**MySQL Assignment Part-2**

**ASSIGNMENT 2**



Table of Contents

[DATABASE MIGRATION 2](#_Toc163939650)

[TASK 1 4](#_Toc163939651)

[TASK 2 4](#_Toc163939652)

[TASK 3 5](#_Toc163939653)

[Task4 6](#_Toc163939654)

[Task5 6](#_Toc163939655)

[Task6 7](#_Toc163939656)

[Task7 8](#_Toc163939657)

[Task8 8](#_Toc163939658)

[Task9 9](#_Toc163939659)

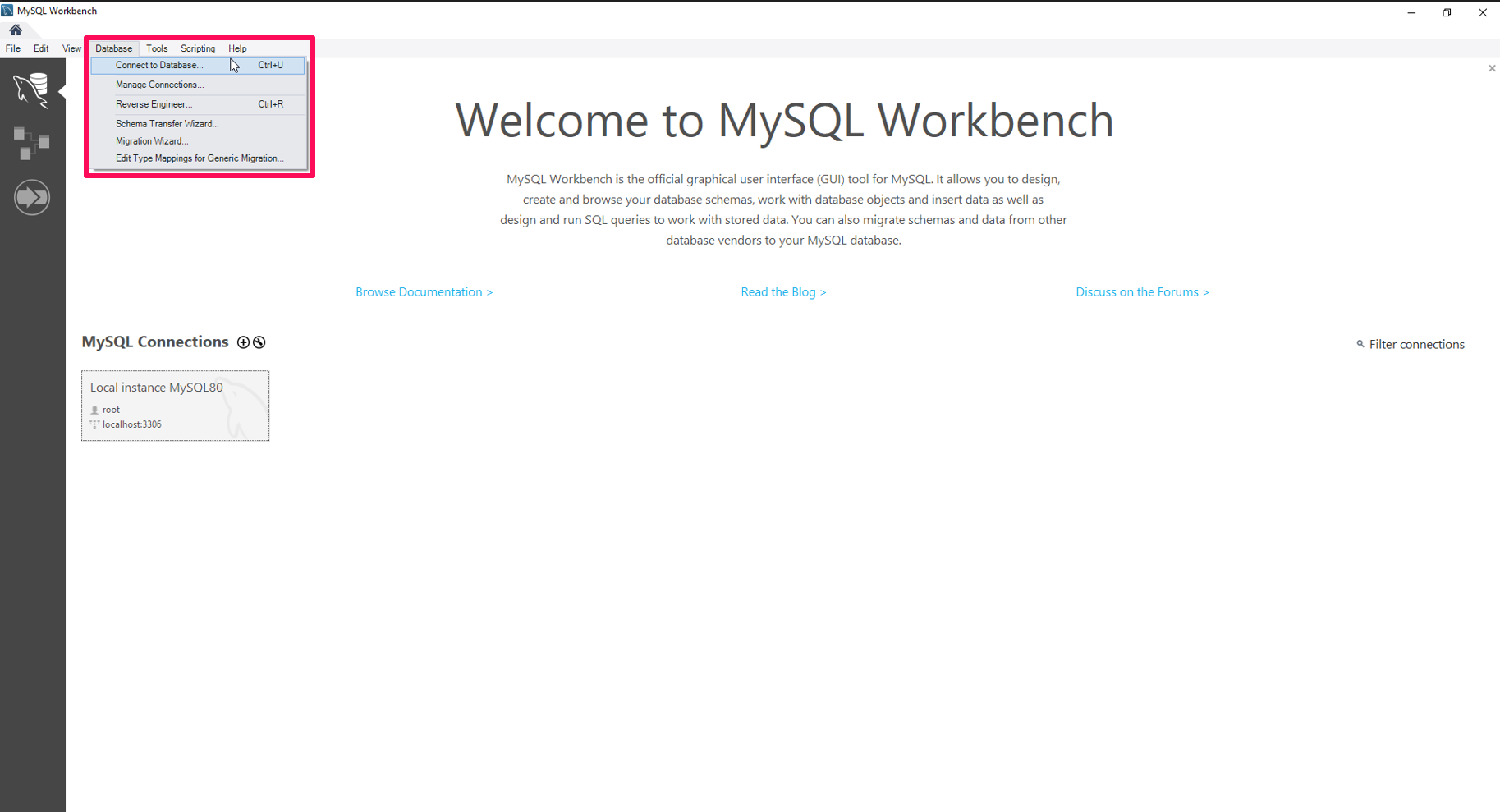
[Bonus Tasks 10](#_Toc163939660)

[EER DIAGRAM 12](#_Toc163939661)

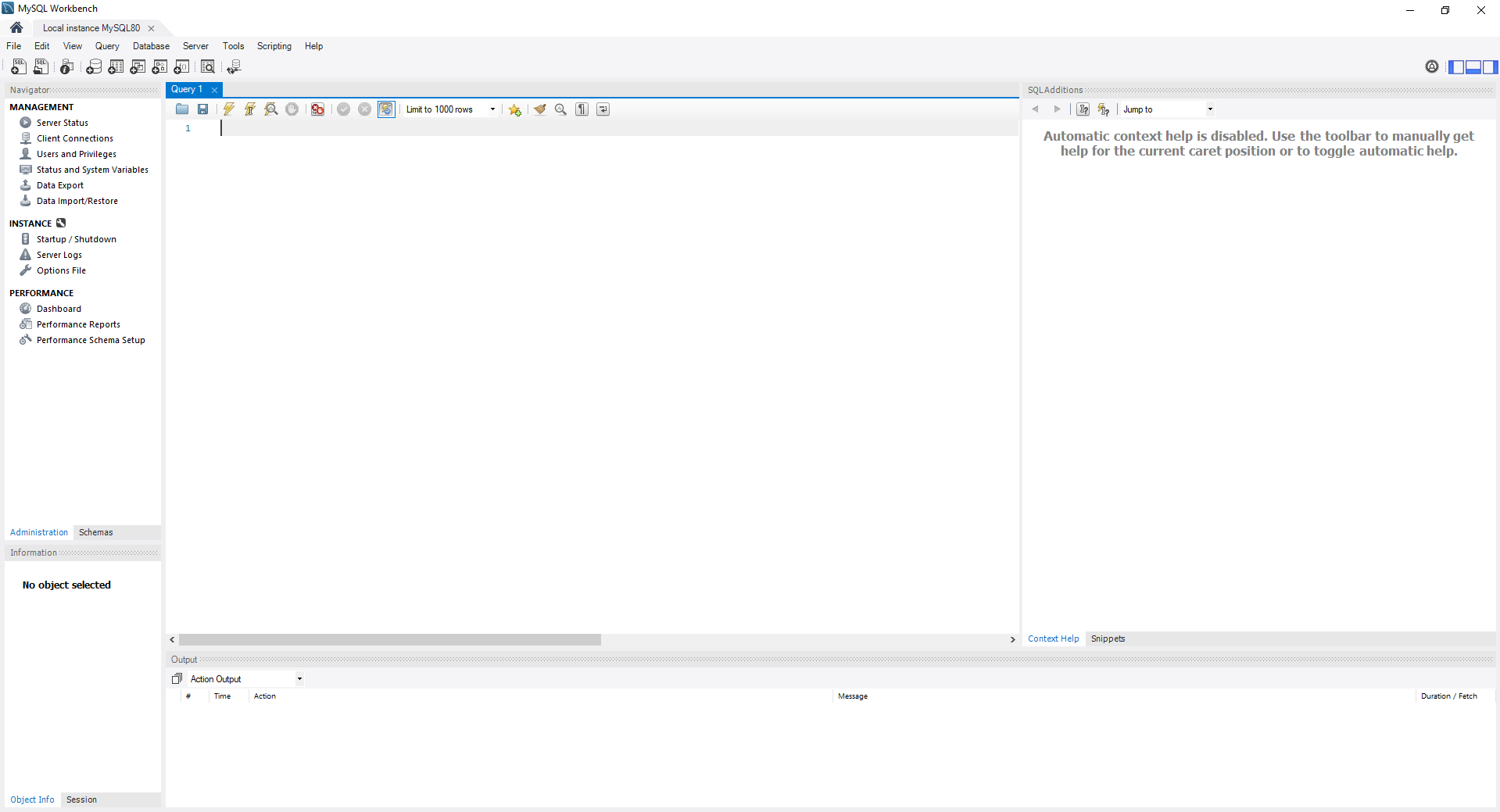
[Task 10 14](#_Toc163939662)

# DATABASE MIGRATION

1. Navigate to MySQL Workbench and connect to the Local Database.

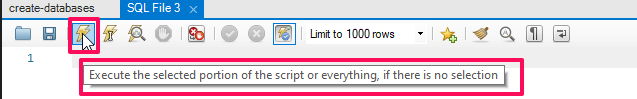


1. Overview of the MySQL Database after setup of the root user password.



1. Open Script File provided in the query editor then execute.





1. Output created after the file has been executed.

A screenshot of a computer

Description automatically generated

1. Navigator view with the Data and Tables created.

A screenshot of a computer

Description automatically generated

# TASK 1

**Task**: Using count, get the number of cities in the USA.

**Solution:** This query will count the number of rows in the cities table where the country column is equal to 'USA', giving you the number of cities in the USA.

**Query:** SELECT COUNT(Countrycode)

FROM city

WHERE Countrycode = 'USA';

Result:

A screenshot of a computer

Description automatically generated

# TASK 2

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

**Solution:** This query will calculate the average life expectancy and total population for people in Argentina by selecting data from the countries table and filtering for the country code 'ARG'.

**Query:** SELECT code, AVG(LifeExpectancy), SUM (population)

FROM country

WHERE code = 'ARG';

A screenshot of a computer

Description automatically generated

# TASK 3

**Task:** Using ORDER BY, LIMIT, what country has the highest life expectancy?

**Solution:** This query will order the countries by life expectancy in descending order and then limit the result to only one row, which corresponds to the country with the highest life expectancy.

**Query:** SELECT name, LifeExpectancy

FROM country

ORDER BY LifeExpectancy DESC

LIMIT 1;

Result:

A screenshot of a computer

Description automatically generated

# Task4

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

**Solution:** This query selects all columns from the city table where the Name column starts with the letter 'F’ and limits the result to 25 rows.

**Query:** SELECT \* FROM city

WHERE Name LIKE 'F%'

LIMIT 25;

Result:

A screenshot of a computer

Description automatically generated

# Task5

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

**Solution:** This query selects specific columns (Id, Name, Population) from the city table and restricts the result to the first 10 rows.

**Query:** SELECT Id, Name, Population

FROM city

LIMIT 10;

Result:

A screenshot of a computer

Description automatically generated

# Task6

**Task:** Create a SQL statement to find only those cities from city table whose population is larger than 2000000.

**Solution:** This query selects all columns from the city table and filters the result to only include cities where the Population column is larger than 2,000,000.

**Query:** SELECT \* FROM city

WHERE Population > 2000000;

Result:A screenshot of a computer

Description automatically generated

# Task7

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

**Solution:** This query selects the Name column from the city table and filters the result to only include city names that begin with the prefix "Be" using the LIKE operator with the % wildcard character, which matches any sequence of characters.

**Query:** SELECT Name FROM city

WHERE Name LIKE 'Be%';

Result:A screenshot of a computer

Description automatically generated

# Task8

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

**Solution:** This query selects all columns (\*) from the city table and filters the result to only include cities where the Population column falls within the range of 500,000 to 1,000,000 using the BETWEEN operator. The BETWEEN operator is inclusive of both endpoints, so it will include cities with a population of exactly 500,000 or 1,000,000 in the result set.

**Query:** SELECT \* FROM city

WHERE Population BETWEEN 500000 AND 1000000;

Result:

A screenshot of a computer

Description automatically generated

# Task9

**Task:** Create a SQL statement to find a city with the lowest population in the city table.

**Solution:** This query selects all columns (\*) from the city table, orders the result by population in ascending order (ASC), which will put the cities with the lowest population at the top, and limits the result to only one row using LIMIT 1. Therefore, it will return the city with the lowest population.

**Query:** SELECT \* FROM city

ORDER BY Population ASC

LIMIT 1;

Result:

A screenshot of a computer

Description automatically generated

# Bonus Tasks

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

**Solution**: This query retrieves the name of the capital city (city.Name) by joining the city and country tables. It matches the CountryCode column from the city table with the Code column from the country table and ensures that the city ID (city.ID) matches the capital ID (country.Capital) in the country table for the specified country code ('ESP').

**Query**: SELECT city.Name AS capital

FROM city

JOIN

country ON city.CountryCode = country.Code

WHERE

country.Code = 'ESP' AND country.Capital = city.ID;

Result:

A screenshot of a computer

Description automatically generated

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

**Solution**: This query retrieves distinct languages (cl.Language) spoken in the Caribbean region by joining the countrylanguage table with the country table. It matches the CountryCode column from the countrylanguage table with the Code column from the country table and filters for countries in the Caribbean region (c.Region = 'Caribbean').

**Query**: SELECT DISTINCT cl.Language FROM CountryLanguage cl

JOIN

Country c ON cl.CountryCode = c.Code

WHERE

c.Region = 'Caribbean';

Result:

A screenshot of a computer

Description automatically generated

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

**Solution**: This query retrieves the names of all cities (city.Name) from the Europe continent by joining the city and country tables. It matches the CountryCode column from the city table with the Code column from the country table and filters for countries in the Europe continent (country.Continent = 'Europe').

**Query**: SELECT city.Name AS city\_name

FROM city

JOIN

country ON city.CountryCode = country.Code

WHERE

country.Continent = 'Europe';

Result:

A screenshot of a computer

Description automatically generated

# EER DIAGRAM

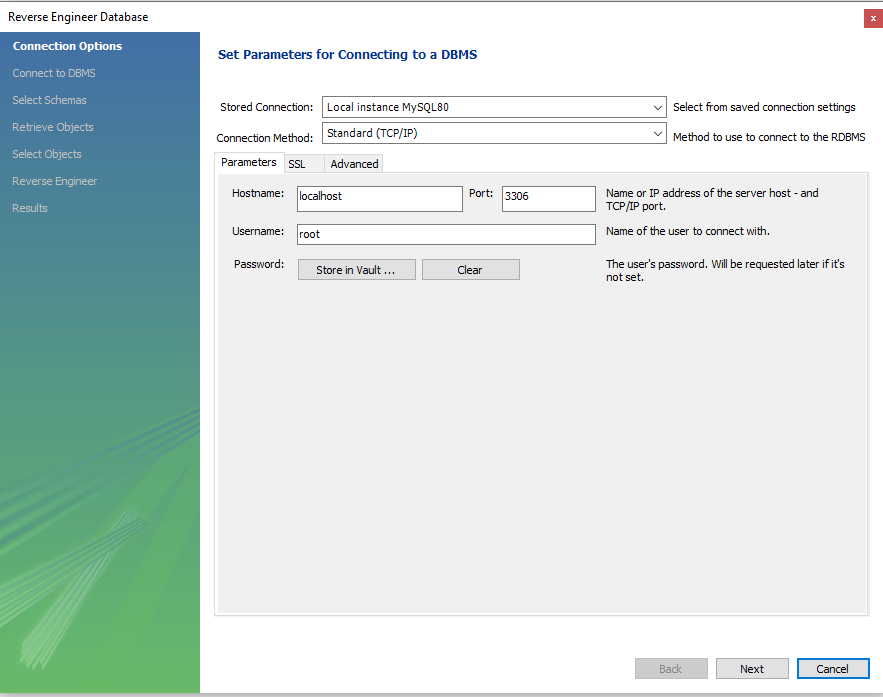
EER (Enhanced Entity-Relationship) diagrams are a type of data modeling tool used in software engineering to design and document complex databases. EER diagrams are an extension of the traditional Entity-Relationship (ER) model, which is used to define the relationships between entities (such as people, places, and things) in a database.

1. Create EER Diagram – navigate to Database tab, choose Reverse Engineer option or use ctrl + R shortcut on Windows devices or cmd + R on Mac OS.

A screenshot of a computer

Description automatically generated

1. Navigate through setup wizard.



1. We want to include the store schema.

A screenshot of a computer

Description automatically generated

1. EER Diagram Result:

A screenshot of a computer

Description automatically generated

# Task 10

1. The Primary Key in the Country Table is ‘Code’
2. The Primary Key in the City Table is ‘ID’
3. The Primary Key in countrylanguage table is ‘CountryCode & Language’
4. The Foreign Key in the City Table is ‘CountryCode’
5. The Foreign Key in the Countrylanguage Table is ‘ CountryCode’