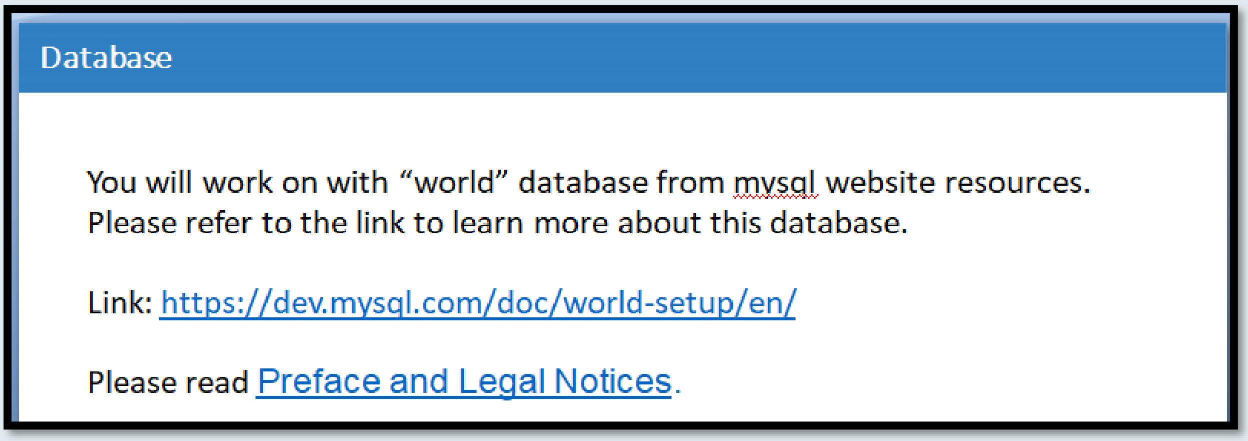


Database

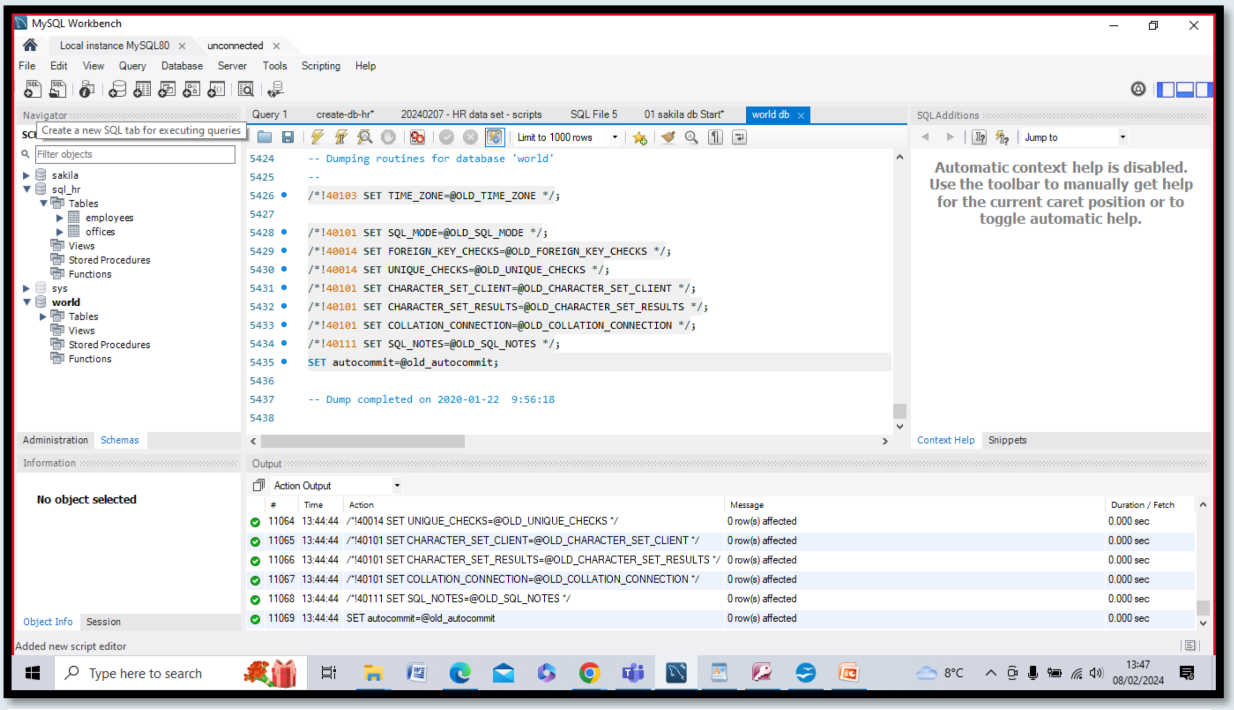
You will work on with “world” database from mysql website resources. Please refer to the link to learn more about this database.

Link: https://dev.mysql.com/doc/world-setup/en/

Please read Preface and Legal Notices.

Overall, the Preface provides legal and usage guidelines for users of the “world db” sample database, including copyright information, licensing restrictions, warrantee disclaimers, and guidelines for using the database.

- Import SQL Script "world db" and then execute the script. Refresh schemas and check if db "world" exits.



```
-- Dumping routines for database 'world'
--
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
SET autocommit=@old_autocommit;

-- Dump completed on 2020-01-22  9:56:18
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

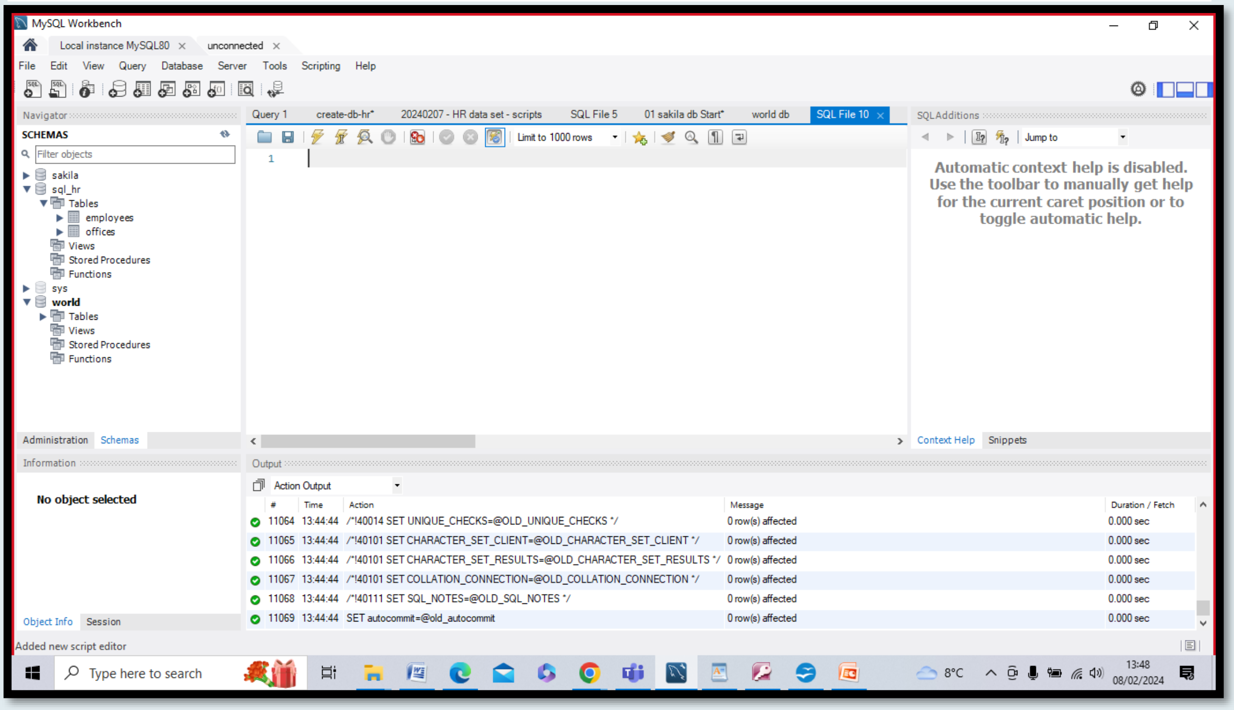
No object selected

```
#      Time      Action                                                              Message              Duration / Fetch
11064  13:44:44  /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */                    0 row(s) affected    0.000 sec
11065  13:44:44  /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */      0 row(s) affected    0.000 sec
11066  13:44:44  /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */    0 row(s) affected    0.000 sec
11067  13:44:44  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */      0 row(s) affected    0.000 sec
11068  13:44:44  /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */                            0 row(s) affected    0.000 sec
11069  13:44:44  SET autocommit=@old_autocommit                                      0 row(s) affected    0.000 sec
```

Added new script editor

- On your keyboard hold down Ctrl and T keys to open new query tab.
- Or you can do it from task bar
- Create word file to include at least 3

of your works screenshots.



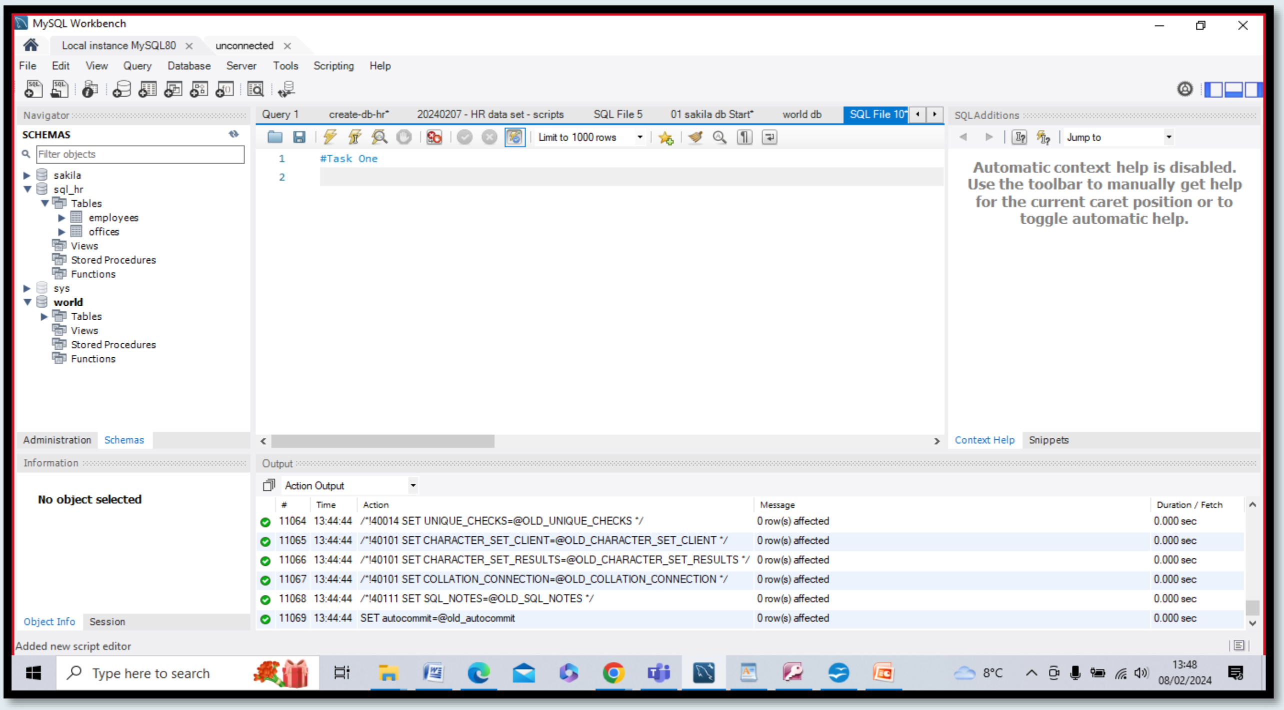
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

```
#      Time      Action                                                              Message              Duration / Fetch
11064  13:44:44  /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */                    0 row(s) affected    0.000 sec
11065  13:44:44  /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */      0 row(s) affected    0.000 sec
11066  13:44:44  /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */    0 row(s) affected    0.000 sec
11067  13:44:44  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */      0 row(s) affected    0.000 sec
11068  13:44:44  /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */                            0 row(s) affected    0.000 sec
11069  13:44:44  SET autocommit=@old_autocommit                                      0 row(s) affected    0.000 sec
```

Added new script editor

- Use # key to comment your tasks. You will need to save this query tab for trainers review later. Please use same query tab for all your tasks.



```
#Task One
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

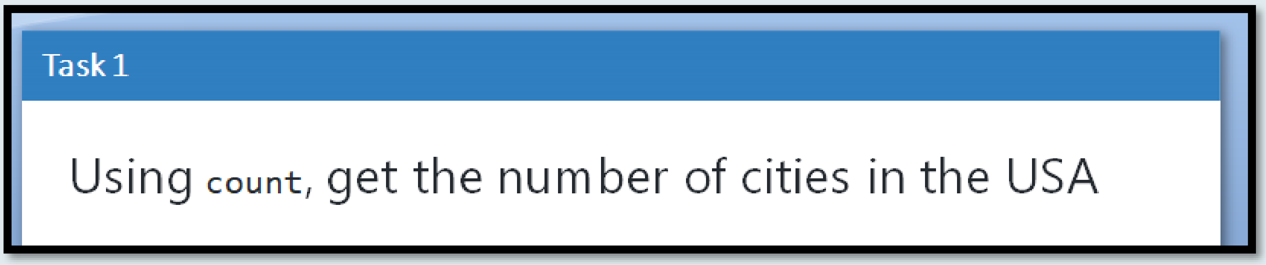
```
#      Time      Action                                                        Message             Duration / Fetch
11064  13:44:44  /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */              0 row(s) affected   0.000 sec
11065  13:44:44  /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */ 0 row(s) affected   0.000 sec
11066  13:44:44  /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */ 0 row(s) affected 0.000 sec
11067  13:44:44  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */ 0 row(s) affected   0.000 sec
11068  13:44:44  /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */                      0 row(s) affected   0.000 sec
11069  13:44:44  SET autocommit=@old_autocommit                                0 row(s) affected   0.000 sec
```

| This project concerns a new Schema called 'World db'. |
|---|
| **World db contains the following tables:** |
| • 'city', |
| • 'country', |
| • 'countrylanguage' |
| |
| **'City' table contains the following Columns:** |
| • 'ID', |
| • 'Name', |
| • 'CountryCode', |
| • 'District', |
| • 'Population' |
| |
| **'Country' table contains the following Columns:** |
| • 'Code', |
| • 'Name', |
| • 'Continent', |
| • 'Region', |
| • 'SurfaceArea', |

- 'IndepYear',
- 'Population',
- 'LifeExpectancy',
- 'GNP',
- 'GNPOld',
- 'LocalName',
- 'GovernmentForm',
- 'HeadOfState',
- 'Capital',
- 'Code2'

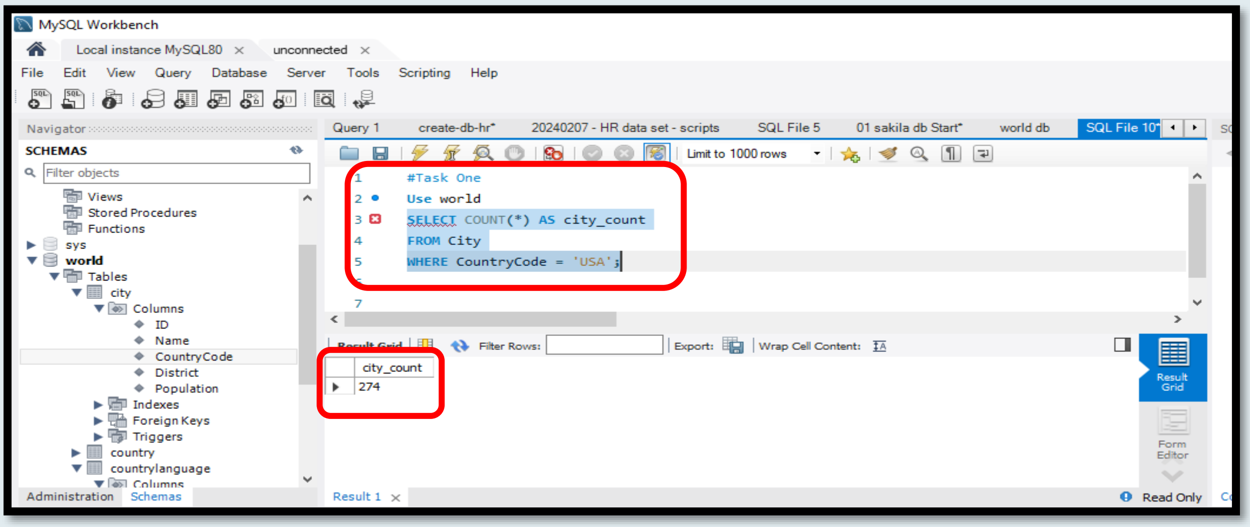
**'CountryLanguage' table contains the following Columns:**

- 'CountryCode',
- 'Language',
- 'IsOfficial',
- 'Percentage’



## Task 1

Using `count`, get the number of cities in the USA



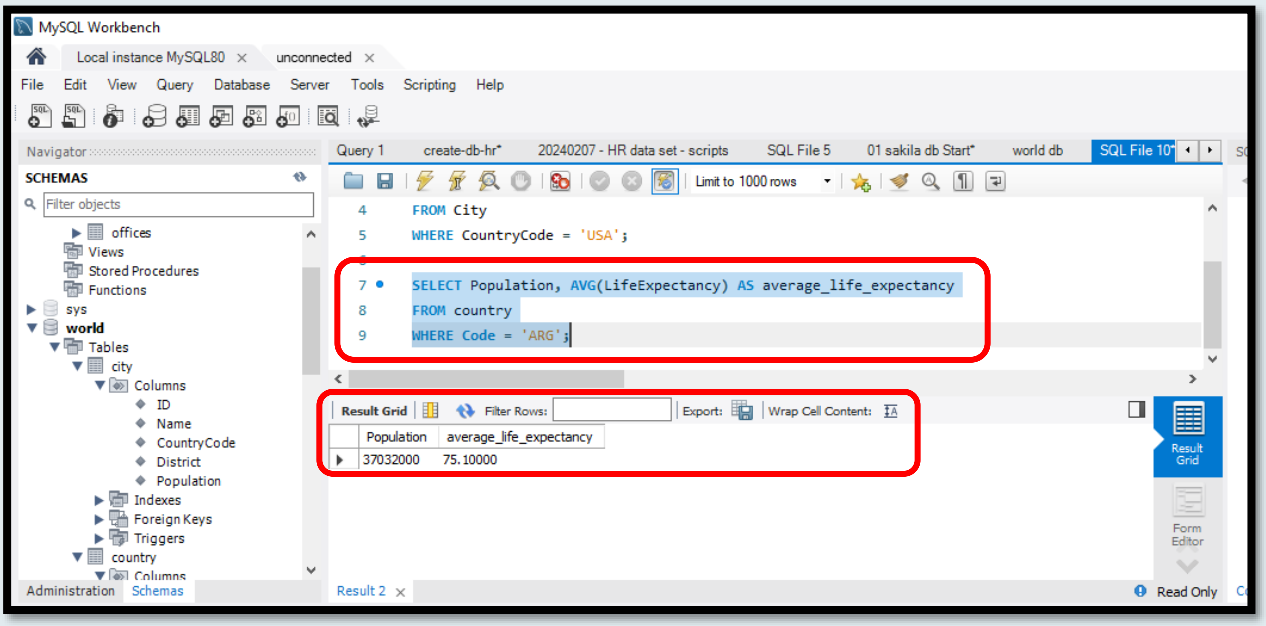
```
#Task One
Use world
SELECT COUNT(*) AS city_count
FROM City
WHERE CountryCode = 'USA';
```

```
city_count
274
```

The Query selects the count of rows (in cities) from the 'city' table selecting the 'CountryCode' column where this is 'USA', indicating that the cities are in the United States. This results in a figure for the number of cities in the USA, which is ‘274’.

## Task 2

Find out what the population and average life expectancy for people in Argentina (ARG) is.



```sql
FROM City
WHERE CountryCode = 'USA';

SELECT Population, AVG(LifeExpectancy) AS average_life_expectancy
FROM country
WHERE Code = 'ARG';
```

```
Population   average_life_expectancy
37032000     75.10000
```

'Population' is selected and used to calculate the average of 'LifeExpectancy' for Argentina from the 'country' table where the 'Code' column is 'ARG'. The result gives total population and average life expectancy for people in Argentina under two created headings:

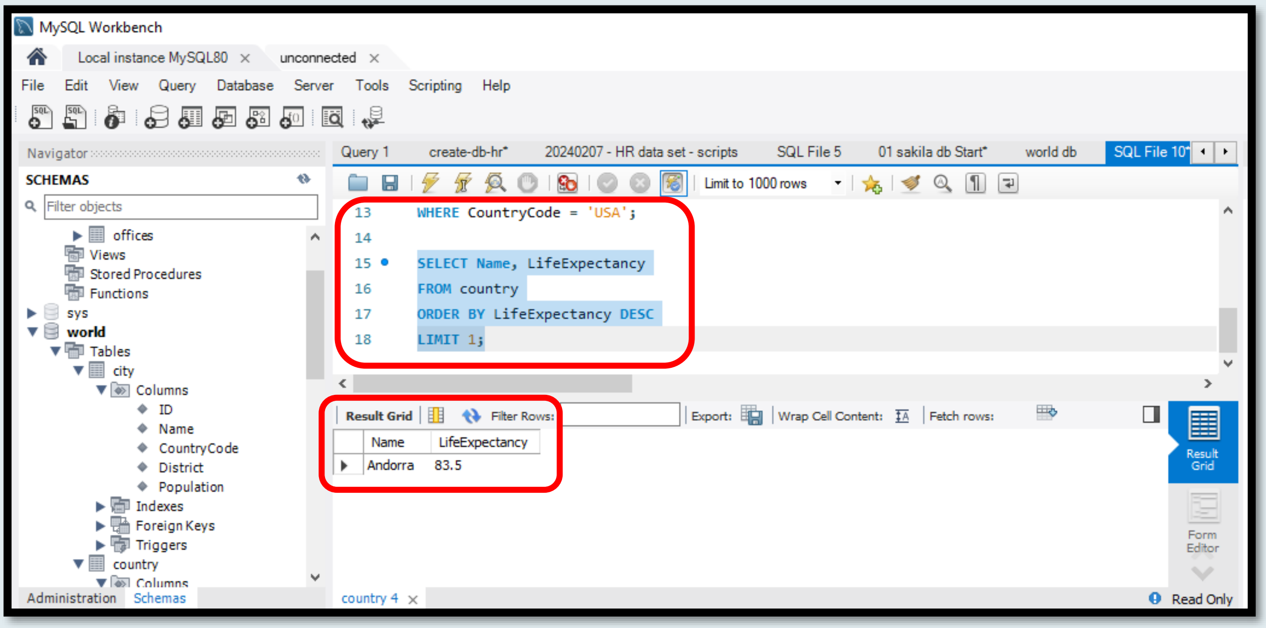
ARG

Population: 37,032,000

Average_Life_Expectancy: 75.1 years

## Task 3

Using `ORDER BY`, `LIMIT`, what country has the highest life expectancy?



```
WHERE CountryCode = 'USA';

SELECT Name, LifeExpectancy
FROM country
ORDER BY LifeExpectancy DESC
LIMIT 1;
```

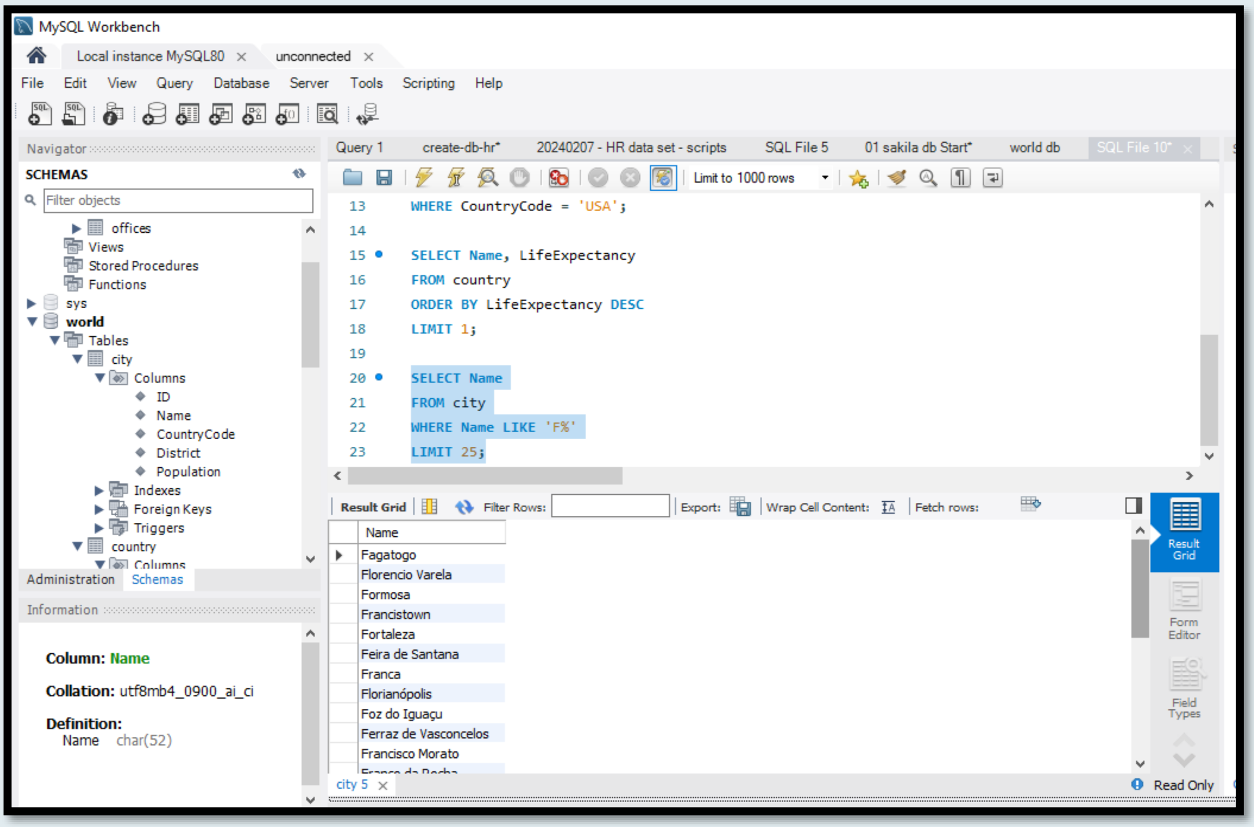
| Name | LifeExpectancy |
|---|---|
| Andorra | 83.5 |

The Query selects 'Name' (country name) and 'LifeExpectancy' columns from the 'country' table and orders the results by 'LifeExpectancy' in descending order using the ORDER BY clause. The LIMIT 1 clause ensures that only the first row (country with the highest life expectancy) is returned.

Andorra – 83.5 years average life expectancy

## Task 4

Select 25 cities around the world that start with the letter 'F' in a single SQL query.



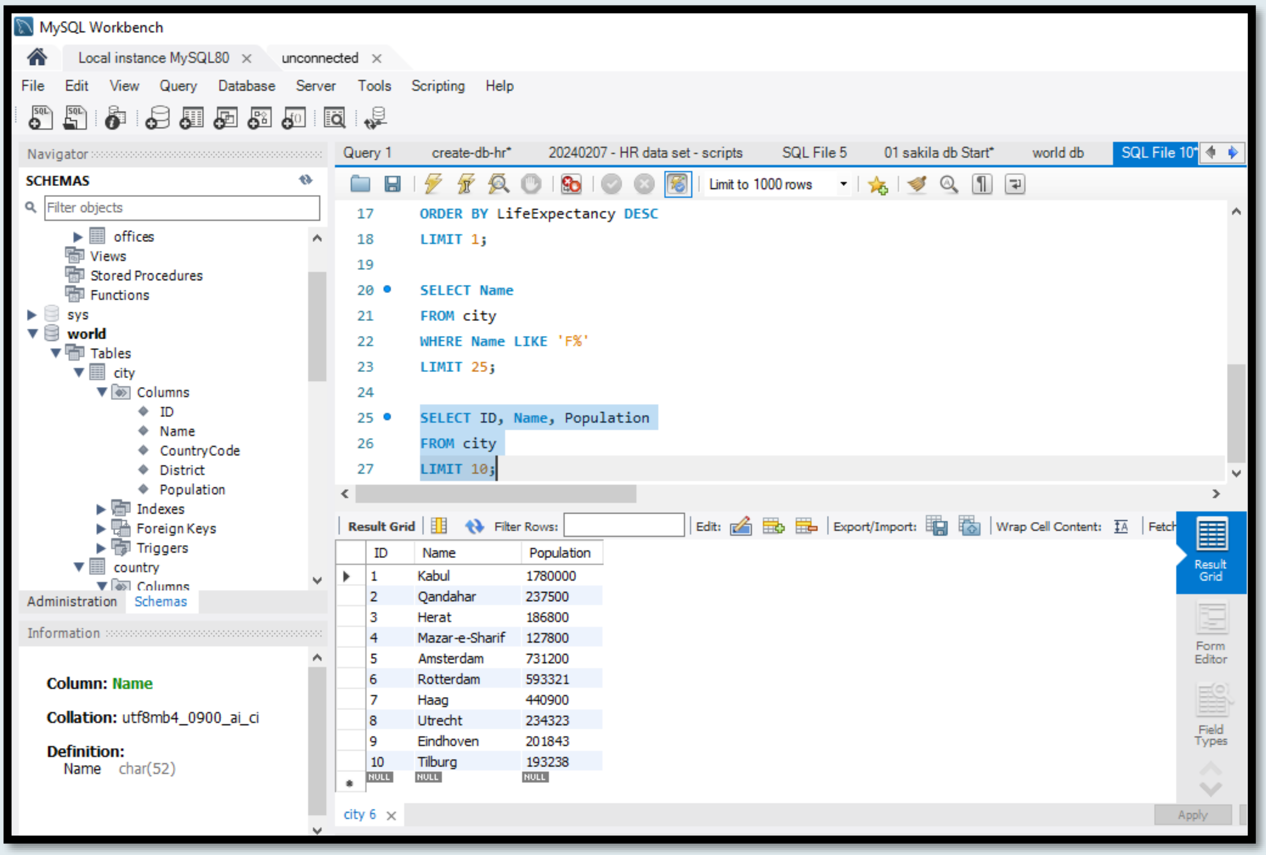
```
WHERE CountryCode = 'USA';

SELECT Name, LifeExpectancy
FROM country
ORDER BY LifeExpectancy DESC
LIMIT 1;

SELECT Name
FROM city
WHERE Name LIKE 'F%'
LIMIT 25;
```

| Name |
|---|
| Fagatogo |
| Florencio Varela |
| Formosa |
| Francistown |
| Fortaleza |
| Feira de Santana |
| Franca |
| Florianópolis |
| Foz do Iguaçu |
| Ferraz de Vasconcelos |
| Francisco Morato |

Tthe names of cities from the 'city' table are selected where the 'Name' column starts with the letter 'F'. This is done using the LIKE operator with the wildcard pattern 'F%', which matches any string that starts with 'F'. The LIMIT 25 clause ensures that only 25 rows are returned.

## Task 5

Create a SQL statement to display columns `Id`, `Name`, `Population` from the `city` table and limit results to first `10` rows only.



```
ORDER BY LifeExpectancy DESC
LIMIT 1;

SELECT Name
FROM city
WHERE Name LIKE 'F%'
LIMIT 25;

SELECT ID, Name, Population
FROM city
LIMIT 10;
```

| ID | Name | Population |
|---|---|---|
| 1 | Kabul | 1780000 |
| 2 | Qandahar | 237500 |
| 3 | Herat | 186800 |
| 4 | Mazar-e-Sharif | 127800 |
| 5 | Amsterdam | 731200 |
| 6 | Rotterdam | 593321 |
| 7 | Haag | 440900 |
| 8 | Utrecht | 234323 |
| 9 | Eindhoven | 201843 |
| 10 | Tilburg | 193238 |
| NULL | NULL | NULL |

Column: Name

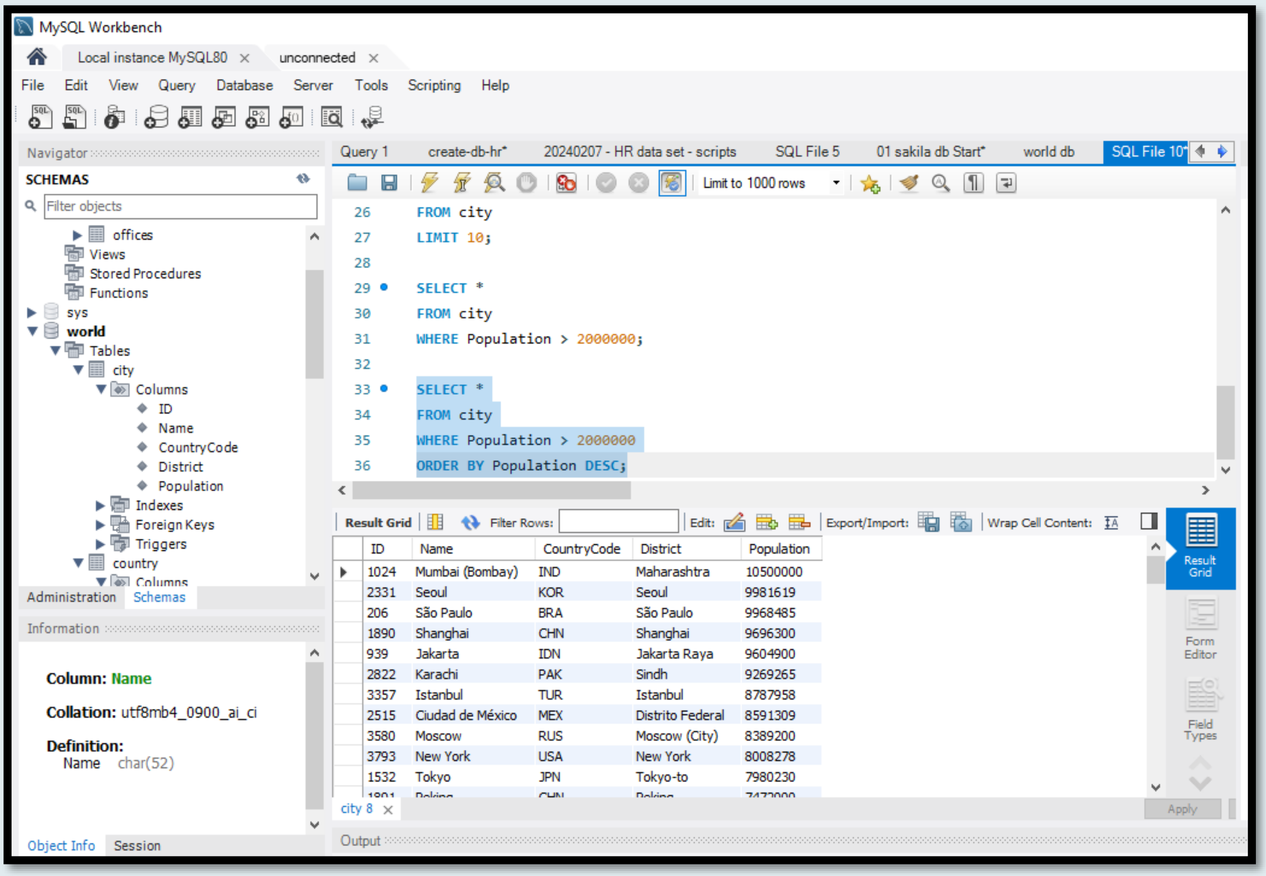
Collation: utf8mb4_0900_ai_ci

Definition:
Name char(52)

The specified columns from the 'city' table are selected; the LIMIT clause limits results to the first 10 rows.

## Task 6

•Create a SQL statement to find only those cities from `city` table whose population is larger than `2000000`.



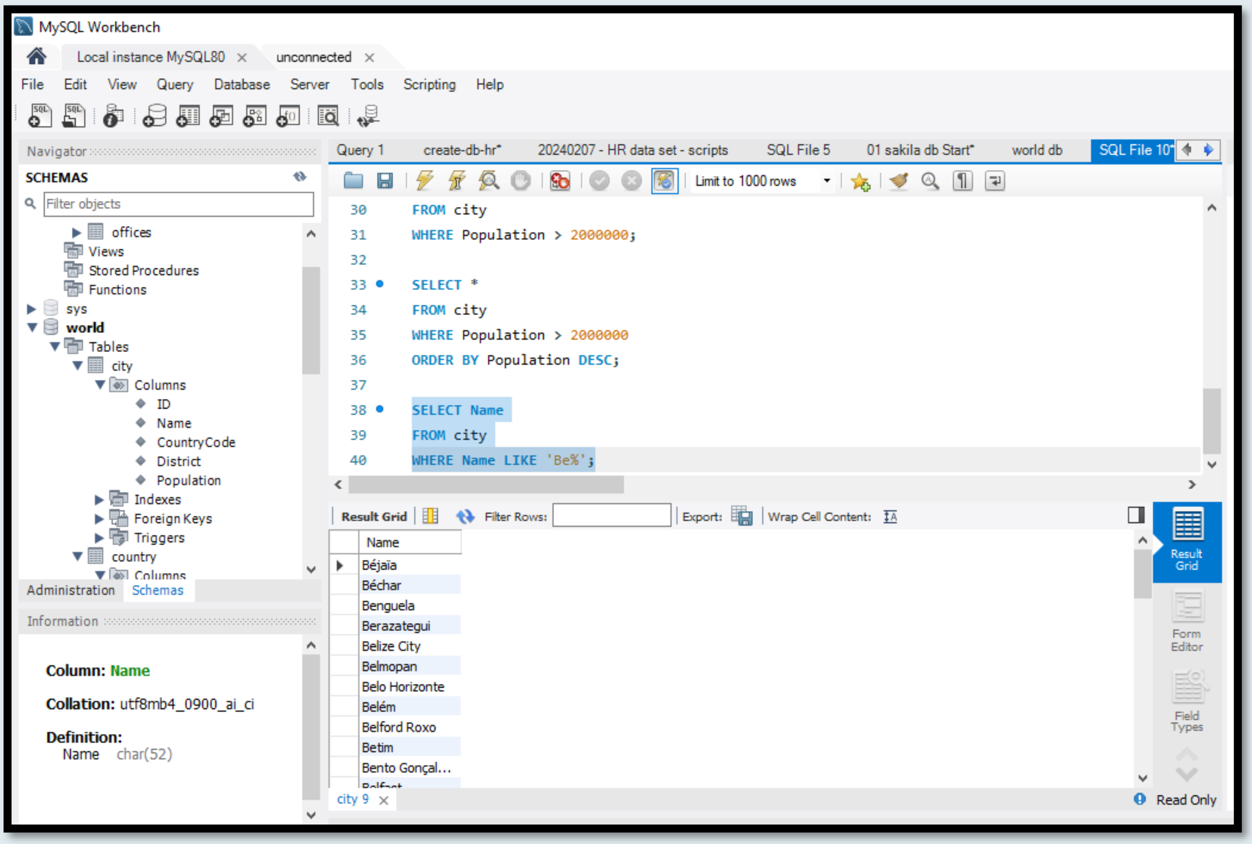
```
FROM city
LIMIT 10;

SELECT *
FROM city
WHERE Population > 2000000;

SELECT *
FROM city
WHERE Population > 2000000
ORDER BY Population DESC;
```

| ID | Name | CountryCode | District | Population |
|---|---|---|---|---|
| 1024 | Mumbai (Bombay) | IND | Maharashtra | 10500000 |
| 2331 | Seoul | KOR | Seoul | 9981619 |
| 206 | São Paulo | BRA | São Paulo | 9968485 |
| 1890 | Shanghai | CHN | Shanghai | 9696300 |
| 939 | Jakarta | IDN | Jakarta Raya | 9604900 |
| 2822 | Karachi | PAK | Sindh | 9269265 |
| 3357 | Istanbul | TUR | Istanbul | 8787958 |
| 2515 | Ciudad de México | MEX | Distrito Federal | 8591309 |
| 3580 | Moscow | RUS | Moscow (City) | 8389200 |
| 3793 | New York | USA | New York | 8008278 |
| 1532 | Tokyo | JPN | Tokyo-to | 7980230 |

This query selects all columns from the 'city' table where the 'Population' column is larger than 2,000,000. It only returned the rows (cities) that met the condition.

## Task 7

- Create a SQL statement to find all city names from city table whose name begins with “Be” prefix.



```sql
FROM city
WHERE Population > 2000000;

SELECT *
FROM city
WHERE Population > 2000000
ORDER BY Population DESC;

SELECT Name
FROM city
WHERE Name LIKE 'Be%';
```

| Name |
|---|
| Béjaïa |
| Béchar |
| Benguela |
| Berazategui |
| Belize City |
| Belmopan |
| Belo Horizonte |
| Belém |
| Belford Roxo |
| Betim |
| Bento Gonçal... |

**Column:** Name

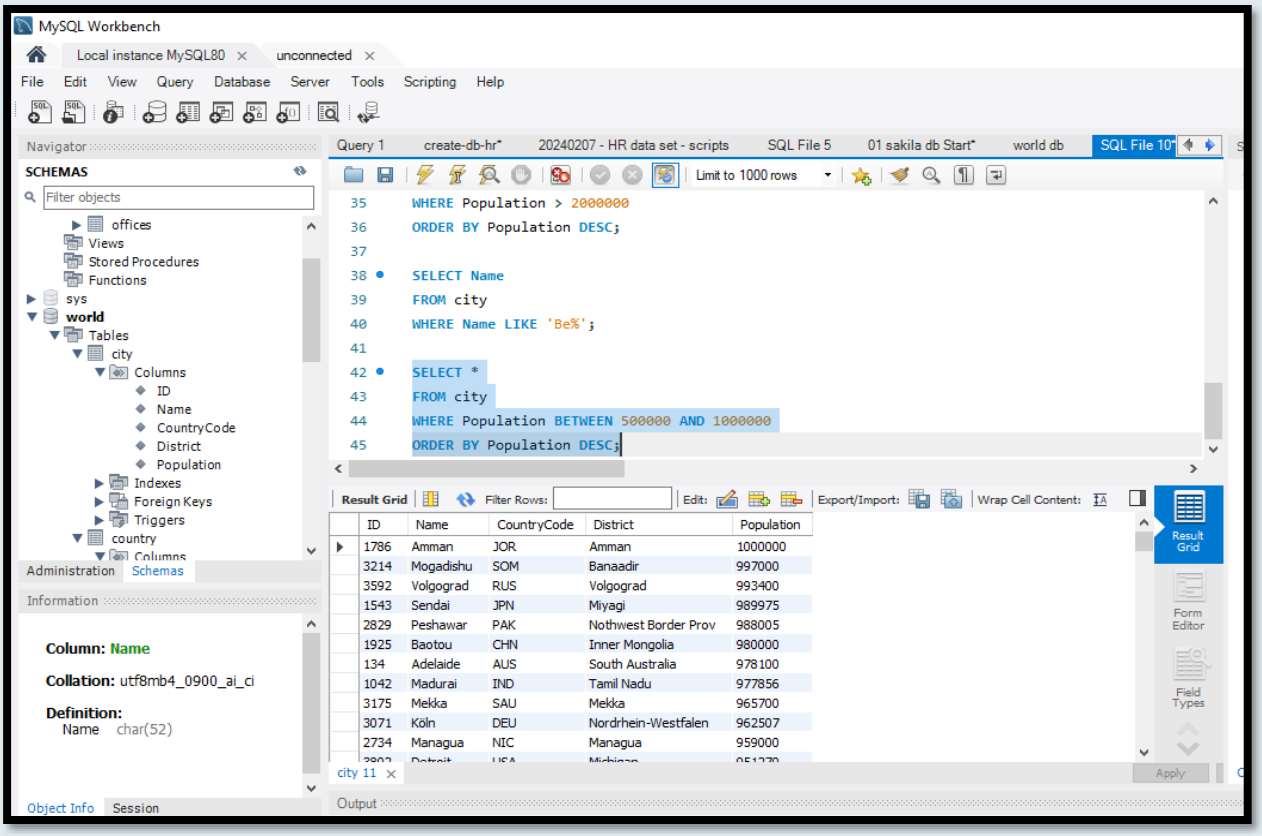
**Collation:** utf8mb4_0900_ai_ci

**Definition:**
Name char(52)

The 'Name' column is selected from the 'city' table, but only where the 'Name' column starts with the prefix "Be". This is done using the LIKE operator with the pattern 'Be%' (wildcard). It returns all city names that match this pattern.

## Task 8

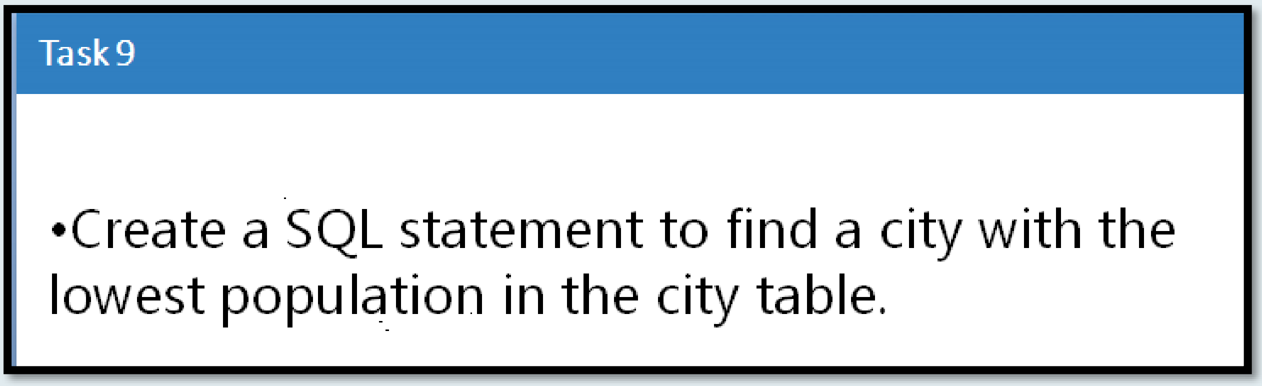
•Create a SQL statement to find only those cities from city table whose population is between 500000-1000000.



```
WHERE Population > 2000000
ORDER BY Population DESC;

SELECT Name
FROM city
WHERE Name LIKE 'Be%';

SELECT *
FROM city
WHERE Population BETWEEN 500000 AND 1000000
ORDER BY Population DESC;
```

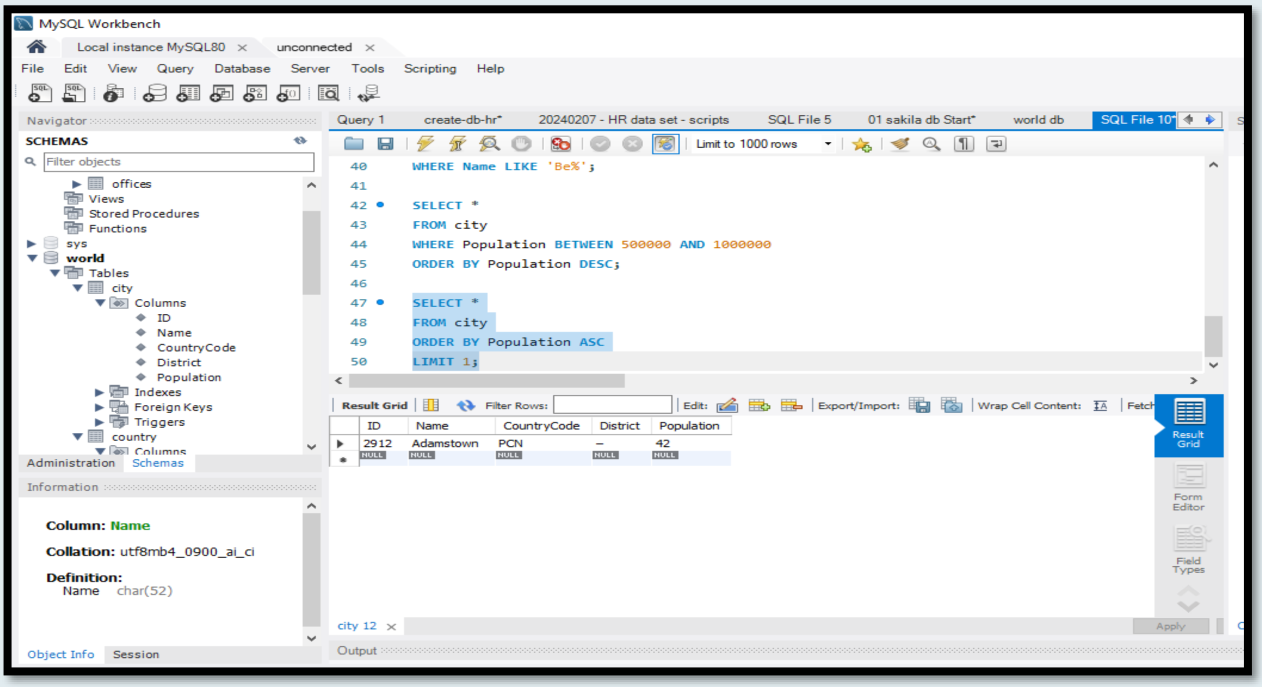
| ID | Name | CountryCode | District | Population |
|---|---|---|---|---|
| 1786 | Amman | JOR | Amman | 1000000 |
| 3214 | Mogadishu | SOM | Banaadir | 997000 |
| 3592 | Volgograd | RUS | Volgograd | 993400 |
| 1543 | Sendai | JPN | Miyagi | 989975 |
| 2829 | Peshawar | PAK | Nothwest Border Prov | 988005 |
| 1925 | Baotou | CHN | Inner Mongolia | 980000 |
| 134 | Adelaide | AUS | South Australia | 978100 |
| 1042 | Madurai | IND | Tamil Nadu | 977856 |
| 3175 | Mekka | SAU | Mekka | 965700 |
| 3071 | Köln | DEU | Nordrhein-Westfalen | 962507 |
| 2734 | Managua | NIC | Managua | 959000 |

This query selects all columns from the 'city' table where the 'Population' column is between 500,000 and 1,000,000 inclusive. This is done using the BETWEEN operator. It returns only the rows (cities) that meet the criteria.



## Task 9

•Create a SQL statement to find a city with the lowest population in the city table.



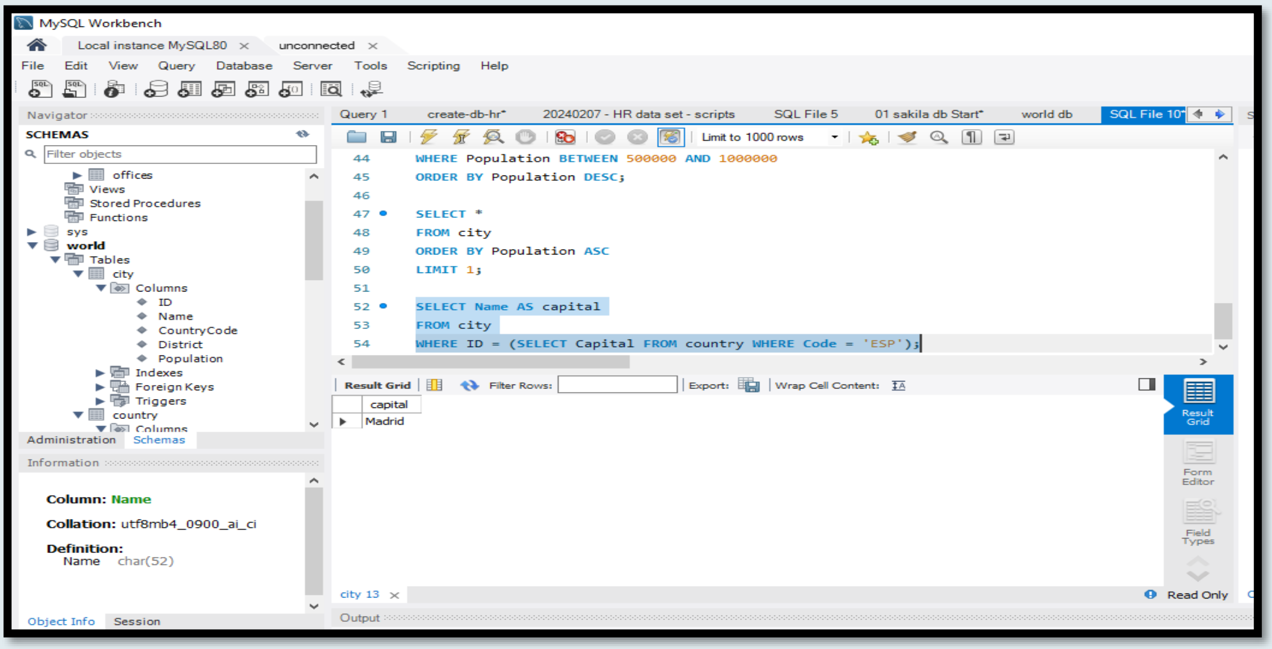
```
WHERE Name LIKE 'Be%';

SELECT *
FROM city
WHERE Population BETWEEN 500000 AND 1000000
ORDER BY Population DESC;

SELECT *
FROM city
ORDER BY Population ASC
LIMIT 1;
```

| ID | Name | CountryCode | District | Population |
|---|---|---|---|---|
| 2912 | Adamstown | PCN | – | 42 |
| NULL | NULL | NULL | NULL | NULL |

This selects all columns from the 'city' table and orders the result by population, but in ascending order (from lowest to highest). The LIMIT 1 clause then ensures that only the first row (city with the lowest population) is returned.

## Task 10

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



```sql
WHERE Population BETWEEN 500000 AND 1000000
ORDER BY Population DESC;

SELECT *
FROM city
ORDER BY Population ASC
LIMIT 1;

SELECT Name AS capital
FROM city
WHERE ID = (SELECT Capital FROM country WHERE Code = 'ESP');
```

```
capital
Madrid
```

**Column:** Name

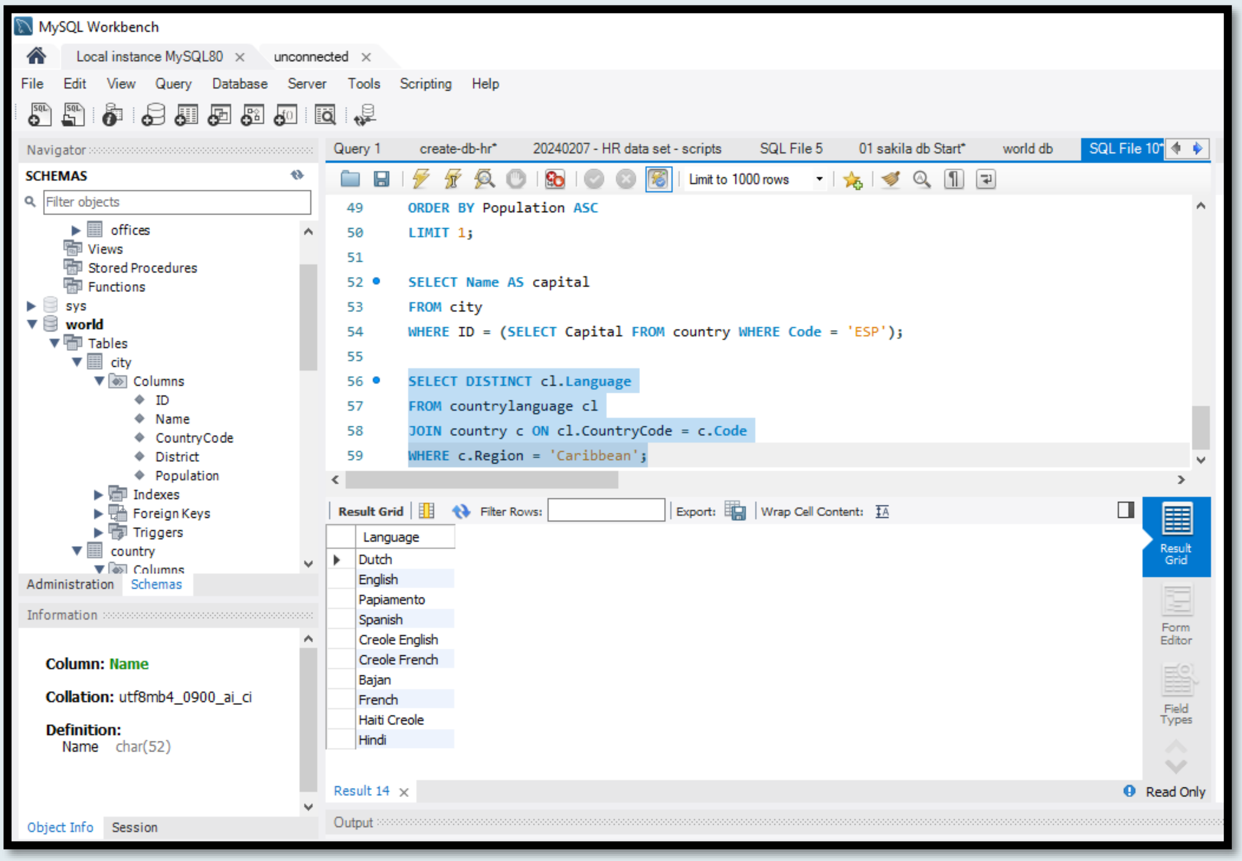
**Collation:** utf8mb4_0900_ai_ci

**Definition:**
Name char(52)

The 'Name' column from the 'city' table is selected and is used to retrieve the capital city's name by matching the 'Capital' column value (which represents the city ID of the capital) from the 'country' table, but only where the country code is 'ESP' (Spain).

## Task 11

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



```
ORDER BY Population ASC
LIMIT 1;

SELECT Name AS capital
FROM city
WHERE ID = (SELECT Capital FROM country WHERE Code = 'ESP');

SELECT DISTINCT cl.Language
FROM countrylanguage cl
JOIN country c ON cl.CountryCode = c.Code
WHERE c.Region = 'Caribbean';
```

| Language |
|---|
| Dutch |
| English |
| Papiamento |
| Spanish |
| Creole English |
| Creole French |
| Bajan |
| French |
| Haiti Creole |
| Hindi |

**Column:** Name

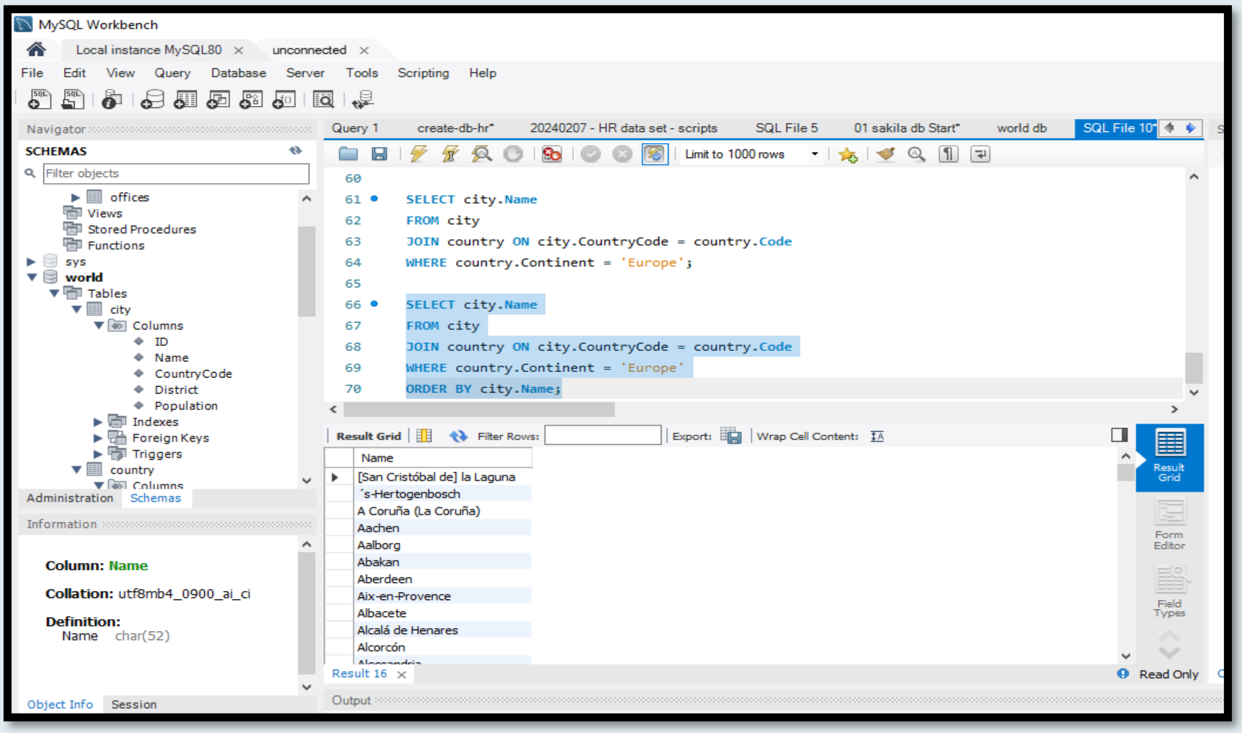
**Collation:** utf8mb4_0900_ai_ci

**Definition:**
Name char(52)

The task is to select the distinct languages from the 'countrylanguage' table and retrieve the languages spoken in the Caribbean region, this is done by joining with the 'country' table on the 'CountryCode' column and filtering rows where the region is 'Caribbean'.

## Task 12

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



```sql
SELECT city.Name
FROM city
JOIN country ON city.CountryCode = country.Code
WHERE country.Continent = 'Europe';

SELECT city.Name
FROM city
JOIN country ON city.CountryCode = country.Code
WHERE country.Continent = 'Europe'
ORDER BY city.Name;
```

| Name |
|---|
| [San Cristóbal de] la Laguna |
| ´s-Hertogenbosch |
| A Coruña (La Coruña) |
| Aachen |
| Aalborg |
| Abakan |
| Aberdeen |
| Aix-en-Provence |
| Albacete |
| Alcalá de Henares |
| Alcorcón |

**Column:** Name

**Collation:** utf8mb4_0900_ai_ci

**Definition:**
Name char(52)

This query selects the city names from the 'city' table by joining it with the 'country' table on the 'CountryCode' column. It then filters the rows where the continent is 'Europe' from the 'country' table.

1. **Primary key in country table:**

- In the 'country' table of the 'world' database, **the primary key is the 'Code' column**. This column uniquely identifies each country in the table.

2. **Primary key in city table:**

- In the 'city' table of the 'world' database, **the primary key is the 'ID' column**. This column uniquely identifies each city in the table.

3. **Primary key in countrylanguage table:**

- In the 'countrylanguage' table of the 'world' database, **the primary key is composed of the combination of the 'CountryCode' and 'Language' columns**. Together, these columns uniquely identify each language associated with a country in the table.

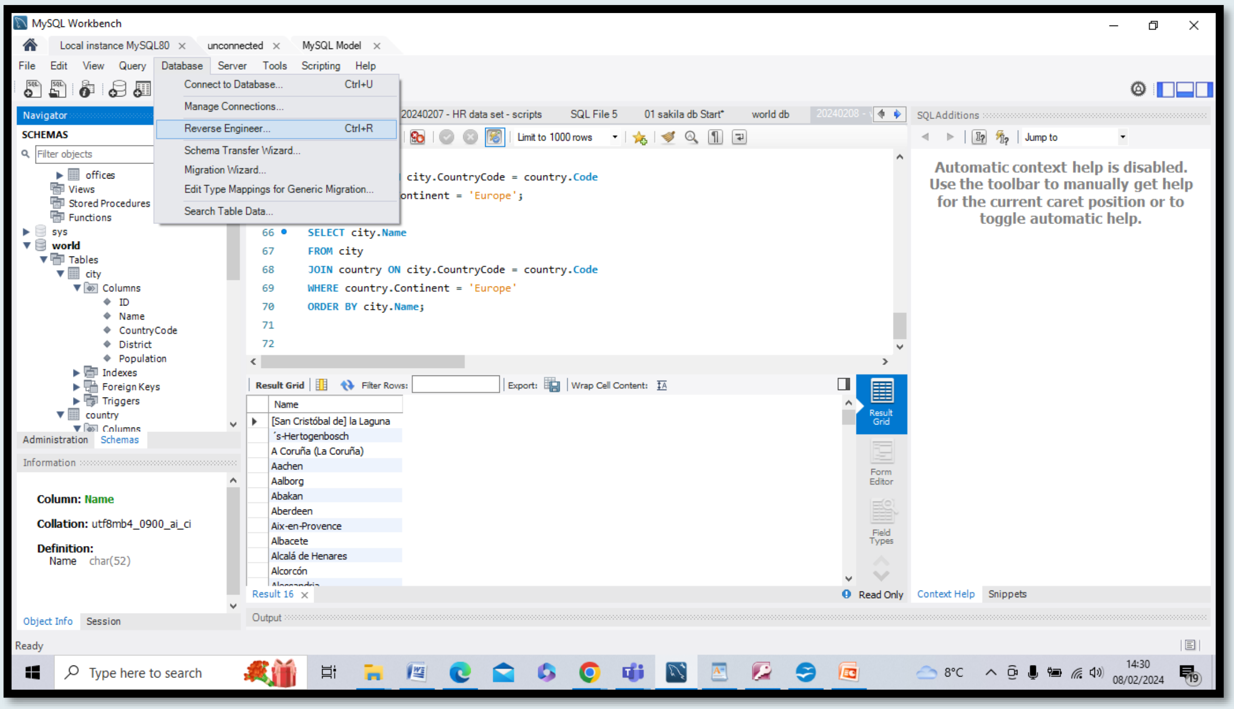
4. **Foreign key in city table:**

- In the 'city' table of the 'world' database, **the 'CountryCode' column serves as a foreign key**. It references the 'Code' column in the 'country' table, establishing a relationship between cities and their respective countries.

5. **Foreign key in countrylanguage table:**

- In the 'countrylanguage' table of the 'world' database, **the 'CountryCode' column serves as a foreign key**. Similar to the 'city' table, it references the 'Code' column in the 'country' table, establishing a relationship between languages and their respective countries.

**An easy way to find this using the EER Diagram is to click on the drop down 'Indexes' which then illuminates the Primary and Foreign Keys active in the Diagram.**



```
city.CountryCode = country.Code
ontinent = 'Europe';

SELECT city.Name
FROM city
JOIN country ON city.CountryCode = country.Code
WHERE country.Continent = 'Europe'
ORDER BY city.Name;
```

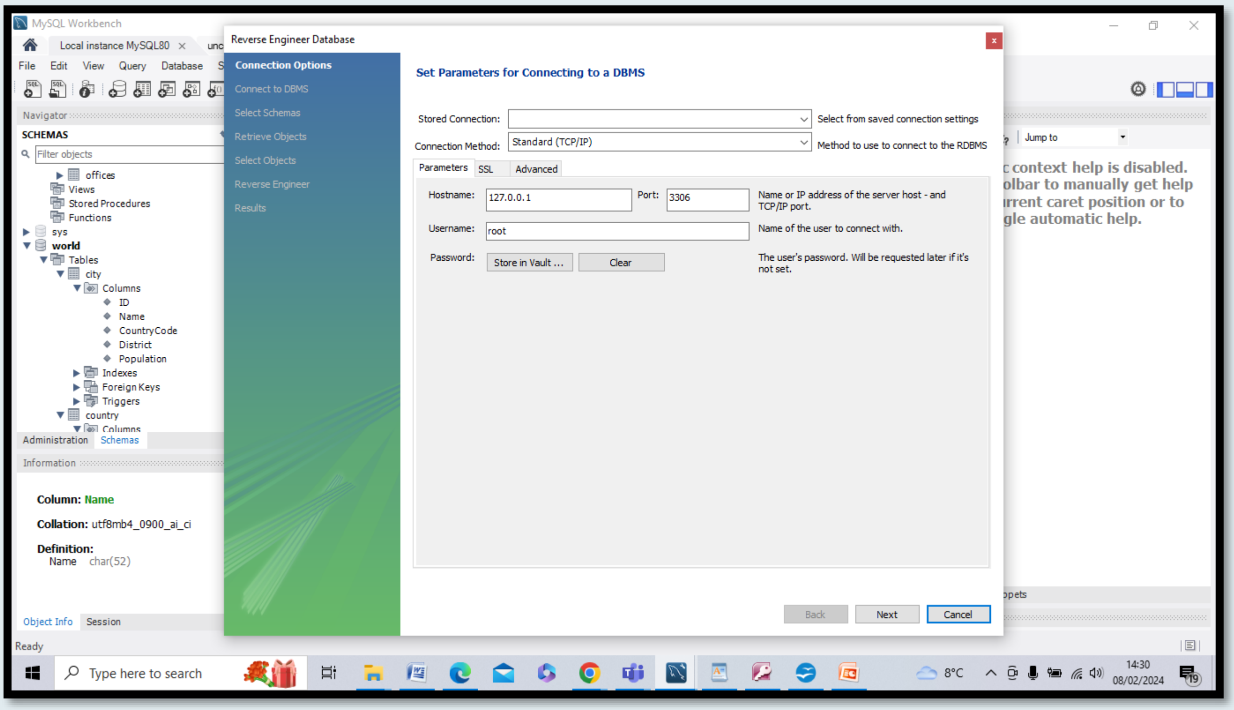
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

**Column:** Name

**Collation:** utf8mb4_0900_ai_ci

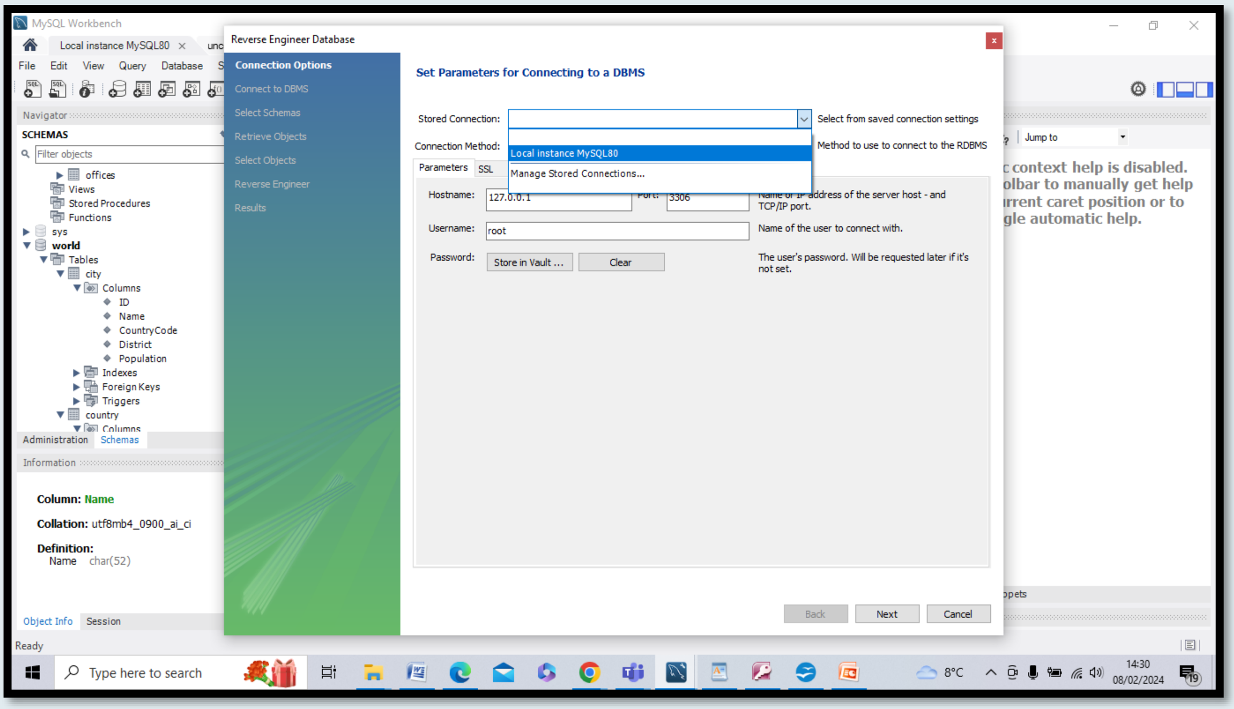
**Definition:**
Name char(52)

| Name |
|---|
| [San Cristóbal de] la Laguna |
| ´s-Hertogenbosch |
| A Coruña (La Coruña) |
| Aachen |
| Aalborg |
| Abakan |
| Aberdeen |
| Aix-en-Provence |
| Albacete |
| Alcalá de Henares |
| Alcorcón |



**Set Parameters for Connecting to a DBMS**

| | | |
|---|---|---|
| Stored Connection: | | Select from saved connection settings |
| Connection Method: | Standard (TCP/IP) | Method to use to connect to the RDBMS |
| Hostname: | 127.0.0.1 Port: 3306 | Name or IP address of the server host - and TCP/IP port. |
| Username: | root | Name of the user to connect with. |
| Password: | | The user's password. Will be requested later if it's not set. |


Reverse Engineer Database

Set Parameters for Connecting to a DBMS

Stored Connection: (dropdown open: Local instance MySQL80 / Manage Stored Connections...) Select from saved connection settings

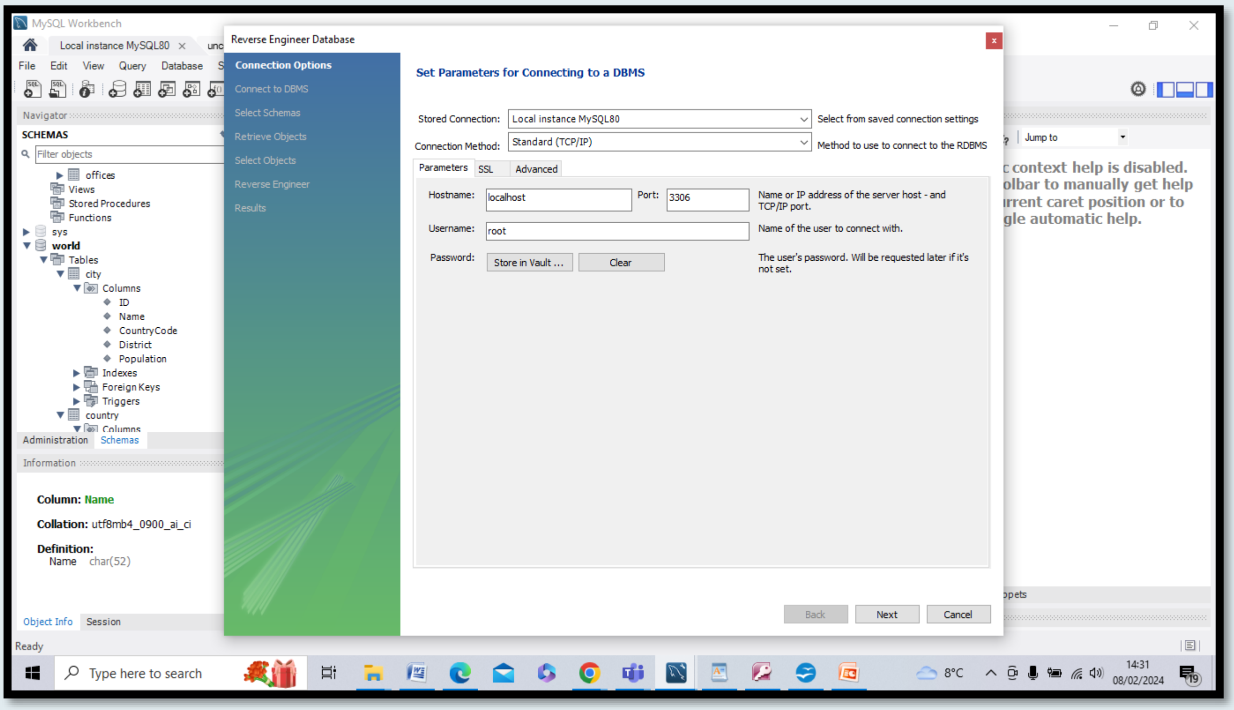
Connection Method: Method to use to connect to the RDBMS

Parameters | SSL

Hostname: 127.0.0.1 Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: root Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.


Reverse Engineer Database

Set Parameters for Connecting to a DBMS

Stored Connection: Local instance MySQL80 Select from saved connection settings

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

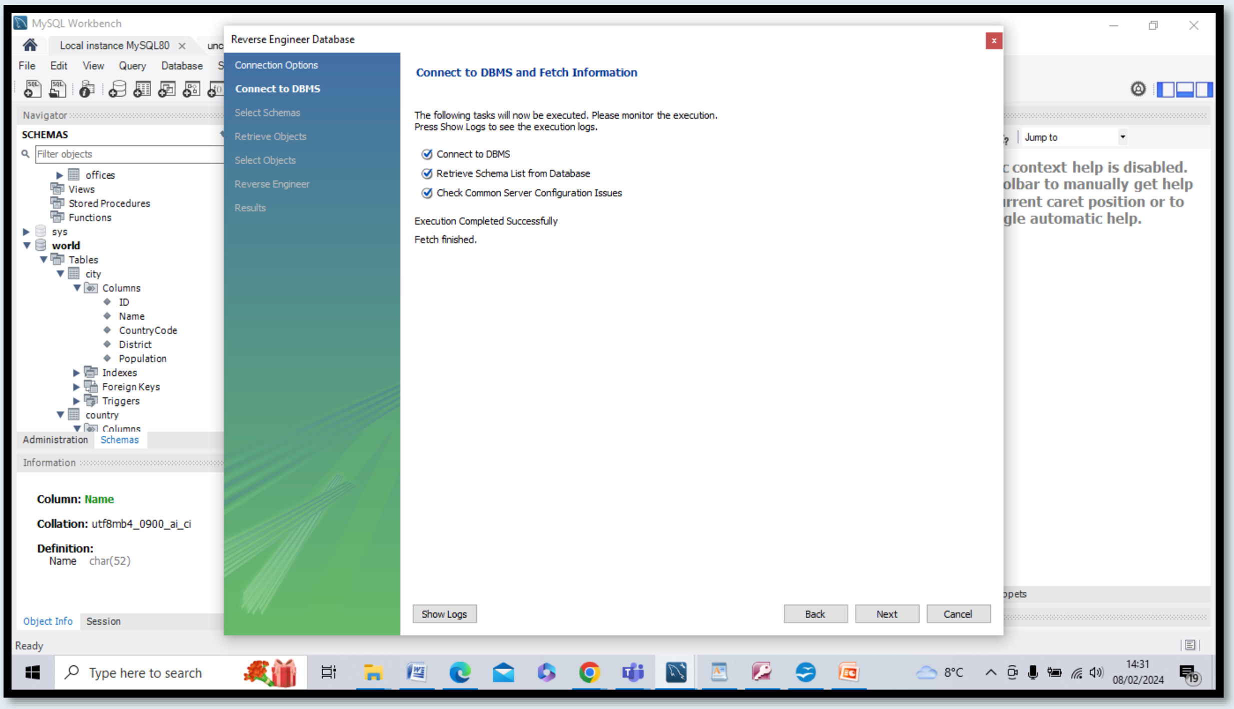
Parameters | SSL | Advanced

Hostname: localhost Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: root Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Press ‘Next’ which produces the following screen:

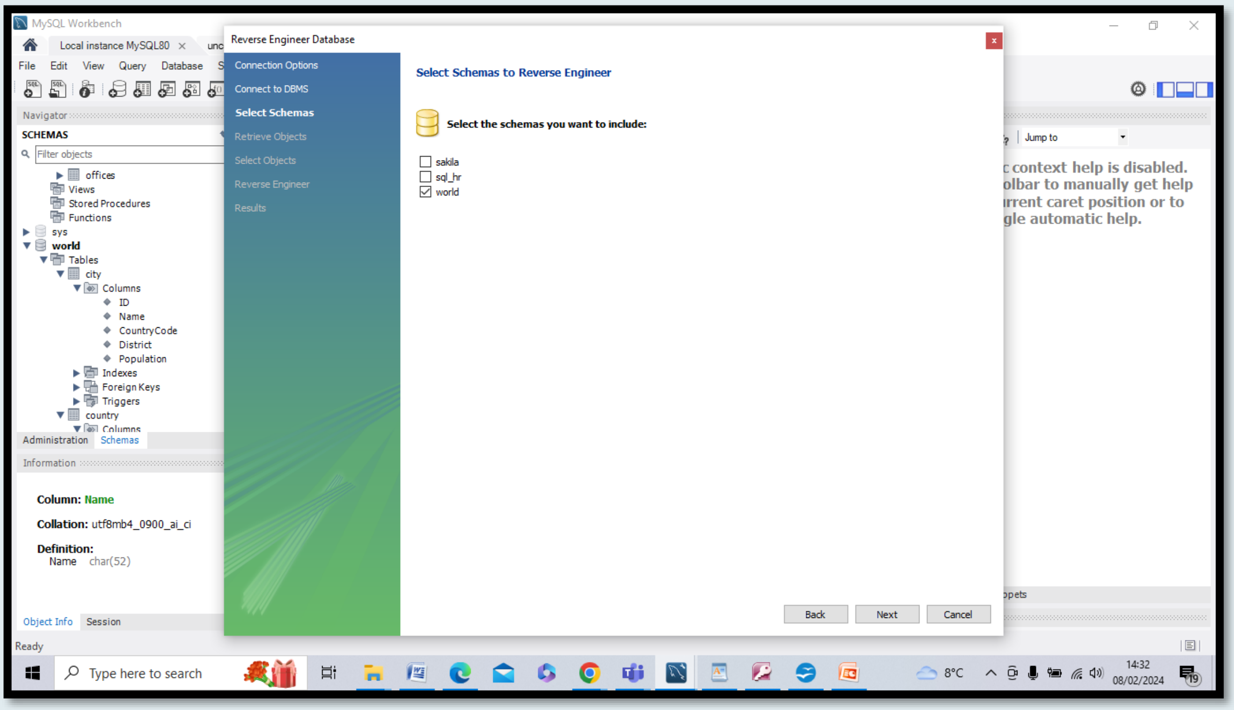

Reverse Engineer Database

Connect to DBMS and Fetch Information

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

✓ Connect to DBMS
✓ Retrieve Schema List from Database
✓ Check Common Server Configuration Issues

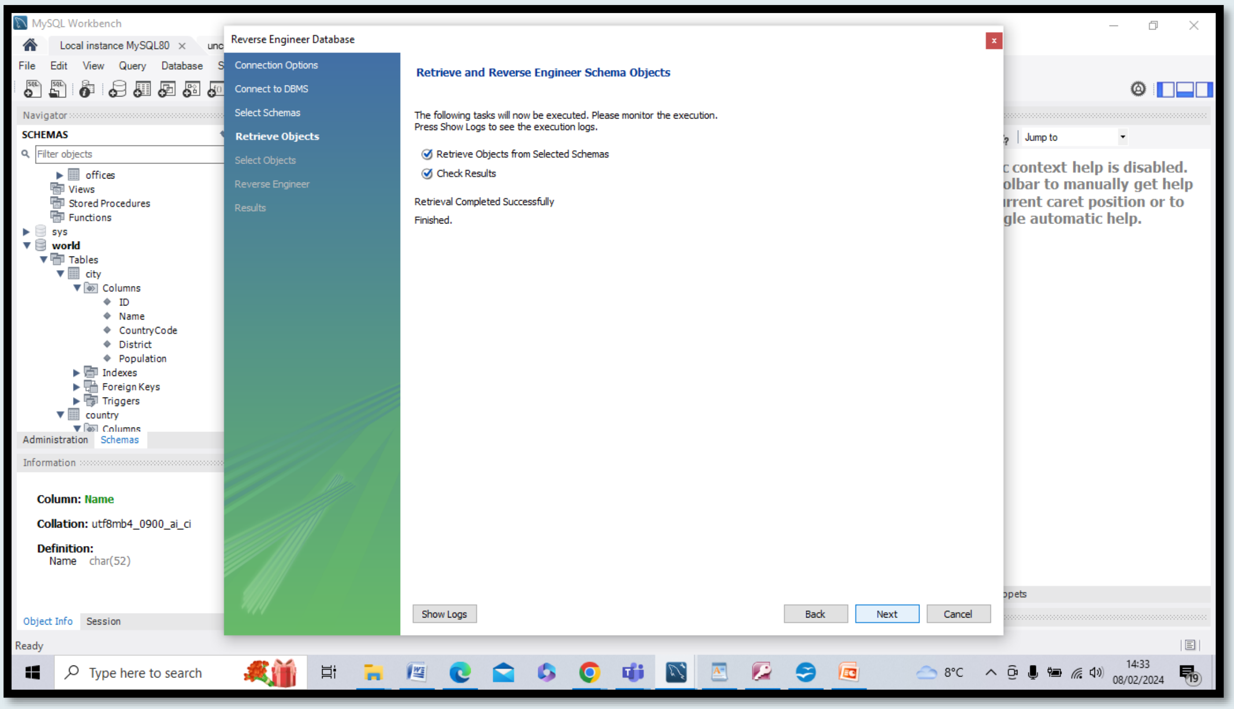
Execution Completed Successfully
Fetch finished.


Reverse Engineer Database

Select Schemas to Reverse Engineer

Select the schemas you want to include:

- [ ] sakila
- [ ] sql_hr
- [x] world

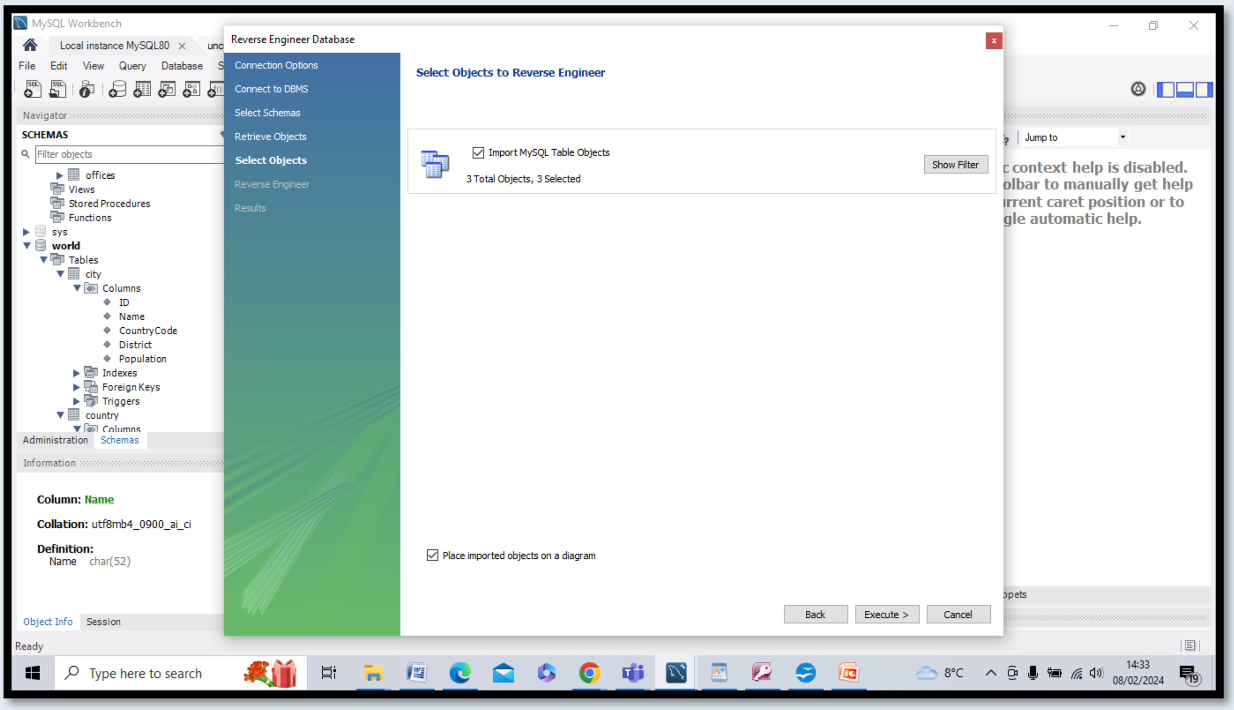

Reverse Engineer Database

**Retrieve and Reverse Engineer Schema Objects**

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

✓ Retrieve Objects from Selected Schemas
✓ Check Results

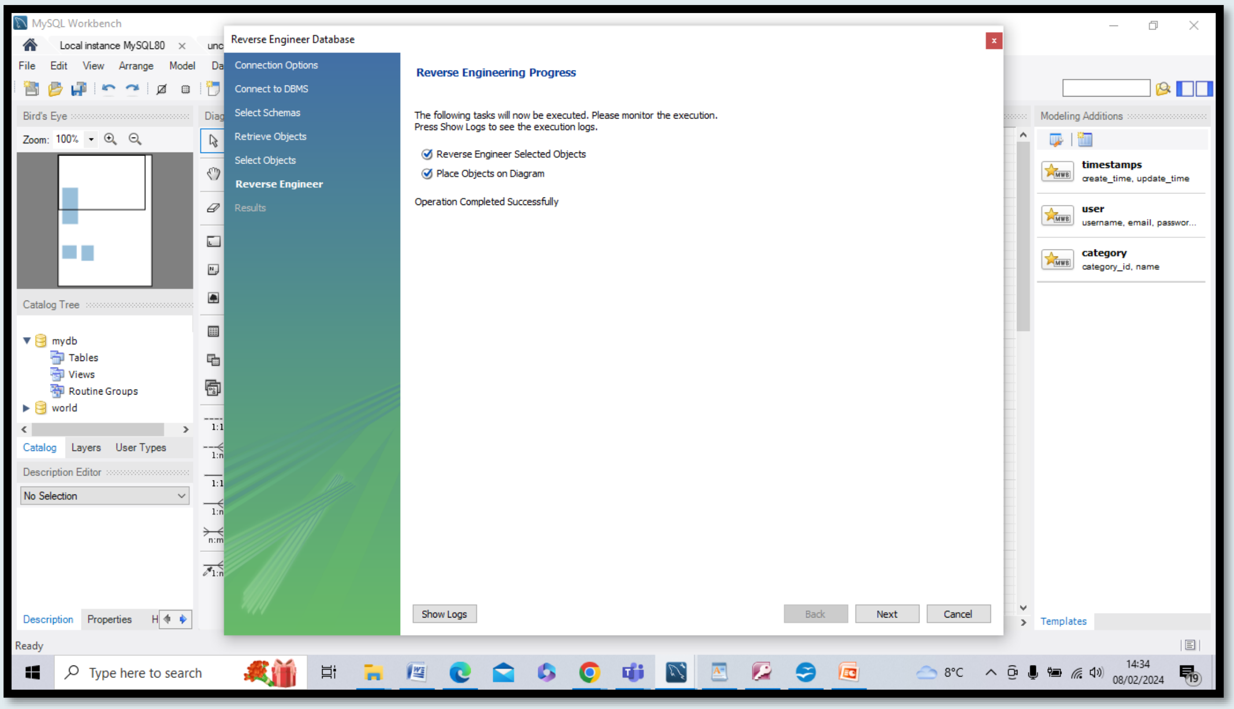
Retrieval Completed Successfully
Finished.


Reverse Engineer Database

**Select Objects to Reverse Engineer**

☑ Import MySQL Table Objects
3 Total Objects, 3 Selected

☑ Place imported objects on a diagram



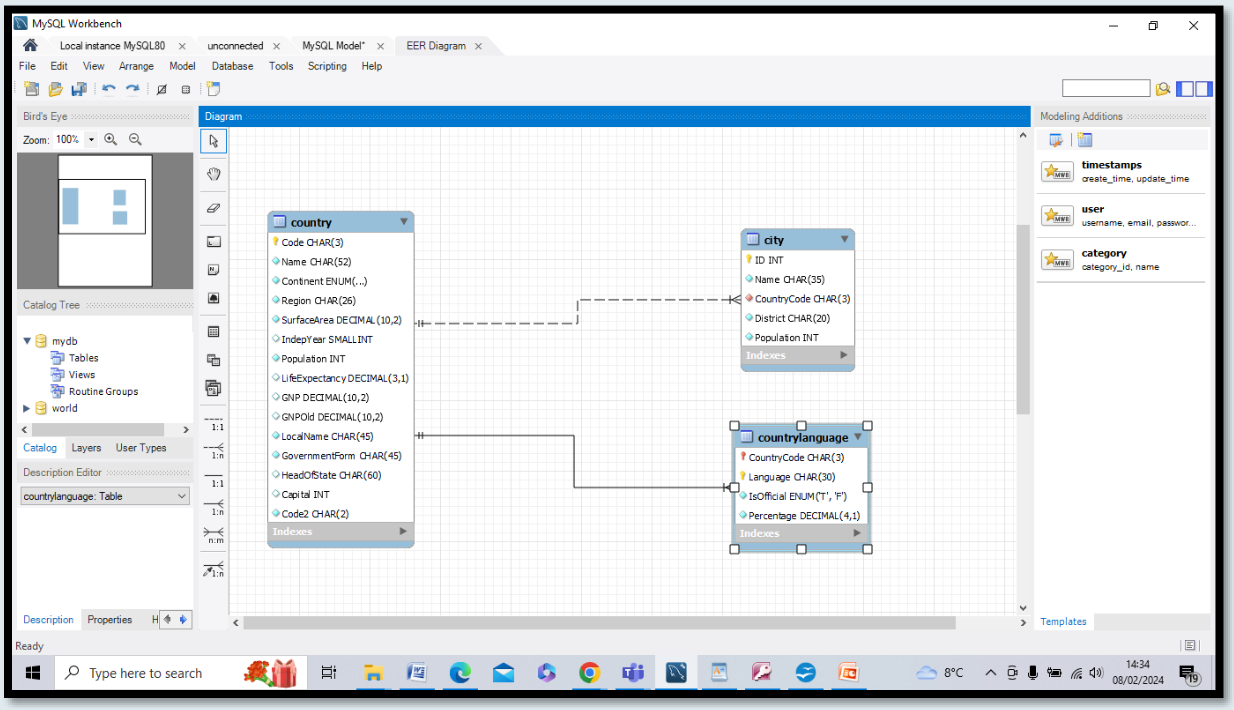
Reverse Engineer Database

**Reverse Engineering Progress**

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

- [✓] Reverse Engineer Selected Objects
- [✓] Place Objects on Diagram

Operation Completed Successfully



MySQL Workbench — EER Diagram

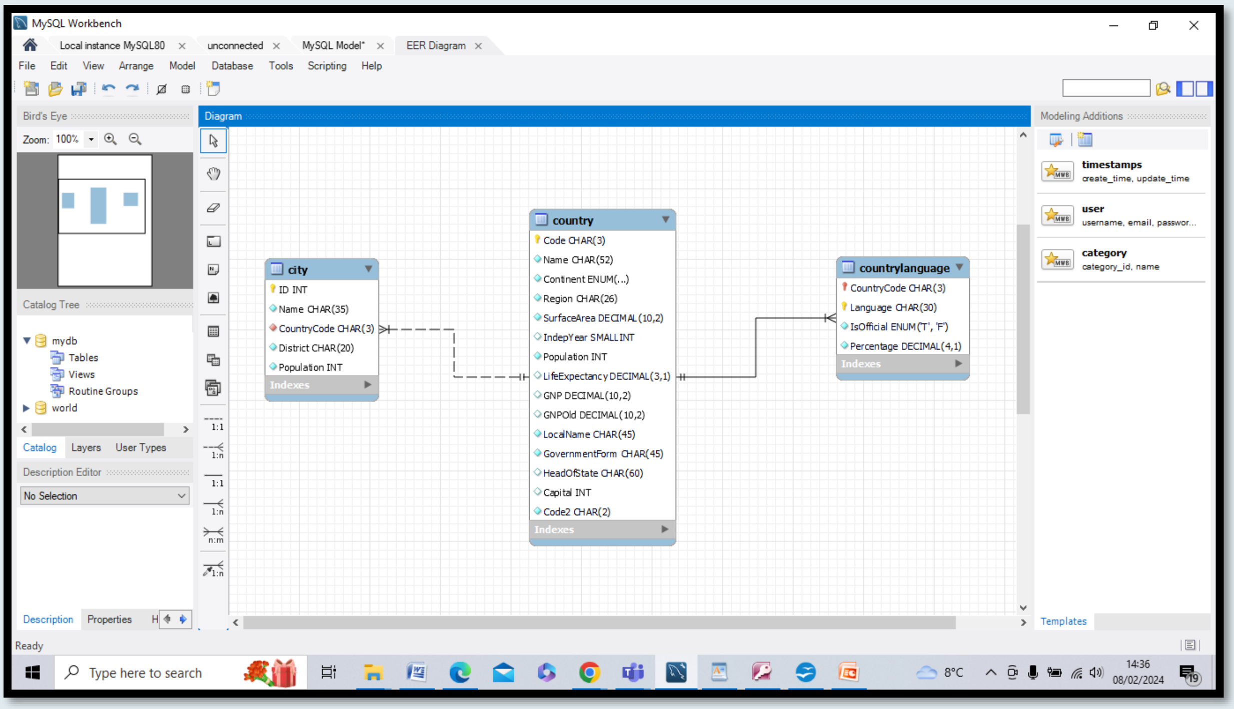
**country**
- Code CHAR(3)
- Name CHAR(52)
- Continent ENUM(...)
- Region CHAR(26)
- SurfaceArea DECIMAL(10,2)
- IndepYear SMALLINT
- Population INT
- LifeExpectancy DECIMAL(3,1)
- GNP DECIMAL(10,2)
- GNPOld DECIMAL(10,2)
- LocalName CHAR(45)
- GovernmentForm CHAR(45)
- HeadOfState CHAR(60)
- Capital INT
- Code2 CHAR(2)

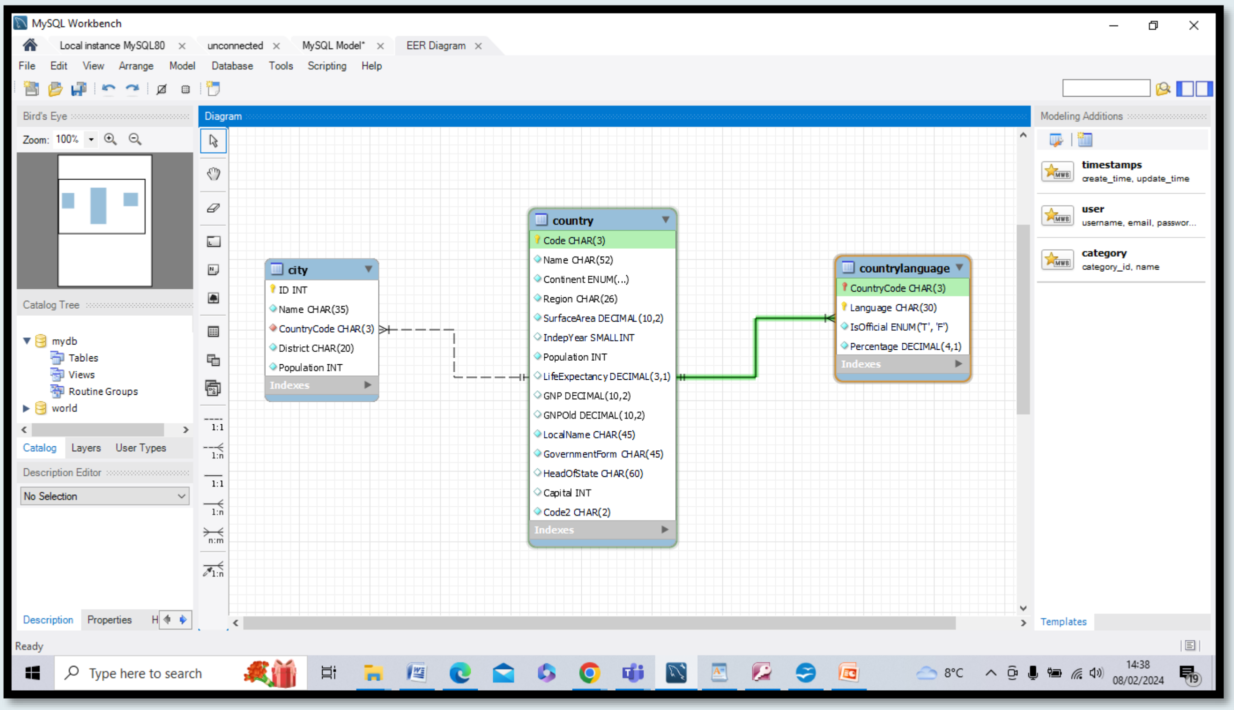
**city**
- ID INT
- Name CHAR(35)
- CountryCode CHAR(3)
- District CHAR(20)
- Population INT

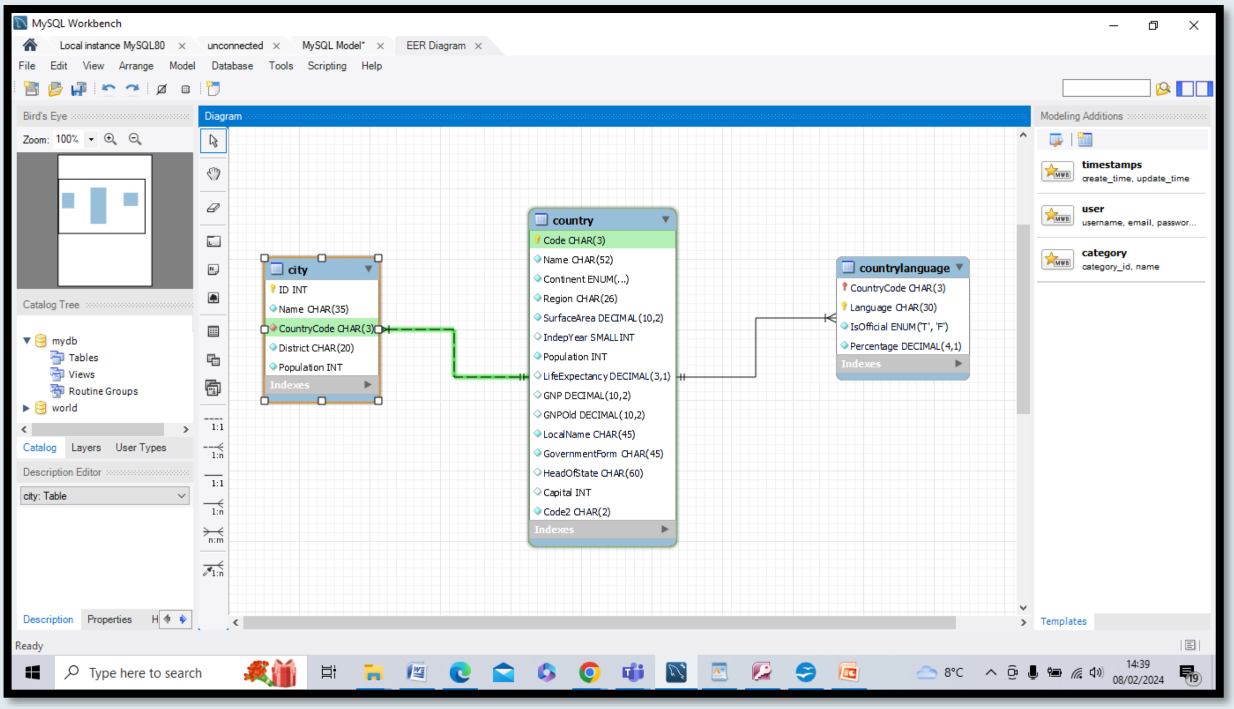
**countrylanguage**
- CountryCode CHAR(3)
- Language CHAR(30)
- IsOfficial ENUM('T', 'F')
- Percentage DECIMAL(4,1)

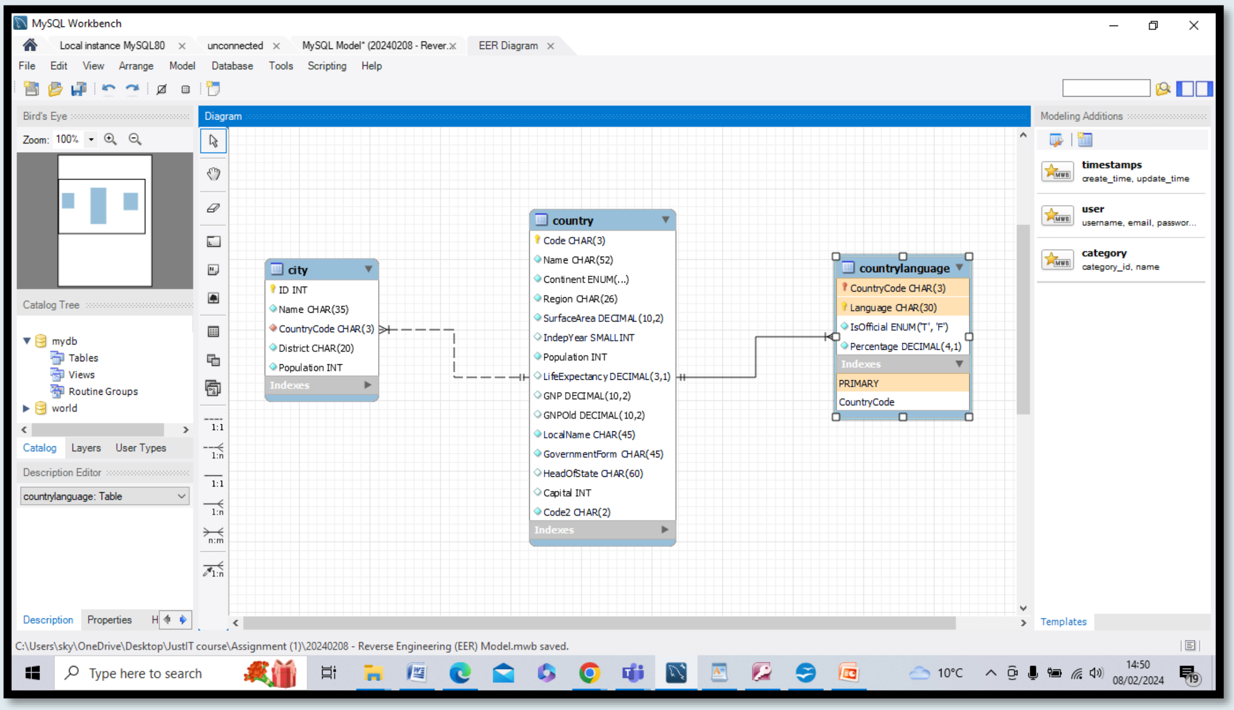
Description Editor: countrylanguage: Table

22

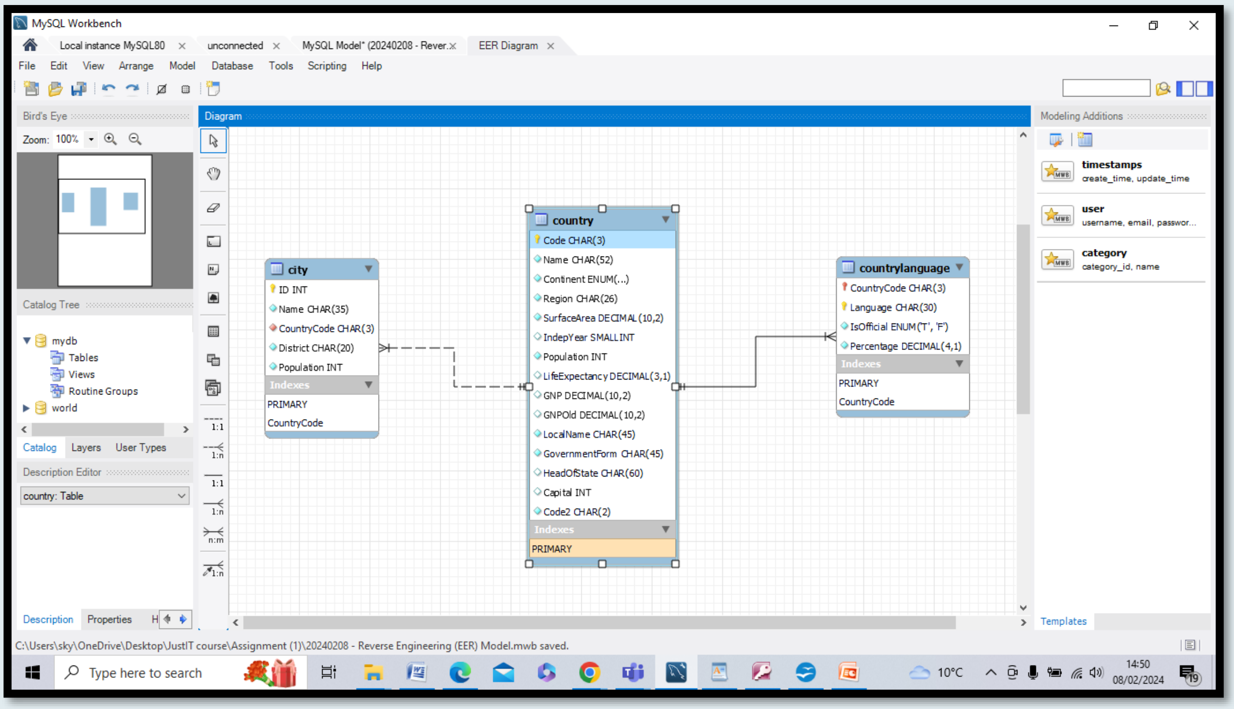

MySQL Workbench
Local instance MySQL80
unconnected
MySQL Model*
EER Diagram
File Edit View Arrange Model Database Tools Scripting Help
Bird's Eye
Zoom: 100%
Diagram
city
ID INT
Name CHAR(35)
CountryCode CHAR(3)
District CHAR(20)
Population INT
Indexes
country
Code CHAR(3)
Name CHAR(52)
Continent ENUM(...)
Region CHAR(26)
SurfaceArea DECIMAL(10,2)
IndepYear SMALLINT
Population INT
LifeExpectancy DECIMAL(3,1)
GNP DECIMAL(10,2)
GNPOld DECIMAL(10,2)
LocalName CHAR(45)
GovernmentForm CHAR(45)
HeadOfState CHAR(60)
Capital INT
Code2 CHAR(2)
Indexes
countrylanguage
CountryCode CHAR(3)
Language CHAR(30)
IsOfficial ENUM('T', 'F')
Percentage DECIMAL(4,1)
Indexes
Modeling Additions
timestamps
create_time, update_time
user
username, email, passwor...
category
category_id, name
Catalog Tree
mydb
Tables
Views
Routine Groups
world
Catalog Layers User Types
Description Editor
No Selection
Description Properties
Templates
Ready
Type here to search
8°C
14:36
08/02/2024


MySQL Workbench
Local instance MySQL80
unconnected
MySQL Model*
EER Diagram
File Edit View Arrange Model Database Tools Scripting Help
Bird's Eye
Zoom: 100%
Diagram
city
ID INT
Name CHAR(35)
CountryCode CHAR(3)
District CHAR(20)
Population INT
Indexes
country
Code CHAR(3)
Name CHAR(52)
Continent ENUM(...)
Region CHAR(26)
SurfaceArea DECIMAL(10,2)
IndepYear SMALLINT
Population INT
LifeExpectancy DECIMAL(3,1)
GNP DECIMAL(10,2)
GNPOld DECIMAL(10,2)
LocalName CHAR(45)
GovernmentForm CHAR(45)
HeadOfState CHAR(60)
Capital INT
Code2 CHAR(2)
Indexes
countrylanguage
CountryCode CHAR(3)
Language CHAR(30)
IsOfficial ENUM('T', 'F')
Percentage DECIMAL(4,1)
Indexes
Modeling Additions
timestamps
create_time, update_time
user
username, email, passwor...
category
category_id, name
Catalog Tree
mydb
Tables
Views
Routine Groups
world
Catalog Layers User Types
Description Editor
No Selection
Description Properties
Templates
Ready
Type here to search
8°C
14:38
08/02/2024





C:\Users\sky\OneDrive\Desktop\JustIT course\Assignment (1)\20240208 - Reverse Engineering (EER) Model.mwb saved.



C:\Users\sky\OneDrive\Desktop\JustIT course\Assignment (1)\20240208 - Reverse Engineering (EER) Model.mwb saved.