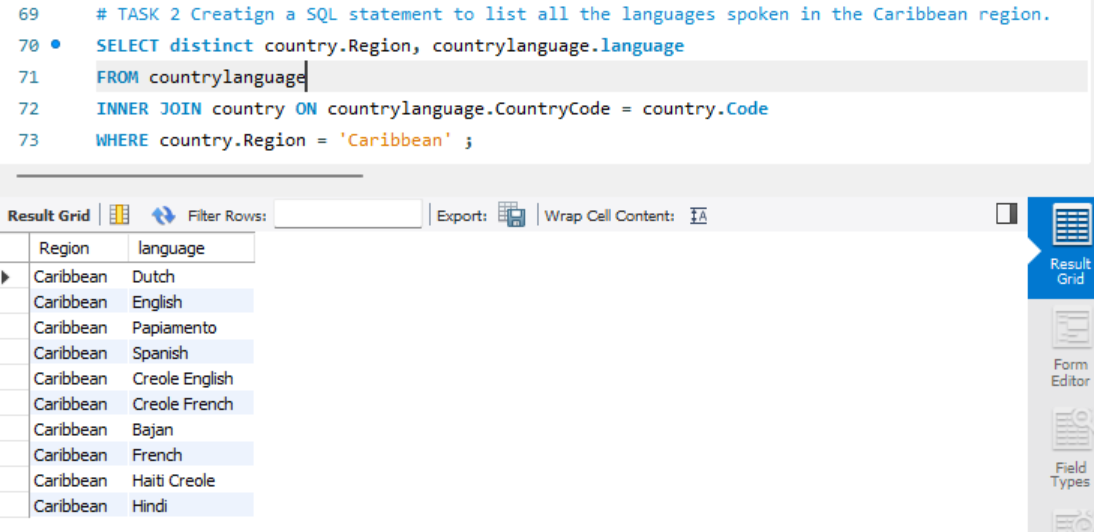
**Ans: CountryCode CHAR(3)**

**BONUS TASKS**

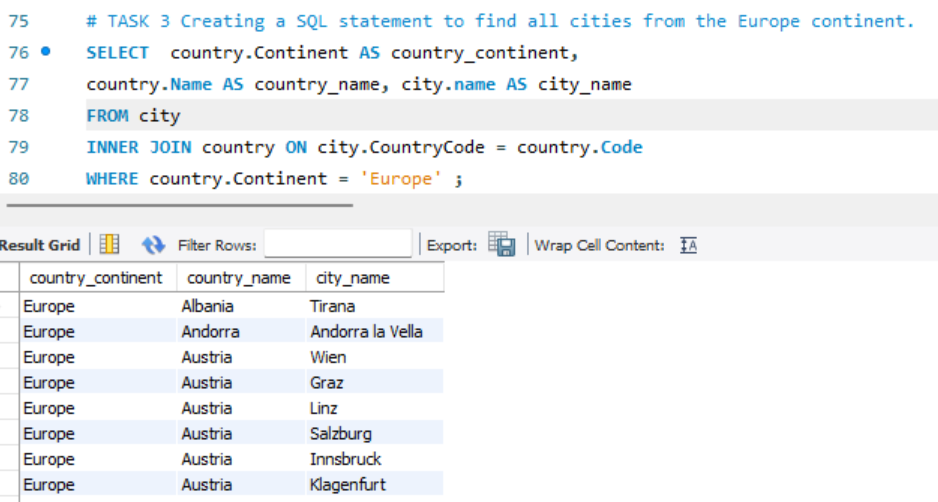
**Create a SQL statement to find the capital of Spain (ESP).**

****

**Create a SQL statement to list all the languages spoken in the Caribbean region.**

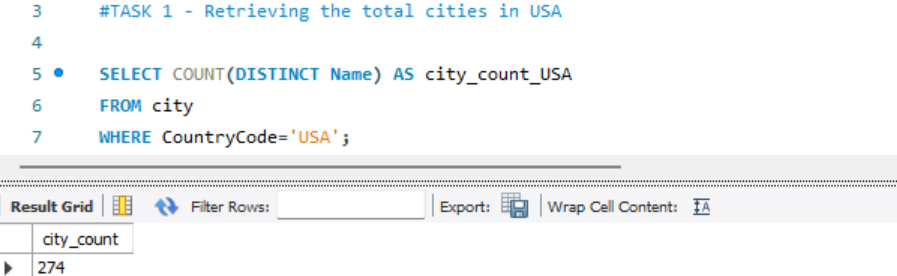
****

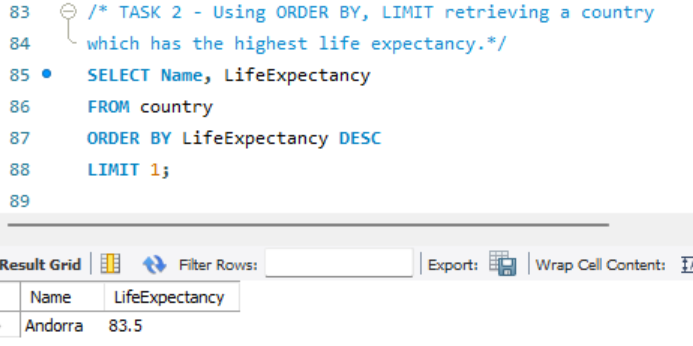
**Create a SQL statement to find all cities from the Europe continent.**

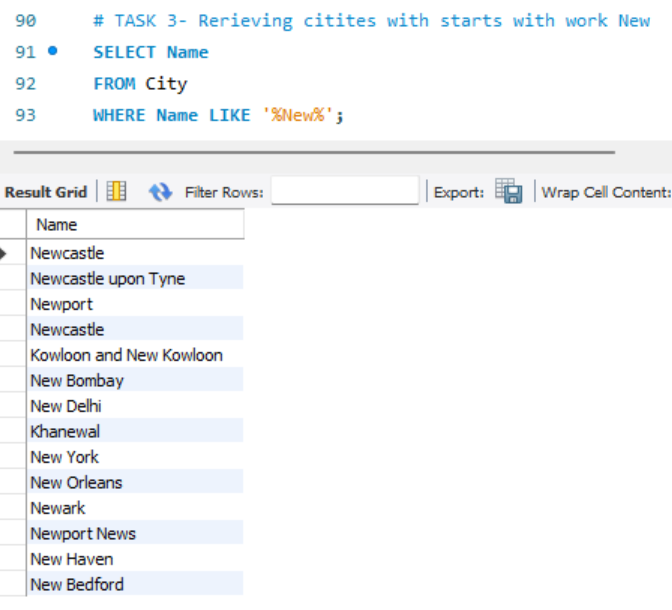


## Tasks

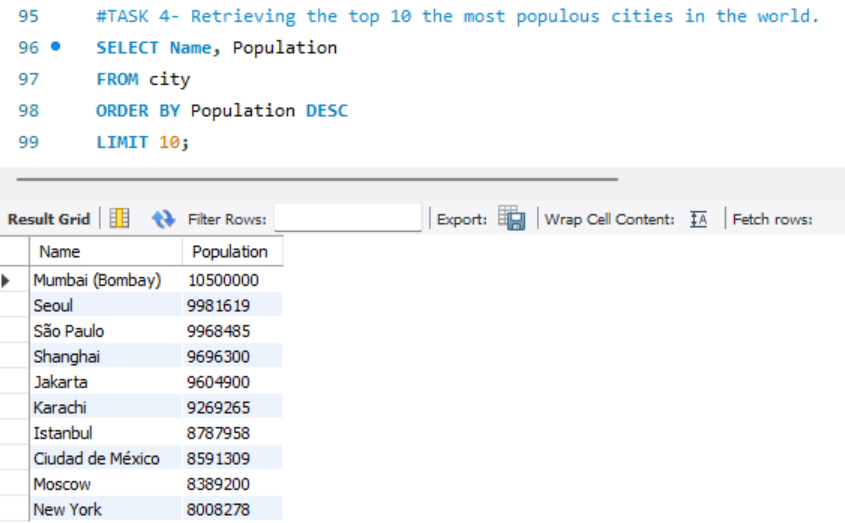
1. **Count Cities in USA:** *Scenario:* You've been tasked with conducting a demographic analysis of cities in the United States. Your first step is to determine the total number of cities within the country to provide a baseline for further analysis.



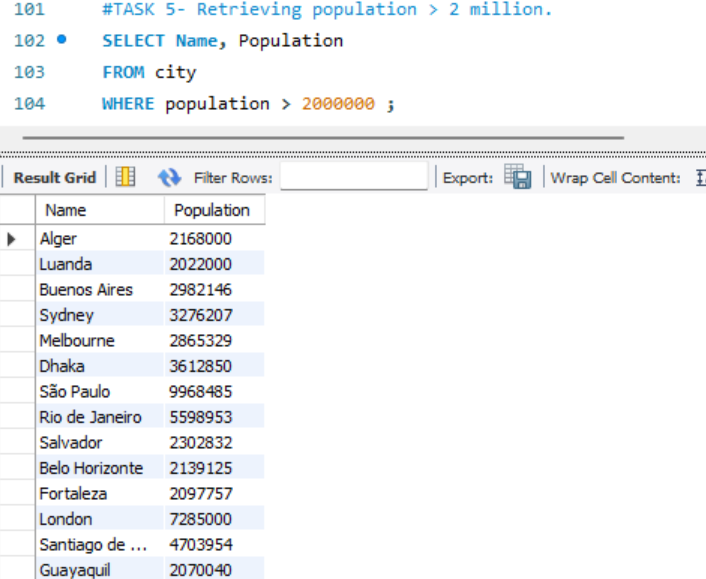
1. **Country with Highest Life Expectancy:** *Scenario:* As part of a global health initiative, you've been assigned to identify the country with the highest life expectancy. This information will be crucial for prioritizing healthcare resources and interventions.
2. **"New Year Promotion: Featuring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



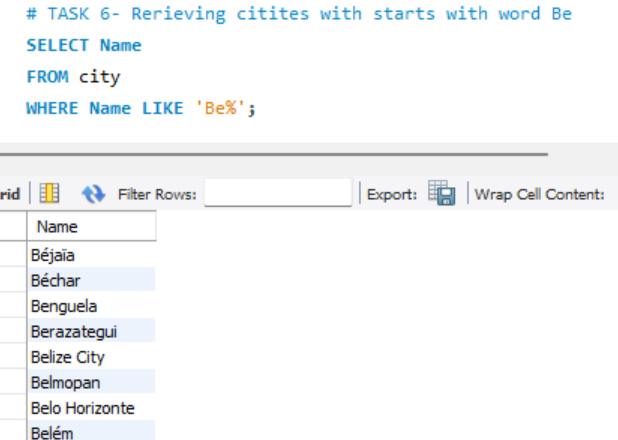
1. **Display Columns with Limit (First 10 Rows):** *Scenario:* You're tasked with providing a brief overview of the most populous cities in the world. To keep the report concise, you're instructed to list only the first 10 cities by population from the database.



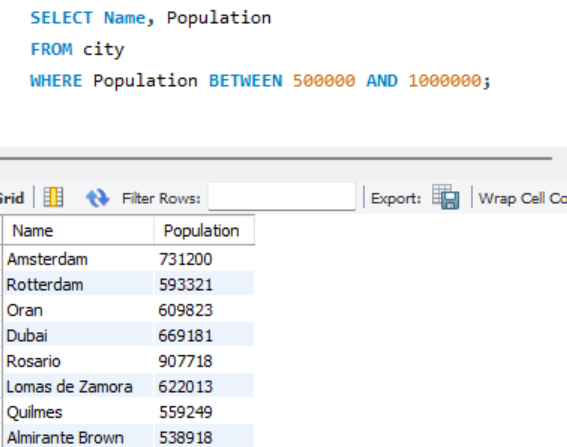
1. **Cities with Population Larger than 2,000,000:** *Scenario:* A real estate developer is interested in cities with substantial population sizes for potential investment opportunities. You're tasked with identifying cities from the database with populations exceeding 2 million to focus their research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



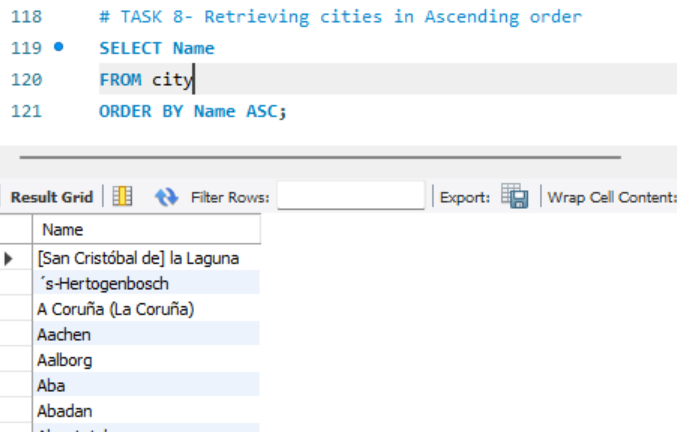
1. **Cities Beginning with 'Be' Prefix:** *Scenario:* A travel blogger is planning a series of articles featuring cities with unique names. You're tasked with compiling a list of cities from the database that start with the prefix 'Be' to assist in the blogger's content creation process.



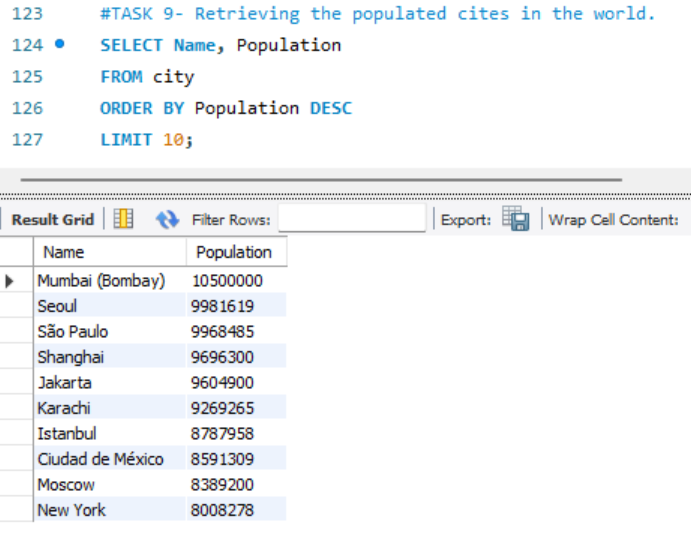
1. **Cities with Population Between 500,000-1,000,000:** *Scenario:* An urban planning committee needs to identify mid-sized cities suitable for infrastructure development projects. You're tasked with identifying cities with populations ranging between 500,000 and 1 million to inform their decision-making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



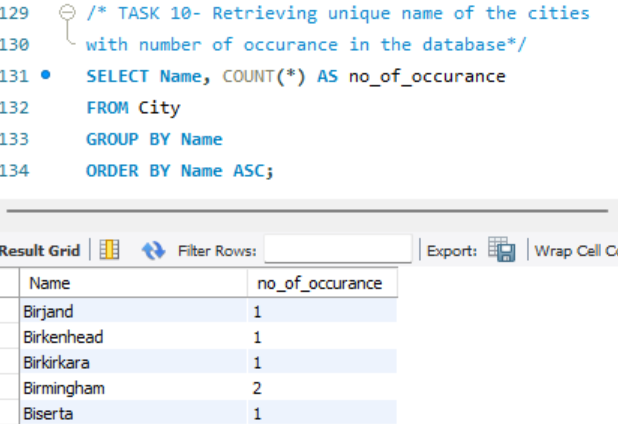
1. **Display Cities Sorted by Name in Ascending Order:** *Scenario:* A geography teacher is preparing a lesson on alphabetical order using city names. You're tasked with providing a sorted list of cities from the database in ascending order by name to support the lesson plan.



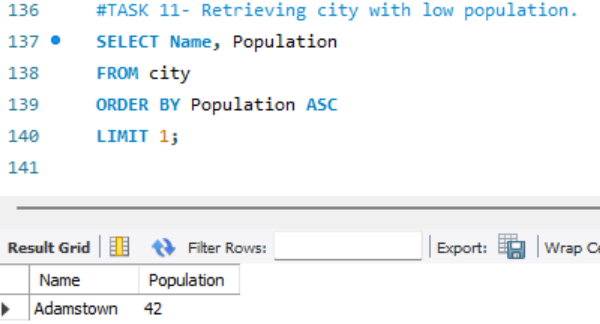
1. **Most Populated City:** *Scenario:* A real estate investment firm is interested in cities with significant population densities for potential development projects. You're tasked with identifying the most populated city from the database to guide their investment decisions and strategic planning.



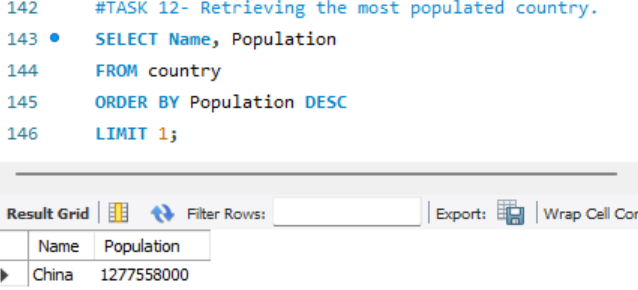
1. **City Name Frequency Analysis: Supporting Geography Education** *Scenario*: In a geography class, students are learning about the distribution of city names around the world. The teacher, in preparation for a lesson on city name frequencies, wants to provide students with a list of unique city names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



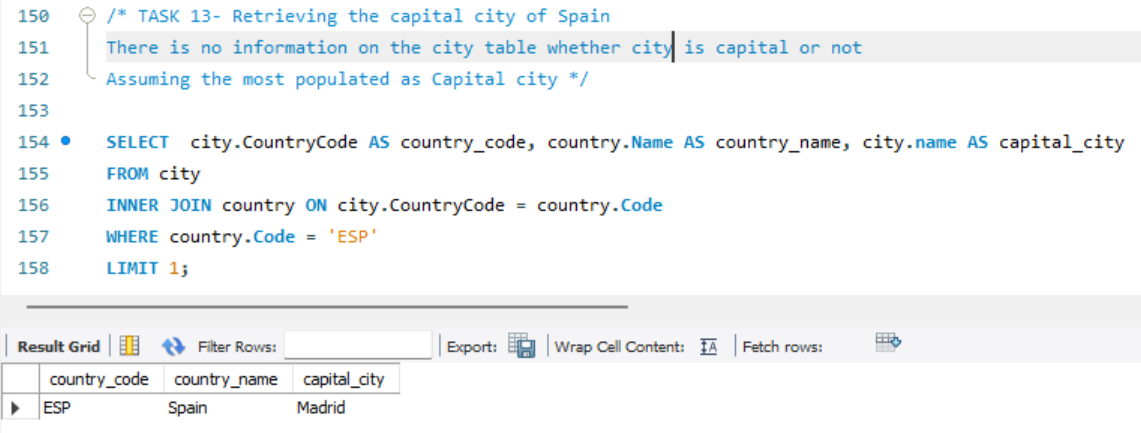
1. **City with the Lowest Population:** *Scenario:* A census bureau is conducting an analysis of urban population distribution. You're tasked with identifying the city with the lowest population from the database to provide a comprehensive overview of demographic trends.



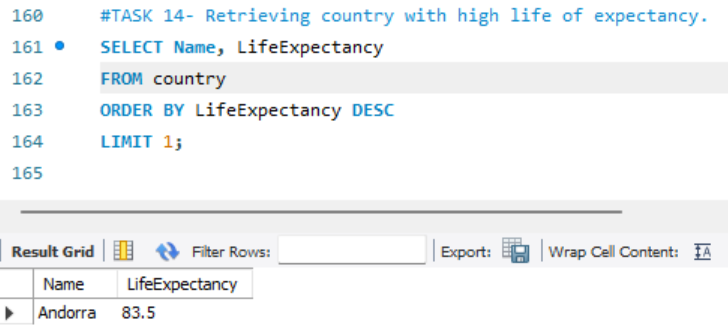
1. **Country with Largest Population:** *Scenario:* A global economic research institute requires data on countries with the largest populations for a comprehensive analysis. You're tasked with identifying the country with the highest population from the database to provide valuable insights into demographic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



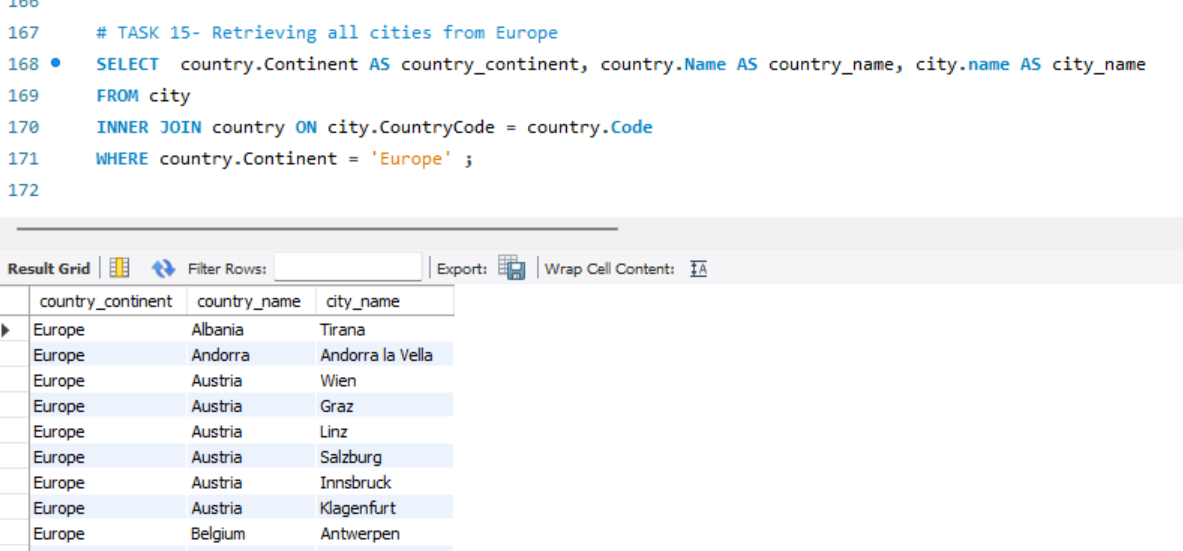
1. **Capital of Spain:** *Scenario:* A travel agency is organizing tours across Europe and needs accurate information on capital cities. You're tasked with identifying the capital of Spain from the database to ensure itinerary accuracy and provide travellers with essential destination information.



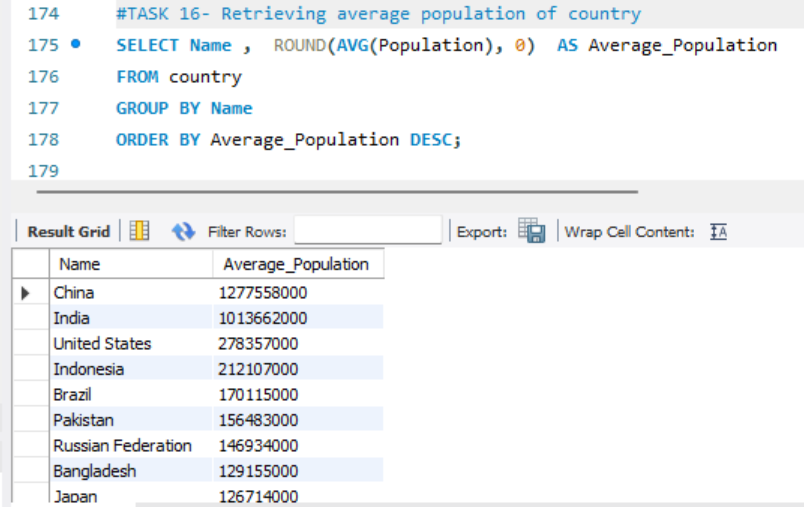
1. **Country with Highest Life Expectancy:** *Scenario:* A healthcare foundation is conducting research on global health indicators. You're tasked with identifying the country with the highest life expectancy from the database to inform their efforts in improving healthcare systems and policies.



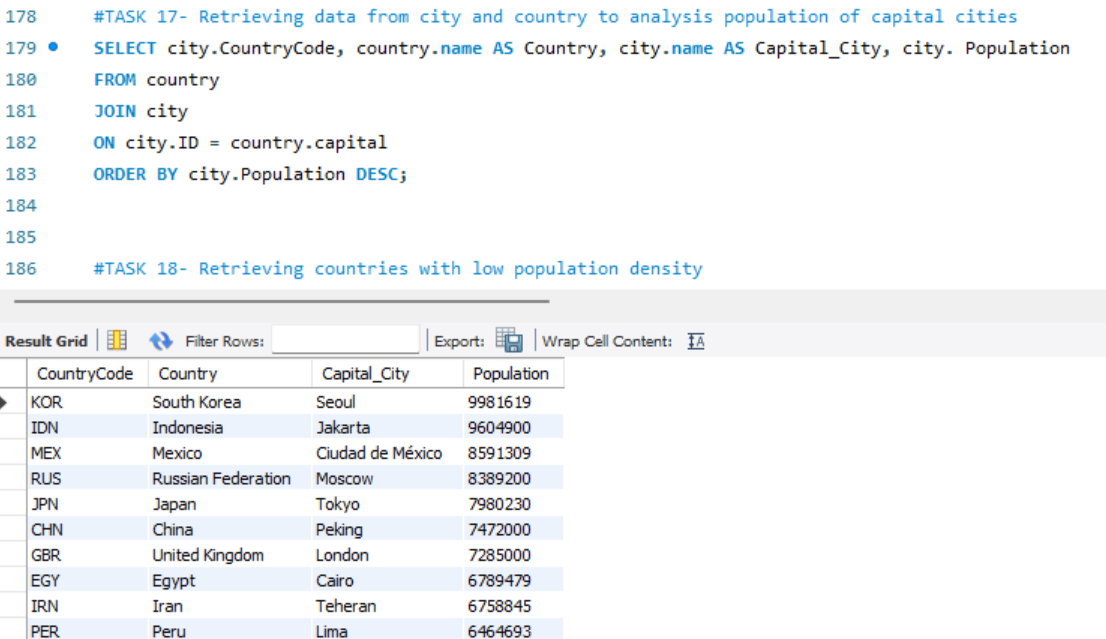
1. **Cities in Europe:** *Scenario:* A European cultural exchange program is seeking to connect students with cities across the continent. You're tasked with compiling a list of cities located in Europe from the database to facilitate program planning and student engagement.



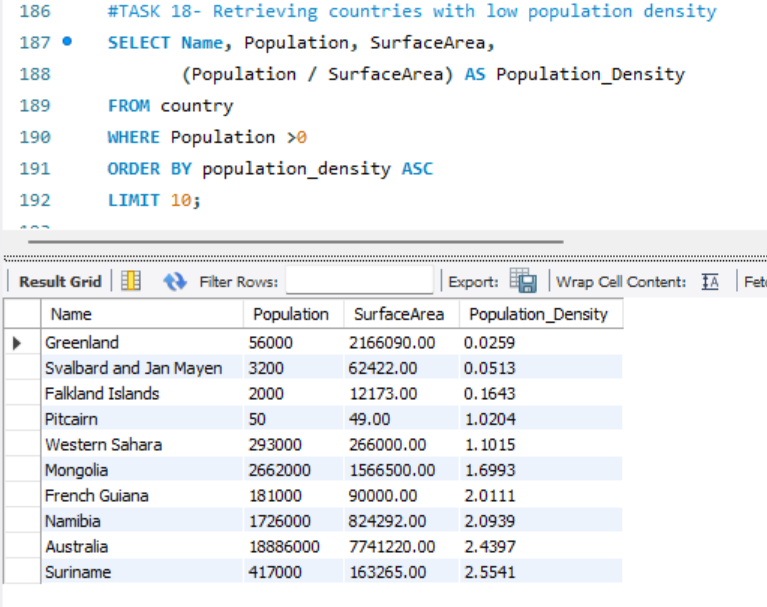
1. **Average Population by Country:** *Scenario:* A demographic research team is conducting a comparative analysis of population distributions across countries. You're tasked with calculating the average population for each country from the database to provide valuable insights into global population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



1. **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.

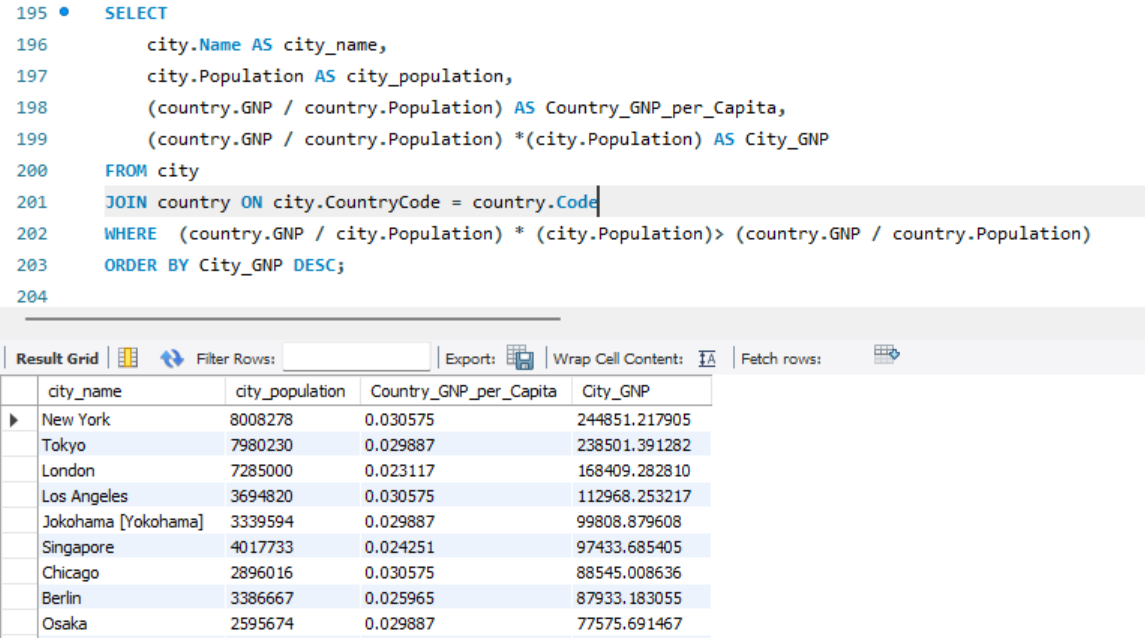


1. **Countries with Low Population Density:** *Scenario:* An agricultural research institute is studying countries with low population densities for potential agricultural development projects. You're tasked with identifying countries with sparse populations from the database to support the institute's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

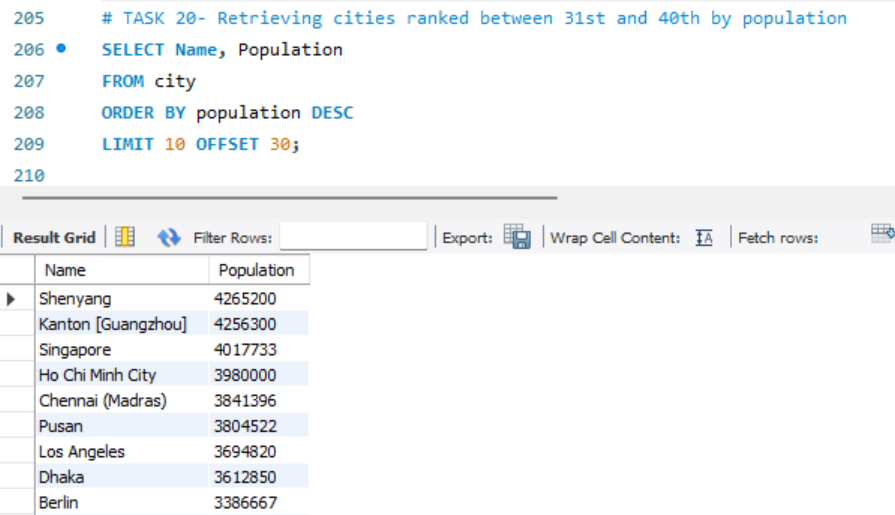


BONUS TASKS: Challenge yourself: These are optional tasks. Feel free to skip.

1. **Cities with High GDP per Capita:** *Scenario:* An economic consulting firm is analyzing 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.



1. **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.



# Task 3 – Interview Part 1:

**What is a Query?**​

A command used to retrieve data from a database. For instance, SQL (Structured Query Language) is commonly used for querying databases, allowing users to specify criteria for selecting, updating, or deleting data.

**What is the SELECT statement?​**

The SELECT statement is one of the most fundamental SQL commands used to retrieve specific data from a database. It allows users to specify columns, tables, and criteria to filter and display data according to particular needs.

**What is the WHERE clause?​**

The WHERE clause in SQL is used to filter records returned by a SELECT, UPDATE, or DELETE statement based on specific conditions. It allows users to specify criteria that records must meet to be included in the results. By using the WHERE clause, you can retrieve or modify only those records that match the specified condition(s).

**What is the Primary key?​**

A primary key is a unique identifier for each record in a database table. It ensures that each row in a table can be uniquely identified, preventing duplicate records and enabling efficient data retrieval.

**Key Characteristics of a Primary Key**

1. **Uniqueness**: Each value in a primary key column must be unique across all rows, ensuring that no two records share the same primary key value.
2. **Non-Null**: Primary keys cannot contain NULL values; every record must have a valid primary key value.
3. **Immutability**: Once set, primary key values should not change frequently, as they play a crucial role in data relationships and indexing.

**What is a Database?​**

A database is an organized collection of data that is stored and managed to enable efficient retrieval, modification, and analysis. Databases are essential for storing information in a structured way that allows users and applications to access, manage, and update data efficiently.

**Key Components of a Database**

**Tables**: A database is organized into tables, which are like spreadsheets with rows and columns. Each table represents a specific entity (e.g., Employees, Products) and contains records (rows) and fields (columns).

**Records (Rows)**: Each row in a table is a record that represents a single instance of the entity. For example, a row in an Employees table might represent one employee.

**Fields (Columns**): Each column in a table is a field that represents a specific attribute of the entity. For example, in an Employees table, columns might include employee\_id, name, and department.

**Primary and Foreign Keys**: Primary keys uniquely identify each record in a table, while foreign keys link records between tables, enabling relationships within a database.

# 

# Task 4 – Interview questions Part 2:

**List the different types of relationships in SQL and give examples.​**

In SQL and relational databases, relationships define how data in one table relates to data in another. The main types of relationships in SQL are:

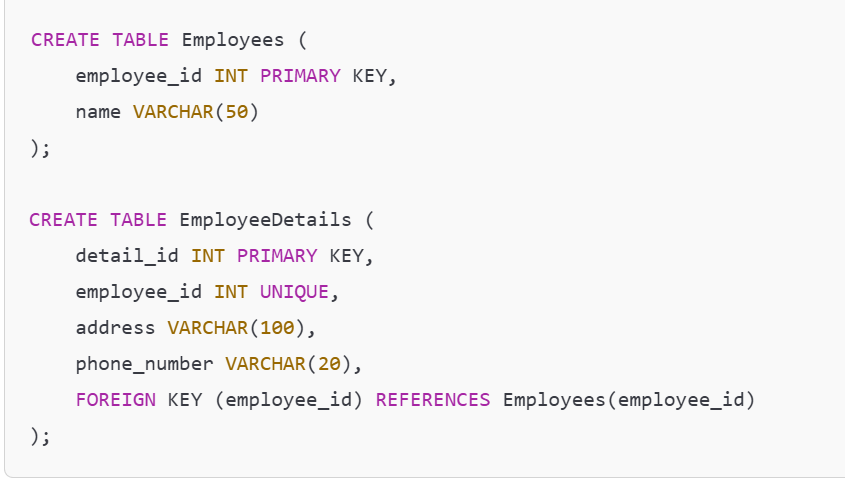
1. **One-to-One (1:1) Relationship**
2. **One-to-Many (1) Relationship**
3. **Many-to-Many (M) Relationship**

**1. One-to-One (1:1) Relationship**

In a one-to-one relationship, each record in one table is associated with only one record in another table, and vice versa. This relationship is often used when splitting data into multiple tables for organization or security.

**Example:**

Consider two tables: Employees and EmployeeDetails. Each employee has one corresponding detail record.



In this example:

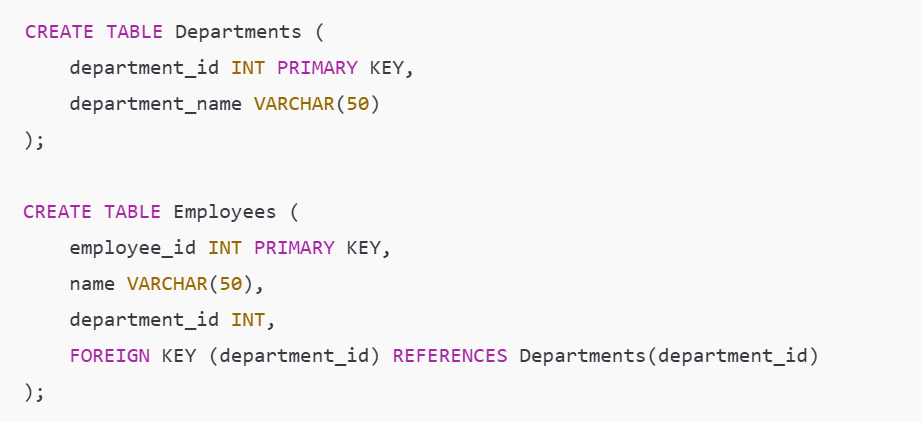
* Each employee\_id in Employees is unique and matches one employee\_id in EmployeeDetails, creating a one-to-one relationship.

**2. One-to-Many (1) Relationship**

In a one-to-many relationship, each record in one table can be associated with multiple records in another table. This is the most common type of relationship in relational databases.

**Example:**

Consider two tables: Departments and Employees. A department can have multiple employees, but each employee belongs to only one department.



In this example:

* Each department in the Departments table can have multiple employees in the Employees table, creating a one-to-many relationship.

**3. Many-to-Many (M**

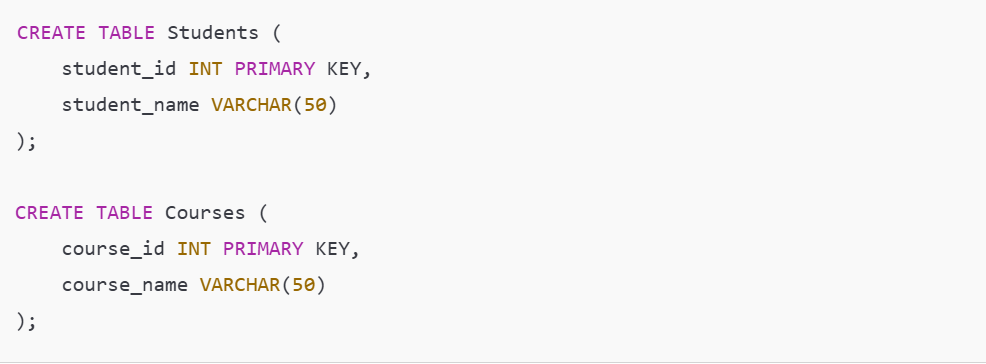
**) Relationship**

In a many-to-many relationship, records in one table can be associated with multiple records in another table, and vice versa. SQL does not support direct many-to-many relationships, so an intermediary table (also called a junction or associative table) is used to handle this.

**Example:**

Consider two tables: Students and Courses. Each student can enroll in multiple courses, and each course can have multiple students.

1. **Create Tables**:



1. **Create Junction Table**:



In this example:

* The Enrollments table establishes the many-to-many relationship between Students and Courses.
* Each row in Enrollments represents a unique combination of student\_id and course\_id, allowing each student to be enrolled in multiple courses and each course to have multiple students.

**Summary**

* **One-to-One**: Each record in Table A relates to one record in Table B.
* **One-to-Many**: Each record in Table A can relate to multiple records in Table B.
* **Many-to-Many**: Each record in Table A can relate to multiple records in Table B, and vice versa, usually managed through a junction table.

These relationships are key to organizing data in relational databases, enabling efficient querying and ensuring data integrity.

**What is Normalization?​**

**Normalization** is a process in database design used to organize data to reduce redundancy (duplicate data) and improve data integrity. By dividing a database into smaller tables and establishing relationships among them, normalization aims to ensure that each piece of data is stored in only one place, minimizing the risk of inconsistencies.

**Key Objectives of Normalization**

* **Eliminate Data Redundancy:** Avoids storing the same data in multiple tables, which reduces storage space and simplifies updates.
* **Ensure Data Integrity and Consistency:** Ensures that each piece of data is accurate and up-to-date across the database.
* **Optimize Database Structure:** Helps in organizing tables to avoid anomalies in insertions, deletions, and updates.

**Normal Forms**

**Normalization** is typically achieved by applying a series of rules known as normal forms. Each form represents a level of normalization, with each level building on the previous one. Here’s a breakdown of the most common normal forms:

**1. First Normal Form (1NF)**

Ensures that all columns contain atomic (indivisible) values. This means each cell in a table should hold a single value, not a list or set.

Requires each column to contain values of a single type.

Example: If a column in an Orders table contains a list of products, splitting each product into a separate row brings it to 1NF.

**2. Second Normal Form (2NF)**

Achieved when the table is already in 1NF and all non-key columns are fully functionally dependent on the primary key.

Eliminates partial dependency, meaning each non-key column should depend on the entire primary key, not just part of it (relevant mainly for tables with composite primary keys).

Example: In a table with order\_id and product\_id as a composite primary key, any information about order\_id alone (such as customer information) should be moved to a different table to satisfy 2NF.

**3. Third Normal Form (3NF)**

Achieved when the table is in 2NF and all non-key columns are directly dependent only on the primary key.

Removes transitive dependency, meaning no non-key column should depend on another non-key column.

**Example:** If a table has employee\_id (primary key), department\_id, and department\_name, then department\_name should be moved to a separate Departments table since it depends on department\_id, not directly on employee\_id.

**Higher Normal Forms**

While 1NF, 2NF, and 3NF are the most commonly used, additional forms like Boyce-Codd Normal Form (BCNF), 4NF, and 5NF exist for further refinement but are less commonly applied in everyday database design.

Example of Normalization

Imagine an unnormalized Students table with duplicate data:

| **student\_id** | **student\_name** | **course\_name** | **instructor\_name** |
| --- | --- | --- | --- |
| 1 | Alice | Math 101 | Prof. Johnson |
| 2 | Bob | Science 101 | Dr. Smith |
| 1 | Alice | Science 101 | Dr. Smith |

After normalization:

1. **Students Table**:

| **student\_id** | **student\_name** |
| --- | --- |
| 1 | Alice |
| 2 | Bob |

1. **Courses Table**:

| **course\_id** | **course\_name** | **instructor\_name** |
| --- | --- | --- |
| 101 | Math 101 | Prof. Johnson |
| 102 | Science 101 | Dr. Smith |

1. **Enrollments Table**:

| **student\_id** | **course\_id** |
| --- | --- |
| 1 | 101 |
| 1 | 102 |
| 2 | 102 |

**Query Question**​

**Modify query to show the population of Germany.​**

​SELECT population FROM world​

WHERE name = 'France'​

Answer:

SELECT population FROM world​

WHERE name = 'Germany'​;

**Select the query which gives the name of countries beginning with U.​**

**Answer:**

SELECT name

FROM world

WHERE name LIKE 'U%' ​;

**Select the answer which shows the problem with this SQL code - the intended result should be the continent of France:​**

SELECT continent FROM world WHERE 'name' = 'France'​

**Answer:** 'name' should be name​

**Select the code which shows the countries that end in A or L.​**

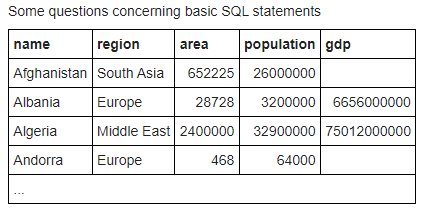
**Answer:**

SELECT name

FROM world

WHERE name LIKE '%A' OR name LIKE '%L';

**Given the table on the left, select the query which produces this table below.**

**** ****

**Answer:**

SELECT name, population

FROM world

WHERE population BETWEEN 1000000 AND 1250000 ;