|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| Databases and SQL  Assignment 2 |
| |  |  |  | | --- | --- | --- | | Steven Laurie | 10/24/23 | Just IT - Data Analyst Bootcamp | |

Contents

[**Assignment Briefing** 2](#_Toc148801861)

[**Task 1** 2](#_Toc148801862)

[**Task 2** 3](#_Toc148801863)

[**SQL Tasks** 4](#_Toc148801864)

[**Task 3** 5](#_Toc148801865)

[**Task 4** 5](#_Toc148801866)

[**Task 5** 6](#_Toc148801867)

[**Task 6** 6](#_Toc148801868)

[**Task 7** 7](#_Toc148801869)

[**Task 8** 7](#_Toc148801870)

[**Task 9** 8](#_Toc148801871)

[**Task 10** 8](#_Toc148801872)

[**Task 11** 9](#_Toc148801873)

[**Task 12** 9](#_Toc148801874)

[**Task 13** 10](#_Toc148801875)

[**Task 14** 13](#_Toc148801876)

# **Assignment Briefing**

In my second assignment I will be carrying out various tasks using the knowledge I have acquired from my sessions and demonstrating the technical skills I have learned within MySQL Workbench software using SQL programming language. Throughout this assignment I should be able to present that I understand the following areas of focus:

* Creating a database
* Writing SQL query language and execution
* SQL query debugging
* Creating an EER Diagram
* Understanding relationships
* Identifying primary and foreign key

# **Task 1**

*“List the different types of relationships in relational databases and provide examples.”*

Relational databases are a type of database which consist of a dataset organised into a table-like form where pre-defined relationships are set for the data to interact with one another.

In these relational databases there are three main types of relationships you can set, they are.

* One-to-one relationships
* One-to-many relationships
* Many-to-many relationships

One-to-one relationships consists of a relationship where one entity is set to uniquely relate with another entity. Examples of this type of relationship in databases are ID numbers. Each person will have a designated ID number that is unique to them, you cannot have another person with the same ID Number.

One-to-many relationships are slightly different in which one entity can be associated with many other entities. This is shown in social media where a user can have multiple different posts, but these posts will always belong to the same user.

Many-to-many relationships are designed with the idea where many entities can be associated with many other entities. An example of where this can be seen in databases is between students and classes. In a school students can be registered on many classes, and these classes can be taken by many students.

# **Task 2**

*‘What is Normalization and why is it important to databases development?”*

Database normalization is the process of organizing data in a database to minimize the likelihood of data redundancy (duplicated data) and improve data integrity. This can involve removing any duplications, errors, or anomalies within a database.

Database normalization is an important and essential part of database management to businesses that use a large amount of data, if businesses do not use database normalization, something as little as the deletion of one field can cause a long list of errors throughout different tables within a database.

There are many advantages to using normalization in your databases, here are a few:

* **Reduced Redundancy** – Reduction of data will improve analysis times by decreasing the amount of data you need to search through to find something specific therefore increasing processing times and shortening file sizes.
* **Improved Database Organisation** – Once normalising a database your data will be constructed and arranged so that it is accessible for all different departments to access. This means that each department will only have access to the information they need and no excess which provides better security and reduces any duplicated data. This will also make it easier to update the database as you will only have to review one dataset rather than many individual ones across departments.
* **Increased Security –** Your database will be more secure as you organise it and transform raw data into more accurate data that is easy to understand to humans through formatting
* **Faster Workflow** – Normalization can improve the workflow of your business through many ways, duplicated values or null values can be filtered out to make data more accurate, this in turn should make queries easier to complete.

# **SQL Tasks**

In this section I will be completing various tasks in MySQL workbench using a dataset known as “World” containing information on many countries, cities, and languages spoken within them.

I will be completing numerous duties such as executing queries, debugging queries, creating, and identifying relationships (primary keys/foreign keys) in EER diagrams. I will be using various statements, clauses, keywords, operators, and functions throughout these tasks. I will list these detailed below so they can be used as a guide where needed.

* **SELECT**: Used to select data from a database.
* **FROM:** Used to specify where to reference to.
* **WHERE:** Used to filter through data.
* **AS:** Used to rename a column/table.
* **LIKE:** Used within where clauses to search for a specific pattern within a column.
* **GROUP BY:** Used to group a set of results.
* **ORDER BY:** Used to order results in an Ascending or Descending order.
* **AND:** Used to filter records based on more than one condition.
* **MIN/MAX:** Used to return a minimum or maximum value.
* **COUNT:** Used to count the number of rows for a specified condition.
* **AVG:** Used to display the average value of a column.
* **BETWEEN:** Used to return values within a certain range.
* **LIMIT:** Used to return a specified number of values.
* **INNER JOIN:** Used to combine records from two tables where there are matching values.

## **Task 3**

*“Using count, get the number of cities in the USA”.*

A screenshot of a computer

Description automatically generated

## **Task 4**

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

A screenshot of a computer

Description automatically generated

## **Task 5**

“Using ORDER BY, LIMIT what country has the highest life expectancy”.

A screenshot of a computer

Description automatically generated

## **Task 6**

“Select 25 cities around the world that start with the letter ‘F’ in a single SQL query.”

A screenshot of a computer

Description automatically generated

## **Task 7**

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

A screenshot of a computer

Description automatically generated

## **Task 8**

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

A screenshot of a computer

Description automatically generated

## **Task 9**

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

A screenshot of a computer

Description automatically generated

## **Task 10**

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

A screenshot of a computer

Description automatically generated

## **Task 11**

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

A screenshot of a computer

Description automatically generated

## **Task 12**

“Create a SQL statement to show the population of Switzerland and all the languages spoken there”.

A screenshot of a computer

Description automatically generated

## **Task 13**

“Create an EER Diagram (Entity Relationship Diagram)”

An EER Diagram is a form of visualisation to showcase how different entities (eg. A place, person, object) are structured and the different relationships they have together. In the following steps I will create an EER Diagram.

* First I will navigate to the Database tab and select “Reverse Engineer”.

A screenshot of a computer

Description automatically generated

* Next leave all the fields as they are and select Next.

A screenshot of a login form

Description automatically generated

* To follow again click next

A screenshot of a computer

Description automatically generated

* In the next slide make sure the “world” schema is select and then click next

A screenshot of a computer

Description automatically generated

* To continue just click next.

A screenshot of a computer

Description automatically generated

* Make sure “Import MySQL Table Objects” is selected, for this database it should show three total objects as there are three tables. Leave “Place imported objects on a diagram” selected and then click Execute.

A screenshot of a computer

Description automatically generated

* From here select next

A screenshot of a computer

Description automatically generated

* And for the last step select finish

A screenshot of a computer

Description automatically generated

* It should bring you to a new tab known as “EER Diagram” which will look like this:

A screenshot of a computer

Description automatically generated

## **A screenshot of a computer Description automatically generatedTask 14**

**“Identify the primary key in country table”.**

1. Using the index, we can see that “Code” is the primary key in the country table which is also shown by the yellow key symbol beside.

**“Identify the primary key in city table”.**

1. A screenshot of a computer

   Description automatically generatedAgain, using the index, we can see that “ID” is the primary key in the city table, also shown by the yellow key beside the field.

A screenshot of a computer

Description automatically generated

**“Identify the primary key in countrylanguage table”.**

1. Within the countrylanguage table both “CountryCode” and “Language” are showing as primary keys. This is because they form to create a composite primary key. This composite key can be used to identify a proportion of people from a specific country.

**“Identify the foreign key in city table”.**

1. Hovering over the “CountryCode” index we can see that “CountryCode” is the foreign key in the city table. Furthermore, we can follow the relationship line connecting the city and country table to see that CountryCode is referencing to the Primary key “Code” in the country table.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**“Identify the foreign key in countrylanguage table”.**

1. A screenshot of a computer

   Description automatically generatedAs known from step 2 I identified that “CountryCode” is a composite key. This key can be used as both a Primary and Foreign Key. Using the index, we can also see that “CountryCode” is highlighted as the foreign key. We can see what exactly “CountyCode” is referencing to by following the relationship arrow in the EER Diagram which is “Code” in the country entity.

A screenshot of a computer

Description automatically generated

# **Reflection**

To recap, in this document I covered:

* The different types of relationships within relational databases
* What normalization is and why it is important to databases.
* Multiple SQL queries using different keywords, functions etc.
* How to create an EER Diagram

I have learned a lot throughout this assignment, I touched up my understanding of different SQL statements, keywords etc. used when creating queries. I have found that when writing my code, I would encounter some errors, but I would be able to quickly amend the problem due to the better understanding I have gained of reading error codes to figure out where I have made a mistake in my code. I have built a better understanding of composite keys and how they can be used within MySQL. I feel that this assignment has been very valuable to me, and I am excited to further expand my knowledge within MySQL Workbench software and SQL language.

# **Reference Code**

Here I will post a transcript of the code I would use within MySQL Workbench if you would like to inspect it:

USE world;

#Task 3

SELECT COUNT(\*) AS 'Cities in USA' FROM city

Where Countrycode = 'USA';

#Task 4

SELECT Name AS 'Country', Population AS 'Population', LifeExpectancy AS 'Life Expectancy'

FROM country

WHERE Name = 'Argentina';

#Task 5

SELECT Name AS 'Country', LifeExpectancy AS 'Life Expectancy'

FROM country

ORDER BY LifeExpectancy DESC

LIMIT 1;

#Task 6

SELECT \* FROM city

WHERE Name LIKE 'F%'

LIMIT 25;

#Task 7

SELECT ID, Name, Population

FROM city

LIMIT 10;

#Task 8

SELECT Name AS 'City', Population AS 'Population' FROM city

WHERE Population > 2000000;

#Task 9

SELECT \* FROM city

WHERE Name LIKE 'Be%';

#Task 10

SELECT Name as 'City', Population as 'Population' FROM city

WHERE Population BETWEEN 500000 AND 1000000;

#Task 11

SELECT Name AS 'City', MIN(Population) AS 'Lowest Population'

FROM city

GROUP BY Name, Population

ORDER BY Population

LIMIT 1;

#Task 12

SELECT CountryCode, Name, Population, Language

FROM countrylanguage

INNER JOIN country

WHERE CountryCode = 'CHE' AND Name = 'Switzerland';