



Data Technician

Name: Rebecca Osewa

Course Date: 18/12/2025

Table of contents

Day 1: Task 1.....	3
Day 1: Task 2.....	5
Day 2: Task 1.....	6
Day 3: Task 1.....	9
Day 3: Task 2.....	12
Day 4: Task 1: SQL Practical.....	16
Course Notes.....	29
Additional Resources.....	29
Additional Information.....	29



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 combination of columns, in a database table that uniquely identifies each record in that table. It ensures that no two rows have the same value in the primary key field, maintaining the integrity of the data. A primary key must always contain a unique value and cannot be left empty, as every record must be identifiable. For example, in a student's table, a Student ID column could serve as the primary key, as each student would have a unique ID that distinguishes them from all others. In essence, the primary key acts as the main identifier for each record in a table, helping to organise and retrieve data efficiently.
How does this differ from a secondary key?	A secondary key is a column or set of columns used to access or organise data in a table but does not uniquely identify each record. Unlike a primary key, a secondary key can have duplicate values and may contain nulls. It is primarily used for searching, sorting, or creating indexes to improve query performance. Essentially, while a primary key ensures uniqueness, a secondary key is used to make data retrieval more efficient.
How are primary and foreign keys related?	A foreign key is a column, or set of columns, in one table that refers to the primary key in another table. This relationship links the two tables and ensures integrity, meaning that values in the foreign key column must match existing values in the primary key column of the referenced table. The primary key uniquely identifies a record in its own table, while the foreign key allows other tables to reference that record. This connection is fundamental for establishing relationships between tables in a relational database.
Provide a real-world example of a one-to-one relationship	A real-world example of a one-to-one relationship is the relationship between a person and their passport. Each person can have only one passport, and each passport is assigned to only one person. In a database, this could be represented with a person table and a passport table, where the Person ID serves as the primary key in the person table and as a foreign key in the passport table. This ensures that each passport record is linked to exactly one person, and no person has more than one passport in the system.
Provide a real-world example of a one-to-many relationship	A real-world example of a one-to-many relationship is the relationship between a teacher and their students. One teacher can teach many students, but each student is assigned to only one teacher in that context. In a database, this could be represented with a teacher table and a student table, teacher ID is the primary key in the teacher table and serves as a foreign key in the student table. This setup allows multiple student records to be associated with a single teacher while maintaining the link to the correct teacher.



Provide a real-world example of a many-to-many relationship

A real-world example of a many-to-many relationship is the relationship between students and courses at a school or university. A student can enrol in multiple courses, and each course can have many students. In a database, this is typically handled with a junction table (or linking table) called something like student courses, which contains foreign keys referencing both the students table and the courses table. This allows the database to track which students are enrolled in which courses without duplicating data in either table.



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?	<p>A relational database organises data into tables, also called relations, which consist of rows and columns. Each row represents a record, and each column represents a field of data. Relationships between tables are defined using keys, allowing for complex queries and data integrity. Relational databases use Structured Query Language (SQL) for managing and querying data. They are best suited for structured data where consistency and relationships are important.</p> <p>A non-relational database, also known as a NoSQL database, stores data in formats other than tables, such as documents, key-value pairs, graphs, or wide-column stores. These databases are designed for flexibility, scalability, and handling large volumes of unstructured or semi-structured data. They do not rely on SQL, and relationships between data are often managed differently, if at all. Non-relational databases are ideal for applications that require rapid development, horizontal scaling, and handling diverse data types</p>
What type of data would benefit off the non-relational model? Why?	<p>A non-relational (NoSQL) model is typically unstructured</p> <p>Examples include:</p> <p>Social media content: Posts, comments, likes, and multimedia that vary in format.</p> <p>IoT sensor data: Continuous streams of readings that may not have a fixed structure.</p> <p>E-commerce product catalogues: Items with varying attributes, such as colour, size, specifications.</p> <p>User-generated content: Reviews, messages, or logs that differ in structure between users.</p> <p>Big data analytics: Large datasets that require horizontal scaling and fast write/read operations.</p> <p>Non-relational databases are flexible and do not require a fixed schema, which makes them ideal for data that changes frequently or has inconsistent formats. They can scale horizontally, handling very large volumes of data efficiently. Additionally, they allow for fast read</p>



and write operations, which is essential for real-time applications like social media feeds, analytics dashboards, and IoT systems.

Day 2: Task 1

You are a junior data analyst at **Northwind Traders**. Your manager has assigned you a series of real-world business tasks. For each scenario:

1. **Write an SQL query** to solve the request. Write an SQL query to solve the request.
 2. Run the query in MySQL Workbench using the Northwind database.
 3. Paste your query in the box below each question as evidence of completion.
-

1. Retrieve Full Customer Data

Your manager has asked you to export the full list of all customer details into a report.

 **Write a SQL query to retrieve all columns from the Customers table.**

Select*

From Customers;

2. Customer Names and Cities for Marketing

A marketing analyst is targeting a campaign based on customer names and their cities.

 **Write a SQL query to retrieve only the CustomerName and City columns from the Customers table.**

SELECT CustomerName, City

FROM Customers;

3. Unique Cities for Delivery Network Expansion

The logistics team wants to know all the different cities where customers are located (no duplicates).

 **Write a SQL query to retrieve distinct values from the City column in the Customers table.**

SELECT DISTINCT City

FROM Customers;



4. High-Value Products Report

The product manager wants to analyse products priced over £50.

- Write a SQL query to retrieve all columns from the **Products** table where the Price is greater than 50.

```
SELECT *
FROM Products
WHERE Price > 50;
```

5. International Customers Targeting (USA & UK)

The marketing team is preparing campaigns for customers in the USA and UK.

- Write a SQL query to retrieve all columns from the **Customers** table where the Country is either 'USA' or 'UK'.

```
SELECT *
FROM Customers
WHERE Country IN ('USA', 'UK');
```

6. Recent Orders Report

Your team needs to analyze the latest orders.

- Write a SQL query to retrieve all columns from the **Orders** table, sorted by OrderDate in descending order.

```
SELECT *
FROM Orders
ORDER BY OrderDate DESC;
```

7. Mid-Range Products Listing

You are preparing a product list for items priced between £20 and £50.

- Write a SQL query to retrieve all columns from the **Products** table where the Price is between 20 and 50, ordered by descending Price.



```
SELECT*  
FROM Products  
WHERE Price BETWEEN 20 AND 50  
ORDER BY Price DESC;
```

8. Local Marketing in the US (Portland & Kirkland)

You're focusing on customers from Portland and Kirkland in the USA.

→ Write a SQL query to retrieve all columns from the Customers table where the Country is 'USA' and City is either 'Portland' or 'Kirkland', ordered by ascending CustomerName.

```
SELECT *  
FROM Customers  
WHERE Country = 'USA'  
AND City IN ('Portland', 'Kirkland')  
ORDER BY CustomerName ASC;
```

9. Customers from the UK or London

Prepare a list for promotional emails targeting customers from the UK or those based in London.

→ Write a SQL query to retrieve all columns from the Customers table where the Country is 'UK' or City is 'London', ordered by descending CustomerName.

```
SELECT *  
FROM Customers  
WHERE Country = 'UK'  
OR City = 'London'  
ORDER BY CustomerName DESC;
```



10. Product Inventory for Selected Categories

The inventory team needs products from Category 1 or 2, sorted alphabetically.

→ Write a SQL query to retrieve all columns from the Products table where the CategoryID is 1 or 2, ordered by ascending ProductName.

SELECT *

FROM Products

WHERE CategoryID IN (1, 2)

ORDER BY ProductName ASC;

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.

When it comes to Self-join this is when a table is joined with itself. This becomes useful when you want to compare rows within the same table.

Question: Which employee reports to which manager, when both employees and managers are stored in the same table

Select

Self-join

```
e.name AS employee,  
m.name as manager  
from employees e  
left join employees m
```

A Right Join returns all records from the right table, and the matched records from the left table. If there's no match, null appears for the left table's columns.

```
SELECT Orders.order_id, Customers.name  
FROM Orders
```



	<pre>RIGHT JOIN Customers ON Orders.customer_id = Customers.id;</pre>
Full join	<p>A full outer join returns all records from both tables, with null for missing matches on either side.</p> <p>Example</p> <pre>SELECT e.name AS Employee, c.name AS Contractor FROM Employees e FULL JOIN Contractors c ON e.id = c.id;</pre>
Inner join	<p>When it comes to Inner Join it returns only the records when there is a match in both tables.</p> <p>Example</p> <pre>SELECT Orders.order_id, Customers.name FROM Orders INNER JOIN Customers ON Orders.customer_id = Customers.id;</pre>
Cross join	<p>Cross Join returns the Cartesian product of two tables-all possible combinations of rows.</p> <p>Example</p> <pre>SELECT Products.name, Discounts.discount_code FROM Products CROSS JOIN Discounts;</pre>



Left join

Left join returns all records from the left table, and the matched records from the right table. If there's no match, again the word null will appear for the right table's columns.

Example

```
SELECT Customers.name, Orders.order_id  
FROM Customers  
LEFT JOIN Orders  
ON Customers.id = Orders.customer_id;
```



Day 3: Task 2

You are a junior data analyst at **Northwind Traders**. Your manager has assigned you a series of real-world business tasks. For each scenario:

1. **Write an SQL query** to solve the request. Write an SQL query to solve the request.
2. Run the query in MySQL Workbench using the Northwind database.
3. Paste your query in the box below each question as evidence of completion.

1. *Linking Products to Suppliers*

Management wants to know which supplier provides each product in the inventory.

-  **Write a SQL query to find the supplier of each product.**

```
SELECT  
    p.ProductName,  
    s.CompanyName AS SupplierName  
FROM  
    Products p  
JOIN  
    Suppliers s ON p.SupplierID = s.SupplierID  
ORDER BY  
    p.ProductName;
```

2. *Classifying Products by Category*

The category manager is reviewing how products are organized.

-  **Write a SQL query to find the category of each product.**

```
SELECT  
    p.ProductName,  
    c.CategoryName  
FROM  
    Products p  
JOIN  
    Categories c ON p.CategoryID = c.CategoryID  
ORDER BY  
    p.ProductName;
```



3. Category-Specific Product Report: Meat/Poultry

The food department wants to view all items in the *Meat/Poultry* category.

→ Write a SQL query to retrieve all products belonging to the *Meat/Poultry* category.

```
SELECT
    p.ProductName,
    c.CategoryName
FROM
    Products p
JOIN
    Categories c ON p.CategoryID = c.CategoryID
WHERE
    c.CategoryName = 'Meat/Poultry'
ORDER BY
    p.ProductName;
```

4. Complete Order Overview

The business team wants to see a detailed order list with customer and employee information.

→ Write a SQL query to retrieve the **Order ID**, **Order Date**, **Customer Name**, and **Employee Name** for all orders.

```
SELECT
    o.OrderID,
    o.OrderDate,
    c.CompanyName AS CustomerName,
    CONCAT(e.FirstName, ' ', e.LastName) AS EmployeeName
FROM
    Orders o
JOIN
```



```
Customers c ON o.CustomerID = c.CustomerID
```

JOIN

```
Employees e ON o.EmployeeID = e.EmployeeID
```

ORDER BY

```
o.OrderID;
```

5. Supply Chain Overview Report

Your manager wants to see the product name, its category, and the name of its supplier all in one report.

 Write a SQL query to retrieve the Product Name, Category Name, and Supplier Name for all products.

```
SELECT
```

```
p.ProductName,
```

```
c.CategoryName,
```

```
s.CompanyName AS SupplierName
```

FROM

```
Products p
```

JOIN

```
Categories c ON p.CategoryID = c.CategoryID
```

JOIN

```
Suppliers s ON p.SupplierID = s.SupplierID
```

ORDER BY

```
p.ProductName;
```

6. Yearly Order Summary – 1996

The team is auditing customer orders made in 1996.

 Write a SQL query to create a report for all the orders of 1996 and their customers.

```
WHERE YEAR(o.OrderDate) = 1996
```



7. Product Count by Category

The product team wants to know how many products exist under each category.

- Write a SQL query to retrieve all categories along with the number of products in each category.

```
SELECT  
    c.CategoryName,  
    COUNT(p.ProductID) AS ProductCount  
FROM Categories c  
LEFT JOIN Products p  
    ON c.CategoryID = p.CategoryID  
GROUP BY c.CategoryName  
ORDER BY ProductCount DESC;
```

8. Sales Volume Breakdown

The sales department wants to analyze how much of each product was ordered and at what price.

- Write a SQL query to retrieve all products with their prices and the quantity ordered for each product.

```
SELECT  
    p.ProductID,  
    p.ProductName,  
    p.Price,  
    SUM(od.Quantity) AS TotalQuantityOrdered  
FROM Products p  
LEFT JOIN OrderDetails od  
    ON p.ProductID = od.ProductID  
GROUP BY  
    p.ProductID,  
    p.ProductName,  
    p.Price  
ORDER BY TotalQuantityOrdered DESC;
```



Day 4: Task 1: SQL Practical

You are working as a **junior data analyst** for an international research organization analyzing countries, cities, and languages from around the world. Your job is to write and run SQL queries that provide useful global insights.

For each task:

1. Write the appropriate SQL query to meet the scenario's objective.
2. Run your query using MySQL Workbench and the world database.
3. Take a screenshot that clearly shows both your SQL code and the result table.
4. Paste the screenshot in the box provided for each question as evidence of completion.

Setting up the database:

1. Download world_db [here](#)
2. Follow each step to create your database [here](#)

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.

The highlighted my SQL formula for each question I didn't remember to screenshot until I completed the questions. I decided to highlight my answers for each question.

The screenshot shows the MySQL Workbench interface with two queries run against the 'world' database.

Query 1:

```
1 • SELECT COUNT(*) AS "Number of Cities in USA"
  2 FROM city
  3 WHERE CountryCode = 'USA';
  4
  5 • SELECT country_name, life_expectancy
  6 FROM country_stats
  7 ORDER BY life_expectancy DESC
  8 LIMIT 1;
```

Result Grid:

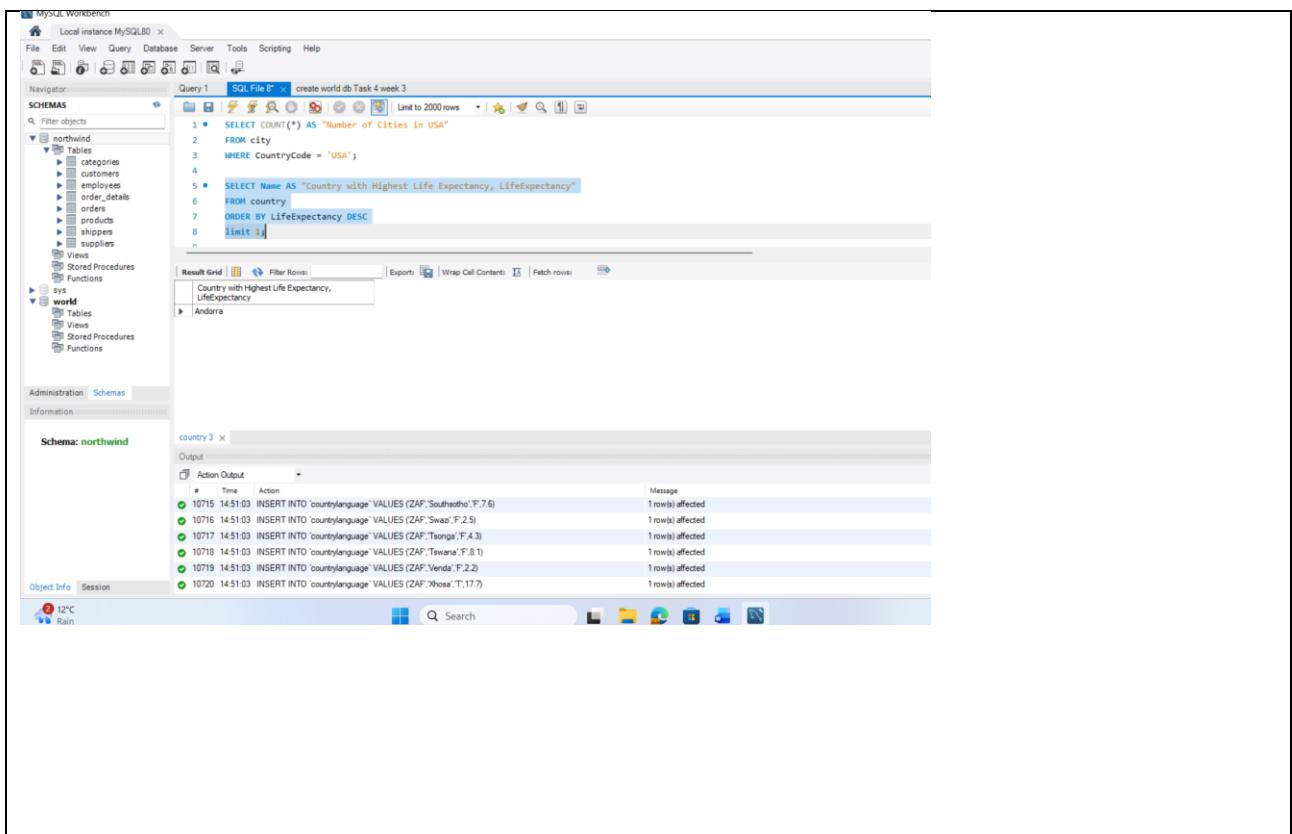
Number of Cities in USA
274

Query 2:

```
Output:
Action Output
# Time Action
10714 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Ndebele','F',9.1)
10715 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','South Sotho','F',7.6)
10716 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Swazi','F',2.5)
10717 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Tsonga','F',4.3)
10718 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Tswana','F',8.1)
10719 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Venda','F',2.2)
```



- 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.



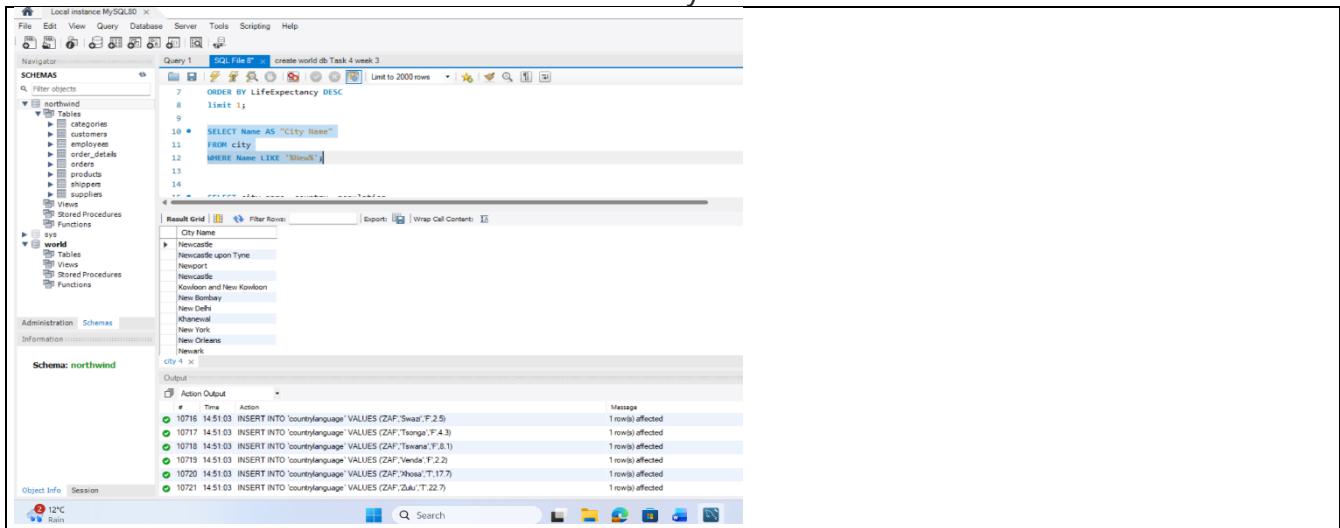
The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:


```

1 • SELECT COUNT(*) AS "Number of Cities in USA"
2   FROM city
3   WHERE CountryCode = "USA";
4
5 • SELECT Name AS "Country with Highest Life Expectancy", LifeExpectancy
6   FROM country
7   ORDER BY LifeExpectancy DESC
8   LIMIT 1;
      
```
- Result Grid:** Displays the result of the query, showing one row: "Andorra".
- Output Panel:** Shows the log of actions taken during the session, including multiple INSERT INTO statements for the 'countrylanguage' table.

- 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.



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:


```

7   ORDER BY LifeExpectancy DESC
8   LIMIT 1;
9
10 • SELECT Name AS "City_Name"
11   FROM city
12   WHERE Name LIKE "%New%";
```
- Result Grid:** Displays a list of cities containing the prefix 'New', such as New-York, Newcastle upon Tyne, Newark, and New Delhi.
- Output Panel:** Shows the log of actions taken during the session, including multiple INSERT INTO statements for the 'countrylanguage' table.



- 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.

```

File Edit View Query Database Server Tools Scripting Help
File Edit View Query Database Server Tools Scripting Help
Navigator: Local instance MySQL80 | Query 1: SQL File 8" x: create world db Task 4 week 3
SCHEMAS: Filter objects
northwind Tables: categories, customers, employees, order_details, orders, products, shippers, supplies
world Tables: city, country, countrylanguage, employee, nation, nationlanguage, region, regionlanguage
sys Functions
Administration Schemas Information
Schema: northwind
Result Grid: Filter Rows: Export/Import: Wrap Cell Content: Fetch rows:
ID Name CountryCode District Population
1024 Mumbai (Bombay) IND Maharashtra 10500000
2331 Seoul KOR Seoul 9981619
206 São Paulo BRA São Paulo 9968465
1890 Shanghai CHN Shanghai 9966300
2515 Ciudad de México MEX Distrito Federal 8591109
3580 Moscow RUS Moscow (City) 8389200
3793 New York USA New York 8008278
2822 Karachi PAK Sindh 8269365
3357 Istanbul TUR İstanbul 8279358
2515 Ciudad de México MEX Distrito Federal 8591109
3580 Moscow RUS Moscow (City) 8389200
3793 New York USA New York 8008278
city 5 x
Output: Action Output
# Time Action Message
10717 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Tsonga','F',4,3) 1 row(s) affected
10718 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Tswana','F',8,1) 1 row(s) affected
10719 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Venda','F',2,2) 1 row(s) affected
10720 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Xhosa','T',17,7) 1 row(s) affected
10721 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Zulu','T',22,7) 1 row(s) affected
10722 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZMB','Bemba','F',29,7) 1 row(s) affected
10723 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZMB','Chewa','F',5,7) 1 row(s) affected
Object Info Session
12°C Rain

```

- 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.

```

File Edit View Query Database Server Tools Scripting Help
File Edit View Query Database Server Tools Scripting Help
Navigator: Local instance MySQL80 | Query 1: SQL File 8" x: create world db Task 4 week 3
SCHEMAS: Filter objects
northwind Tables: categories, customers, employees, order_details, orders, products, shippers, supplies
world Tables: city, country, countrylanguage, employee, nation, nationlanguage, region, regionlanguage
sys Functions
Administration Schemas Information
Schema: northwind
Result Grid: Filter Rows: Export: Wrap Cell Content: Fetch rows:
City Name Population
Algiers 2185000
Cairo 2022000
Buenos Aires 2982146
Sydney 3276207
Melbourne 3865329
Dhaka 3612850
São Paulo 9968465
Rio de Janeiro 5598953
Salvador 2302832
Belo Horizonte 2139125
Fortaleza 2097757
city 5 x
Output: Action Output
# Time Action Message
10718 14:51:03 ORDER BY Population DESC 1 row(s) affected
10719 14:51:03 SELECT Name AS "City Name", Population 1 row(s) affected
10720 14:51:03 FROM city 1 row(s) affected
10721 14:51:03 WHERE Population > 2000000 1 row(s) affected
10722 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Tswana','F',8,1) 1 row(s) affected
10723 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Venda','F',2,2) 1 row(s) affected
10724 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Xhosa','T',17,7) 1 row(s) affected
10725 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Zulu','T',22,7) 1 row(s) affected
10726 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZMB','Bemba','F',29,7) 1 row(s) affected
10727 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZMB','Chewa','F',5,7) 1 row(s) affected
Object Info Session
11°C Rain

```

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.

The screenshot shows the MySQL Workbench interface with the following details:

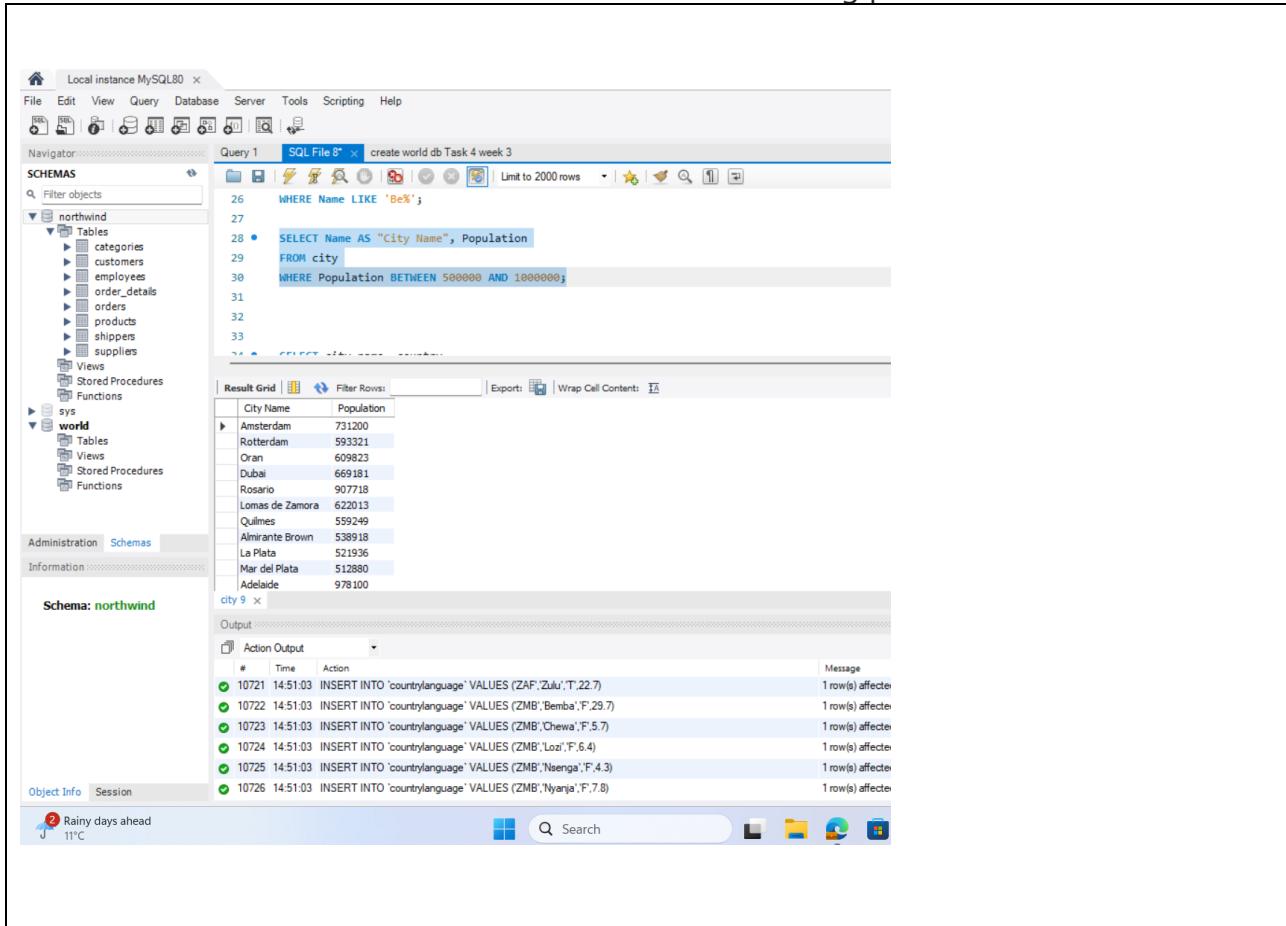
- File Bar:** Local Instance MySQL80 x, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (northwind, world), Tables (categories, customers, employees, order_details, orders, products, shippers, supplies), Views, Stored Procedures, Functions, sys, world.
- Query Editor:** Query 1 - SQL File 8*, create world db Task 4 week 3. The query is:

```
23
24 •    SELECT Name AS "City Name"
25     FROM city
26     WHERE Name LIKE 'Be%';
27
28
29
30 •    SELECT city_name, country, population
31     FROM ...
```
- Result Grid:** Shows the results of the query, listing cities starting with 'Be':

City Name
Béjaïa
Béchar
Benguela
Berazategui
Belize City
Belmopan
Belo Horizonte
Belém
Belford Roxo
Betim
Bento Gonçal...
- Object Info:** Schema: northwind.
- Action Output:** Shows the history of actions taken on the 'countrylanguage' table:

#	Time	Action
10719	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZAF','Venda','F',2.2)
10720	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZAF','Xhosa','T',17.7)
10721	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZAF','Zulu','T',22.7)
10722	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Bemba','F',29.7)
10723	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Chewa','F',5.5)
10724	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Lozi','F',6.4)
- Session:** Rainy days ahead.

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.



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (northwind, world), Tables (categories, customers, employees, order_details, orders, products, shippers, suppliers, Views, Stored Procedures, Functions).
- Query Editor:** SQL File 8*, create world db Task 4 week 3

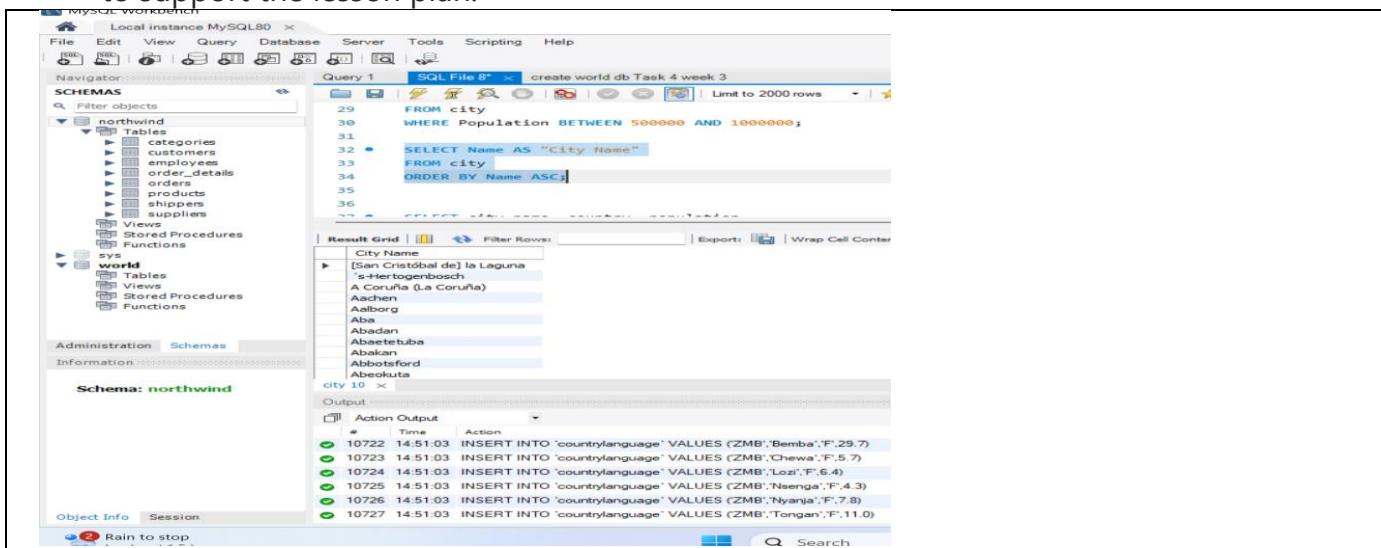

```

26 WHERE Name LIKE 'Be%';
27
28 • SELECT Name AS "City Name", Population
29   FROM city
30   WHERE Population BETWEEN 500000 AND 1000000;
31
32
33
34 • SELECT city_name, population
      
```
- Result Grid:**

City Name	Population
Amsterdam	731200
Rotterdam	593321
Oran	609823
Dubai	669181
Rosario	907718
Lomas de Zamora	622013
Quilmes	559249
Almirante Brown	538918
La Plata	521936
Mar del Plata	512880
Adelaide	978100
- Action Output:**

#	Time	Action	Message
10721	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZAF','Zulu','T',22.7)	1 row(s) affected
10722	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Bemba','F',29.7)	1 row(s) affected
10723	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Chewa','F',5.7)	1 row(s) affected
10724	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Lozi','F',6.4)	1 row(s) affected
10725	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Nsenga','F',4.3)	1 row(s) affected
10726	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Nyanja','F',7.8)	1 row(s) affected
- Object Info:** Rainy days ahead, 11°C.

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.



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (northwind, world), Tables (categories, customers, employees, order_details, orders, products, shippers, suppliers, Views, Stored Procedures, Functions).
- Query Editor:** SQL File 8*, create world db Task 4 week 3


```

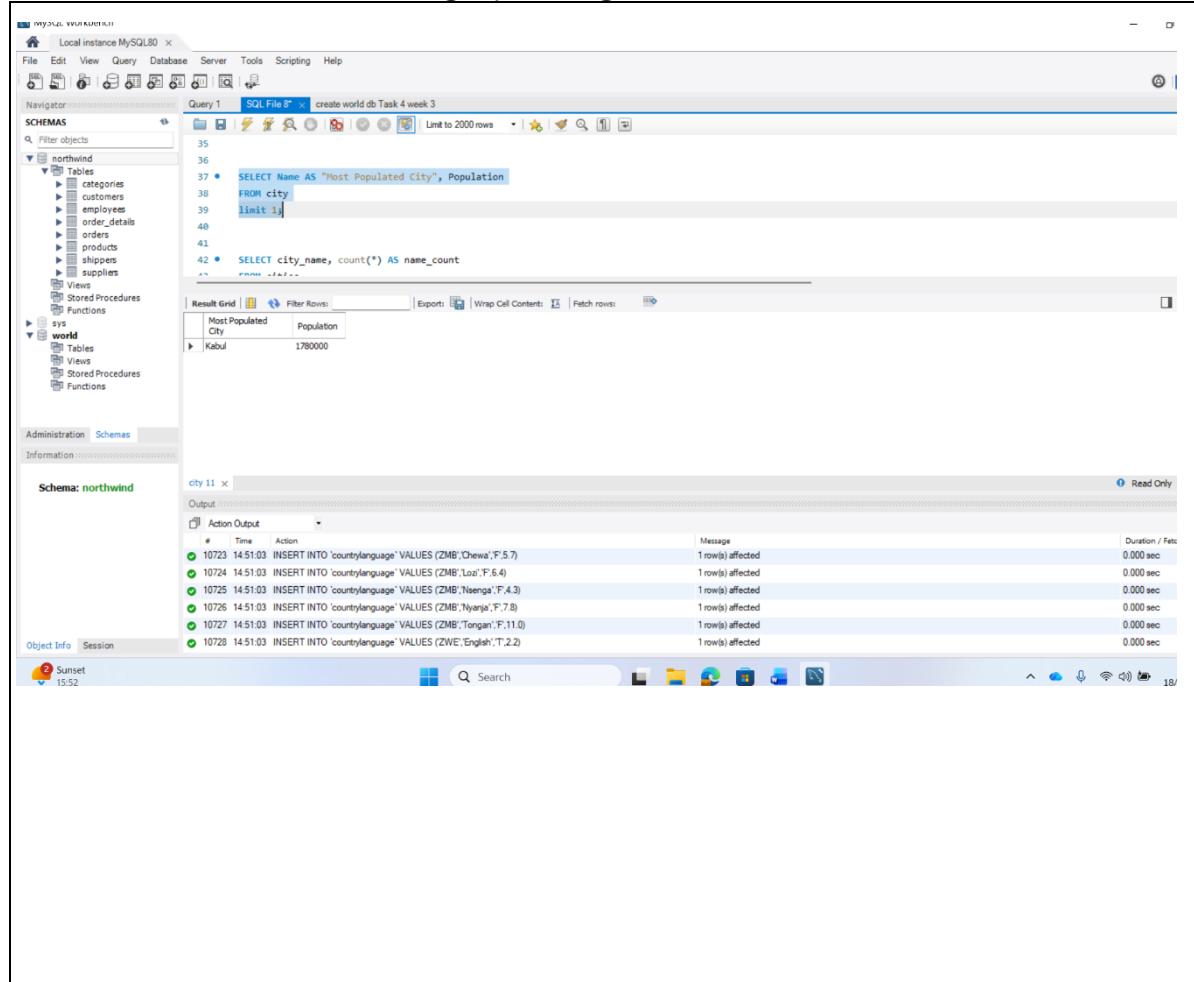
29
30   WHERE Population BETWEEN 500000 AND 1000000;
31
32 • SELECT Name AS "City Name"
33   FROM city
34   ORDER BY Name ASC;
      
```
- Result Grid:**

City Name
San Cristóbal de la Laguna
's-Hertogenbosch
A Coruña (La Coruña)
Aachen
Aalborg
Aba
Abadan
Abaetetuba
Abbotsford
Abeokuta
- Action Output:**

#	Time	Action	Message
10722	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Bemba','F',29.7)	1 row(s) affected
10723	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Chewa','F',5.7)	1 row(s) affected
10724	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Lozi','F',6.4)	1 row(s) affected
10725	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Nsenga','F',4.3)	1 row(s) affected
10726	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Nyanja','F',7.8)	1 row(s) affected
10727	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Tongan','F',11.0)	1 row(s) affected
- Object Info:** Rain to stop.



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.



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (northwind, sys, world), Tables, Views, Functions, Procedures.
- Query Editor:** Query 1 (SQL File 8) titled "create world db Task 4 week 3". The code is:

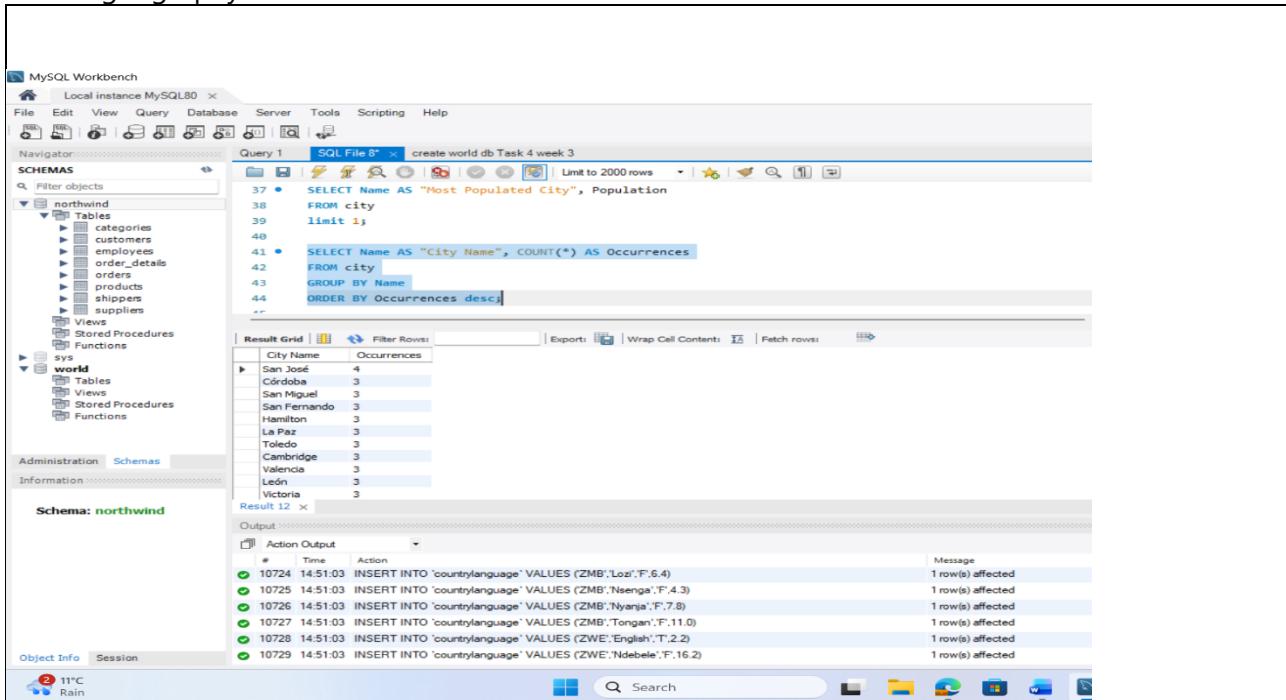

```

35
36
37 • SELECT Name AS "Most Populated City", Population
38 FROM city
39 limit 1;
40
41
42 • SELECT city_name, count(*) AS name_count
      
```
- Result Grid:** Shows the result of the query:

Most Populated	Population
Kabul	1780000
- Output Tab:** Action Output (10223 to 10272 actions listed).
- Object Info:** Session tab.



- 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.



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (northwind), Tables (categories, customers, employees, order_details, orders, products, shippers, suppliers), Views, Stored Procedures, Functions.
- Query Editor:** SQL File 8" x, create world db Task 4 week 3


```

37 •    SELECT Name AS "Most Populated City", Population
38     FROM city
39     limit 1;
40
41 •    SELECT Name AS "City Name", COUNT(*) AS Occurrences
42     FROM city
43     GROUP BY Name
44     ORDER BY Occurrences desc;
```
- Result Grid:**

City Name	Occurrences
San José	4
Córdoba	3
San Miguel	3
San Fernando	3
Hamilton	3
La Paz	3
Toledo	3
Cambridge	3
Valencia	3
León	3
Victoria	3
- Output:** Action Output

#	Time	Action	Message
10724	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Lozi','F',6.4)	1 row(s) affected
10725	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Nsenga','F',4.3)	1 row(s) affected
10726	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Nyanga','F',7.8)	1 row(s) affected
10727	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Tongan','F',11.0)	1 row(s) affected
10728	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','English','T',2.2)	1 row(s) affected
10729	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Ndebele','F',16.2)	1 row(s) affected

- 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.



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for database management tasks. The left sidebar displays the 'SCHEMAS' tree, which includes the 'northwind' schema (containing tables like categories, customers, employees, etc.) and the 'world' schema (containing tables like cities, countries, languages, etc.). The 'Information' section at the bottom of the sidebar shows the current schema is 'northwind'. The main area contains a 'Query 1' tab titled 'SQL File 8*' with the following SQL code:

```
33 FROM city
34 ORDER BY Name ASC;
35
36
37 • SELECT Name AS "Most Populated City", Population
38 FROM city
39 limit 1;
40
41 • SELECT Name AS "City Name", COUNT(*) AS Occurrences
42 FROM city
43 GROUP BY Name
44 ORDER BY Occurrences desc;
45
46 • SELECT name AS "City with Lowest Population", Population
47
48 FROM city
49 ORDER BY Population ASC
50 limit 1
51
52
```

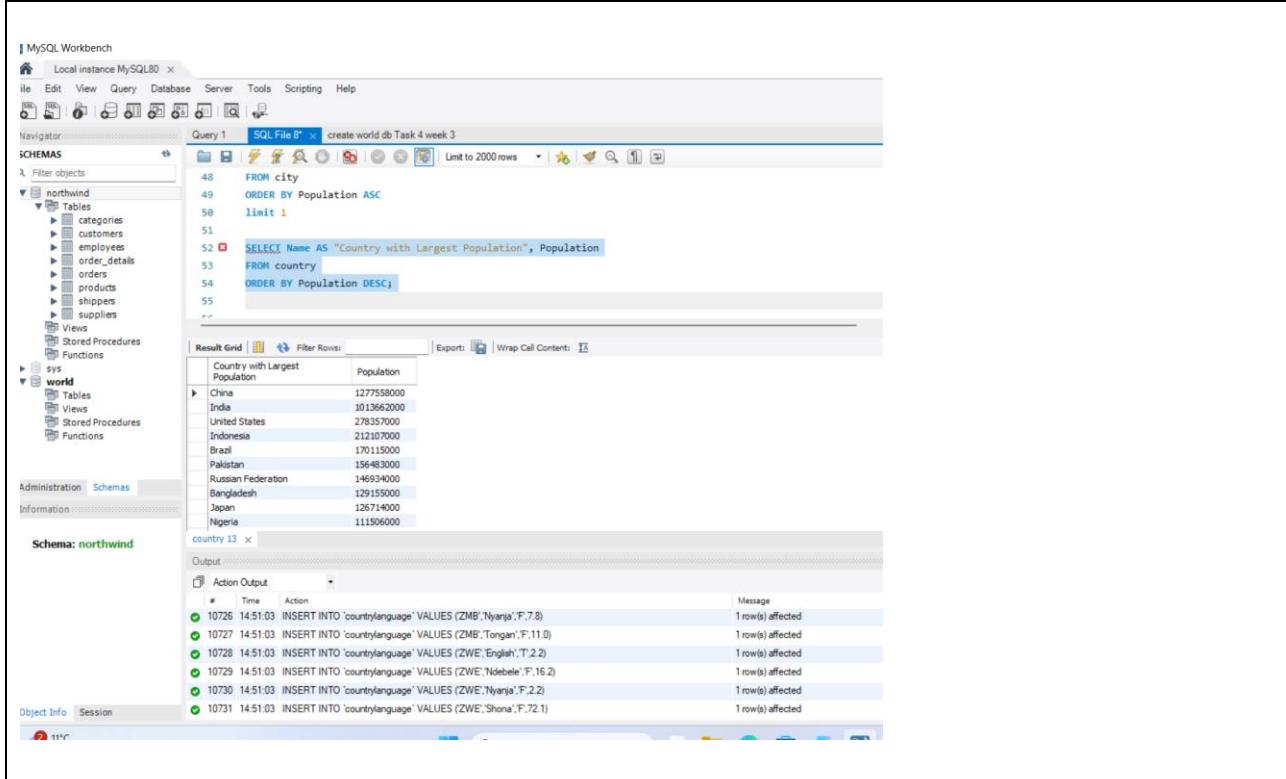
The 'Output' pane below the query editor shows the results of the executed queries, specifically the INSERT statements into the 'countrylanguage' table:

#	Time	Action	Message
10725	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Nsenga','F',4.3)	1 row(s) affected
10726	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Nyanja','F',7.8)	1 row(s) affected
10727	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Tongan','F',11.0)	1 row(s) affected
10728	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','English','T',2.2)	1 row(s) affected
10729	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Ndebele','F',16.2)	1 row(s) affected
10730	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Nyanja','F',2.2)	1 row(s) affected

The bottom of the interface includes a weather icon (11°C Rain), a search bar, and a toolbar with various icons for file operations.



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.



```

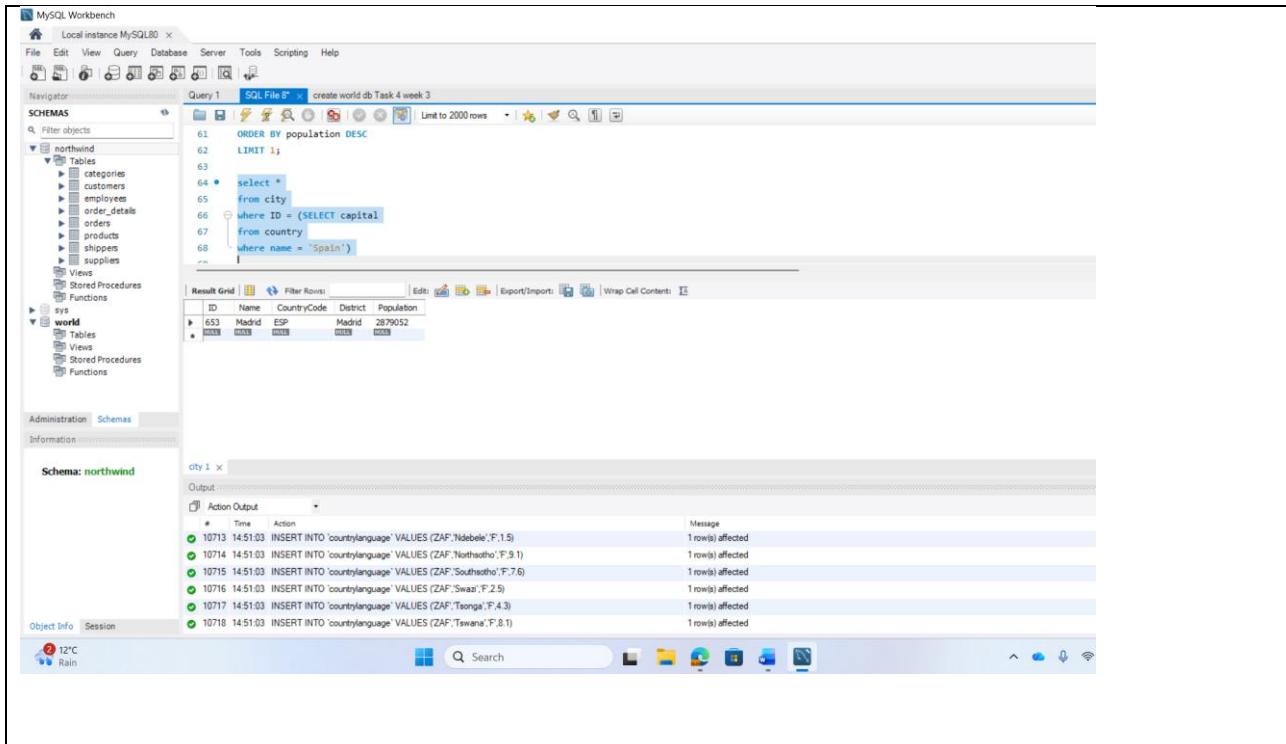
MySQL Workbench
Local instance MySQL8.0 x
File Edit View Query Database Server Tools Scripting Help
Navigator: Query 1 SQL File B create world db Task 4 week 3
SCHEMAS
northwind Tables
categories customers employees order_details orders products shippers suppliers Views Stored Procedures Functions
world Tables Views Stored Procedures Functions
Administration Schemas Information
Schema: northwind
country 13 x
Output
Action Output
# Time Action
10726 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZMB','Nyanya','F',7.8) 1 row(s) affected
10727 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZMB','Tongan','F',11.0) 1 row(s) affected
10728 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZWE','English','T',2.2) 1 row(s) affected
10729 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZWE','Ndebele','F',16.2) 1 row(s) affected
10730 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZWE','Nyanya','F',2.2) 1 row(s) affected
10731 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZWE','Shona','F',72.1) 1 row(s) affected
Object Info Session
11°C

```

The screenshot shows the MySQL Workbench interface with the 'Query 1' tab selected. The query retrieves the top 1 country by population from the 'country' table in the 'world' schema. The results are displayed in a grid:

Country with Largest Population	Population
China	1277558000
India	1013662000
United States	278357000
Indonesia	212107000
Brazil	170115000
Pakistan	156482000
Russian Federation	146934000
Bangladesh	129155000
Japan	126714000
Nigeria	111506000

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.



```

MySQL Workbench
Local instance MySQL8.0 x
File Edit View Query Database Server Tools Scripting Help
Navigator: Query 1 SQL File B create world db Task 4 week 3
SCHEMAS
northwind Tables
categories customers employees order_details orders products shippers suppliers Views Stored Procedures Functions
world Tables Views Stored Procedures Functions
Administration Schemas Information
Schema: northwind
city 1 x
Output
Action Output
# Time Action
10713 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Ndebele','F',1.5) 1 row(s) affected
10714 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Northern Sotho','F',9.1) 1 row(s) affected
10715 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','South Sotho','F',7.6) 1 row(s) affected
10716 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Swazi','F',2.5) 1 row(s) affected
10717 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Tsonga','F',4.3) 1 row(s) affected
10718 14:51:03 INSERT INTO `countrylanguage` VALUES ('ZAF','Tswana','F',8.1) 1 row(s) affected
Object Info Session
12°C Rain

```

The screenshot shows the MySQL Workbench interface with the 'Query 1' tab selected. The query retrieves the capital city of Spain from the 'city' table in the 'world' schema. The results are displayed in a grid:

ID	Name	CountryCode	District	Population
653	Madrid	ESP	Madrid	2879052

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.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80 X, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (northwind, world), Tables (categories, customers, employees, order_details, orders, products, shippers, suppliers), Views, Stored Procedures, Functions.
- Query Editor:** Query 1 (SQL File 8*) - create world db Task 4 week 3


```

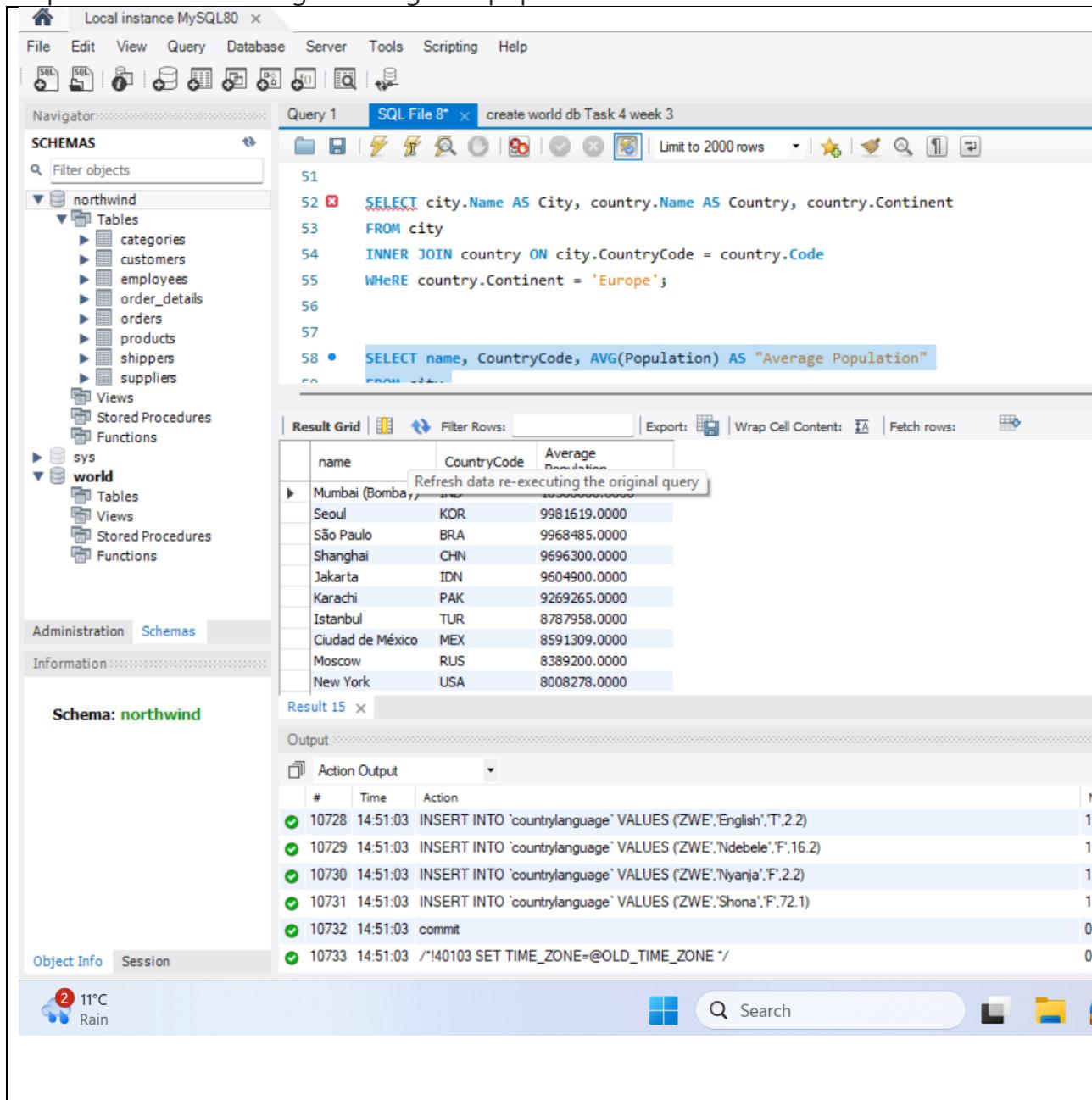
45 • SELECT name AS "City with Lowest Population", Population
46
47
48 FROM city
49 ORDER BY Population ASC
50
51 limit 1
52
      
```
- Result Grid:** Shows the results of the query:

City	Country	Continent
Tirana	Albania	Europe
Andorra la Vella	Andorra	Europe
Wien	Austria	Europe
Graz	Austria	Europe
Linz	Austria	Europe
Salzburg	Austria	Europe
Innsbruck	Austria	Europe
Klagenfurt	Austria	Europe
Antwerpen	Belgium	Europe
Gent	Belgium	Europe
Charleroi	Belgium	Europe
- Action Output:** Shows a log of actions taken by the session:

#	Time	Action	Message
10727	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZMB','Tongan','F',11,0)	1 row(s) affected
10728	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','English','T',2,2)	1 row(s) affected
10729	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Ndebele','F',16,2)	1 row(s) affected
10730	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Nyanja','F',2,2)	1 row(s) affected
10731	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Shona','F',72,1)	1 row(s) affected
10732	14:51:03	commit	0 row(s) affected



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.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the Navigator with Schemas (northwind, sys, world) and their respective tables, views, stored procedures, and functions. The main area shows a query editor with the following SQL code:

```

51
52  SELECT city.Name AS City, country.Name AS Country, country.Continent
53  FROM city
54  INNER JOIN country ON city.CountryCode = country.Code
55  WHERE country.Continent = 'Europe';
56
57
58 •  SELECT name, CountryCode, AVG(Population) AS "Average Population"
      FROM ...
  
```

The Result Grid shows the output of the query for cities in Europe:

name	CountryCode	Average Population
Mumbai (Bombay)	IND	9981619.0000
Seoul	KOR	9968485.0000
São Paulo	BRA	996300.0000
Shanghai	CHN	9696300.0000
Jakarta	IDN	9604900.0000
Karachi	PAK	9269265.0000
Istanbul	TUR	8787958.0000
Ciudad de México	MEX	8591309.0000
Moscow	RUS	8389200.0000
New York	USA	8008278.0000

The Action Output pane shows the history of actions taken:

#	Time	Action
10728	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','English','T',2.2)
10729	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Ndebele','F',16.2)
10730	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Nyanja','F',2.2)
10731	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Shona','F',72.1)
10732	14:51:03	commit
10733	14:51:03	/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */

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.



Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Query 1 SQL File 8* create world db Task 4 week 3

```

51
52  SELECT city.Name AS City, country.Name AS Country, country.Continent
53  FROM city
54  INNER JOIN country ON city.CountryCode = country.Code
55  WHERE country.Continent = 'Europe';
56
57
58  • SELECT name, CountryCode, AVG(Population) AS "Average Population"
59  FROM city
60  GROUP BY CountryCode, name
61  ORDER BY AVG(Population) DESC;
62
63  • SELECT c.Name AS "Capital City", c.Population AS Population, co.Name AS Country
64
65  FROM city c
66  INNER JOIN country co ON c.CountryCode = co.Code
67  WHERE co.Capital IN (SELECT DISTINCT capital FROM country)
68  ORDER BY c.Population DESC;
69
70

```

Schema: northwind

Output:

Action	Time	Action	Message
10729	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Ndebele','F',16.2)	1 row(s) affected
10730	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Nyanya','F',2.2)	1 row(s) affected
10731	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Shona','F',72.1)	1 row(s) affected
10732	14:51:03	commit	0 row(s) affected
10733	14:51:03	/!40103 SET TIME_ZONE=@OLD_TIME_ZONE /	0 row(s) affected
10734	14:51:03	/!40101 SET SQL_MODE=@OLD_SQL_MODE /	0 row(s) affected
10735	14:51:03	/!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS /	0 row(s) affected

Object Info Session

11°C Rain

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.

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Query 1 SQL File 8* create world db Task 4 week 3

```

63  • SELECT c.Name AS "Capital City", c.Population AS Population, co.Name AS Country
64
65  FROM city c
66  INNER JOIN country co ON c.CountryCode = co.Code
67  WHERE co.Capital IN (SELECT DISTINCT capital FROM country)
68  ORDER BY c.Population DESC;
69
70  • SELECT Name AS "Country with Low Population Density", Population, SurfaceArea,

```

Result Grid

Country with Low Population Density	Population	SurfaceArea	PopulationDensity
Greenland	56000	2166090.00	0.259
Svalbard and Jan Mayen	3200	62422.00	0.0513
Falkland Islands	2000	12173.00	0.1643
Pitcairn	50	49.00	1.0204
Western Sahara	293000	266000.00	1.1015
Mongolia	2662000	1566500.00	1.6993
French Guiana	181000	90000.00	2.0111
Namibia	1726000	824292.00	2.0939
Australia	18886000	7741220.00	2.4397
Suriname	417000	163265.00	2.5541

Result 16 x

Output:

Action	Time	Action	Message
10730	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Nyanya','F',2.2)	1 row(s) affected
10731	14:51:03	INSERT INTO `countrylanguage` VALUES ('ZWE','Shona','F',72.1)	1 row(s) affected
10732	14:51:03	commit	0 row(s) affected
10733	14:51:03	/!40103 SET TIME_ZONE=@OLD_TIME_ZONE /	0 row(s) affected
10734	14:51:03	/!40101 SET SQL_MODE=@OLD_SQL_MODE /	0 row(s) affected
10735	14:51:03	/!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS /	0 row(s) affected

Object Info Session

11°C Rain



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.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL8.0 x, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard database management icons.
- Navigator:** Shows the schema tree for the "northwind" database, which contains Tables (categories, customers, employees, order_details, products, shippers, supplies) and Views, Stored Procedures, Functions.
- Query Editor:** Title: "create world db Task 4 week 3". The query is a complex multi-table join and aggregation query:

```
66 INNER JOIN country co ON c.CountryCode = co.Code
67 WHERE co.Capital IN (SELECT DISTINCT capital FROM country)
68 ORDER BY c.Population DESC;
69
70 • SELECT Name AS "Country with Low Population Density", Population, SurfaceArea,
71 (Population / SurfaceArea) AS PopulationDensity
72 FROM country
73 WHERE (Population / SurfaceArea) > @
74 ORDER BY PopulationDensity ASC;
75
76 • SELECT c.Name AS City, c.Population, co.GNP, (co.GNP / c.Population) AS "GDP per Capita"
77
78 FROM city c
79
80 INNER JOIN country co ON c.CountryCode = co.Code
81
82 WHERE co.GNP IS NOT NULL
83 AND c.Population IS NOT NULL
84 AND c.Population > 0
85 AND co.GNP / c.Population > (SELECT AVG(co.GNP / c.Population) FROM city c INNER JOIN country co ON c.CountryCode = co.Code WHERE co.GNP IS NOT NULL AND c.Population IS NOT NULL)
```

- Output:** Action Output pane showing the execution log:

Action	Time	Message	Duration / Fetch
commit	10732 14:51:03	0 rows affected	0.000 sec
/40103 SET TIME_ZONE=@OLD_TIME_ZONE'	10733 14:51:03	0 rows affected	0.000 sec
/40101 SET SQL_MODE=@OLD_SQL_MODE'	10734 14:51:03	0 rows affected	0.000 sec
/40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS'	10735 14:51:03	0 rows affected	0.000 sec
/40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS'	10736 14:51:03	0 rows affected	0.000 sec
/40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT'	10737 14:51:03	0 rows affected	0.000 sec

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.

The screenshot shows the MySQL Workbench interface. On the left is the Navigator pane displaying the database schema for 'northwind' and 'world'. The main area contains a query editor titled 'Query 1' with the following SQL code:

```

81 WHERE co.GNP IS NOT NULL
82 AND c.Population IS NOT NULL
83 AND c.Population > 0
84 AND co.GNP / c.Population > (SELECT AVG(co.GNP / c.Population) FROM city c INNER JOIN country co ON c.Country
85
86
87
88 • SELECT *

```

The results grid shows data from the 'city' table, with columns ID, Name, CountryCode, and District. The action output log at the bottom lists several MySQL commands run at 14:51:03, all of which affected 0 rows.

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:

Additional Resources

- 1) [SQLBolt - Learn SQL - Introduction to SQL](#) : Welcome to SQLBolt, a series of interactive lessons and exercises designed to help you quickly learn SQL right in your browser.
- 2) [MySQL Basics Tutorial Series - YouTube](#): MySQL Basics Tutorial Series
- 3) Hackerrank Challenges: [How to use hackerrank challenges.odt](#)
- 4) Sakila DB Exercises:
 - a. Use this file to create the schema: [create sakila db.sql](#)
 - b. Access exercises here: [sakila db exercises combined.sql](#)

You can watch the video on how to open sql file:[How to create a schema in workbench.webm](#)



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.

