Outline

Tutorial 4 comprises two parts. In Part I, we discuss exercises from Chapters 5 and 6 from the book by Garcia-Molina, Ullman, and Widom (2014) and some further queries using MySQL Workbench. In Part II, we will connect to your MySQL Server using Tableau, an interactive tool for data visualization. While working on Part II, we will be joined by Ignas Krikštaponis, who has been much involved in crafting that part of the tutorial.

Important installation preparations

Detailed installation instructions can be found in the Syllabus. Please make sure you have installed **MySQL Server** and **MySQL Workbench**, together with the **"sakila" database**. Also, make sure the **MySQL ODBC connector** is installed. Moreover, please make sure you have **Tableau** installed, and activated using the **product key** in the Syllabus.

Finally, make sure the **"world" database**, which is used in Part II of this tutorial, is present on your MySQL Server. If the "world" database is not yet installed, please go over the following steps:

1. download `E_EOR2_DBFA.TUT4.world.sql`,
   available on Canvas, under E_EOR2_DBFA $>$ Files $>$ Tutorials,
2. open MySQL Workbench,
3. open a connection to your server using the workbench,
4. in the local instance that is then openend, go to File $>$ Run SQL Script,
5. select the `E_EOR2_DBFA.TUT4.world.sql` file, and
6. click Run in the wizard.

Part I

Exercises from Chapter 5

Please go over these exercises **before** the tutorial.

- Exercise 2.1

Exercises from Chapter 6

Please go over these exercises **before** the tutorial.

- Exercise 1.1
- Exercise 1.2

- Exercise 2.1
- Exercise 3.1
- Exercise 3.6

Finally, revisit Exercises 3.1 and 4.1 from Chapter 2, discussed in Tutorial 1. Here, pay special attention to the SQL code in `E_EOR2_DBFA.TUT1.sql`, found on Canvas, under E_EOR2_DBFA > Files > Tutorials.

Exercises for "sakila" using MySQL Workbench

You will work on the following exercises in break-out rooms during the tutorial itself. So you do not need to prepare these exercises prior to the tutorial. Please make sure, though, that you have followed all installation instructions at the beginning of this document **before** the tutorial. I will be moving between break-out rooms, to see if there are any questions or if I can give some assitance.

In these exercises, all you need to do is write the right query, and test if it works as expected. There is no need to store all results, especially as some results may contain hundreds of rows.

Hint 1. When writing these queries, use the SCHEMAS panel on the left-hand side of MySQL Workbench to navigate through the "sakila" database. By navigating through the database, you will figure out e.g. which tables (i.e. relations) there are and what columns (i.e. attributes) those tables have.

Hint 2. You need the following tables for these exercises: film, film_actor, actor, film_category, category.

Hint 3. The right order of the keywords in a basic SQL statement is: `SELECT`, `FROM`, `WHERE`, `GROUP BY`, `HAVING`, `ORDER BY`. Also, don't forget the semicolon (`;`) to end a SQL statement!

Exercises:

1. How many unique film_id's are there in film?
2. What is the average film length?
3. Which unique rental durations (in days) are there for the films? Please make sure the result is ordered.
4. What is the average film length per rental duration? Also make sure this result is ordered, in this case specifically by rental_duration.
5. What is the average replacement cost of a film for each distinct rental rate? In your query, make sure to order the result by the average replace cost. In this case, the easiest way to sort the data is by assigning an alias, such as ARC, to the average replacement cost.

6. For each distinct film_id, give the title of the film, and the number of actors that appeared in it. Make sure the number of actors is shown as an attribute called NumberOfActors.
7. What's the title of each film in which more than 12 actors appeared? Also here, return the NumberOfActors attribute.
8. Give the first and last name of each actor that appeared in more than 35 movies. Also indicate in how many movies he/she has appeared, make sure this attribute is called NumberOfFilms.
9. How many films are there in each category? Give the name of each category and the count.
10. What's the average duration per film in each category? Call this average duration AverageLength, and sort the result by AverageLength.

Part II

In this part, you will connect to your DBMS (MySQL Server, in your case) using Tableau, an interactive tool for data visualization, quite similar to Microsoft Power BI. As with the questions on "sakila", you will also be working on these exercises in break-out rooms. So you do not need to prepare these exercises prior to the tutorial.

Please make sure, though, that you have followed all installation instructions at the beginning of this document **before** the tutorial. Ignas and I will be moving between break-out rooms, to see if there are any questions or if we can give some assitance.

Please note that Tableau will also be used in the last group assignment.

Exercise 1. Connect Tableau to the database.

Start Tableau.

- Upon starting the application, you will be greeted by the main page. In the *Connect* panel on the left side of the page, you can connect to various data sources such as Excel spreadsheets, PDF files and SQL databases (see Figure 1).
- Select MySQL under the *To a Server* section. Note: if you do not have the MySQL ODBC connector installed on your machine, you will be asked to install it—in that case, click on *Download Driver* and follow the instructions.
- Use the details below to connect to the database that you will use in this part of the tutorial:
  - Server: localhost
  - Port: 3306
  - Database: world
  - Username: root
  - Password: *enter your root user password here*

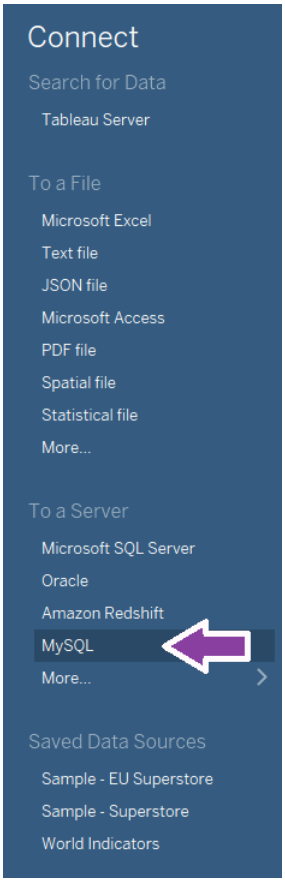

Figure 1. *Connect* panel in Tableau

You should now be redirected to the *Data Sources* tab (note: tabs are show on the bottom left side of screen), in which you see the *world* database on the left, together with its constituent tables (see Figure 2).

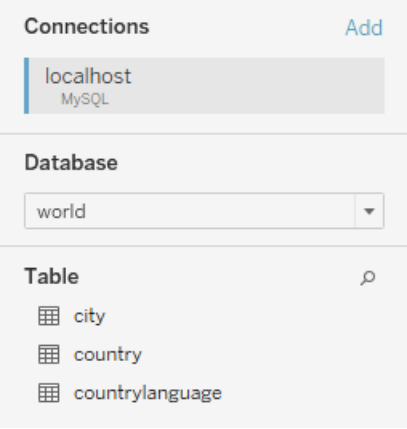

Figure 2. *world* database from Tableau

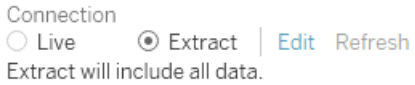

Figure 3. *Connection* type in data model

Exercise 2. Prepare the data.

In Tableau, you often create a data model, which is effectively a diagram telling Tableau how the various tables mash together. That is, such a model is much like an E/R diagram. Let's start with a basic data model that includes the *country* table.

- Double click on *country* table to include it into the data model.
- Now, in the upper-right part of the data model, make sure to select *Connection* type: *Extract* (see Figure 3).
- In the data model, click on the *country* box, and then click the *update now* button (see Figure 4).
- Notice that Tableau automatically recognises what certain data types represent. E.g. the first column (*Code*) is a string attribute, which Tableau recognises to reflect a country code. Clicking on *Abc*, above the attribute name, allows you to force Tableau to interpret the data type differently.
- Now EITHER drag the *city* table to the data model OR add *city* to the data model by double clicking on it.

The moment *city* is added to the data model, you will see an *Edit Relationship* window appear (see Figure 5). This window basically allows you to define the join condition that should apply when bringing data from *city* and *country* together.

- For *country* click on the field *Code* and for *city* click on the field *Country Code* and close the Edit Relationship window, by clicking on the cross in the upper-right corner of that window (note:

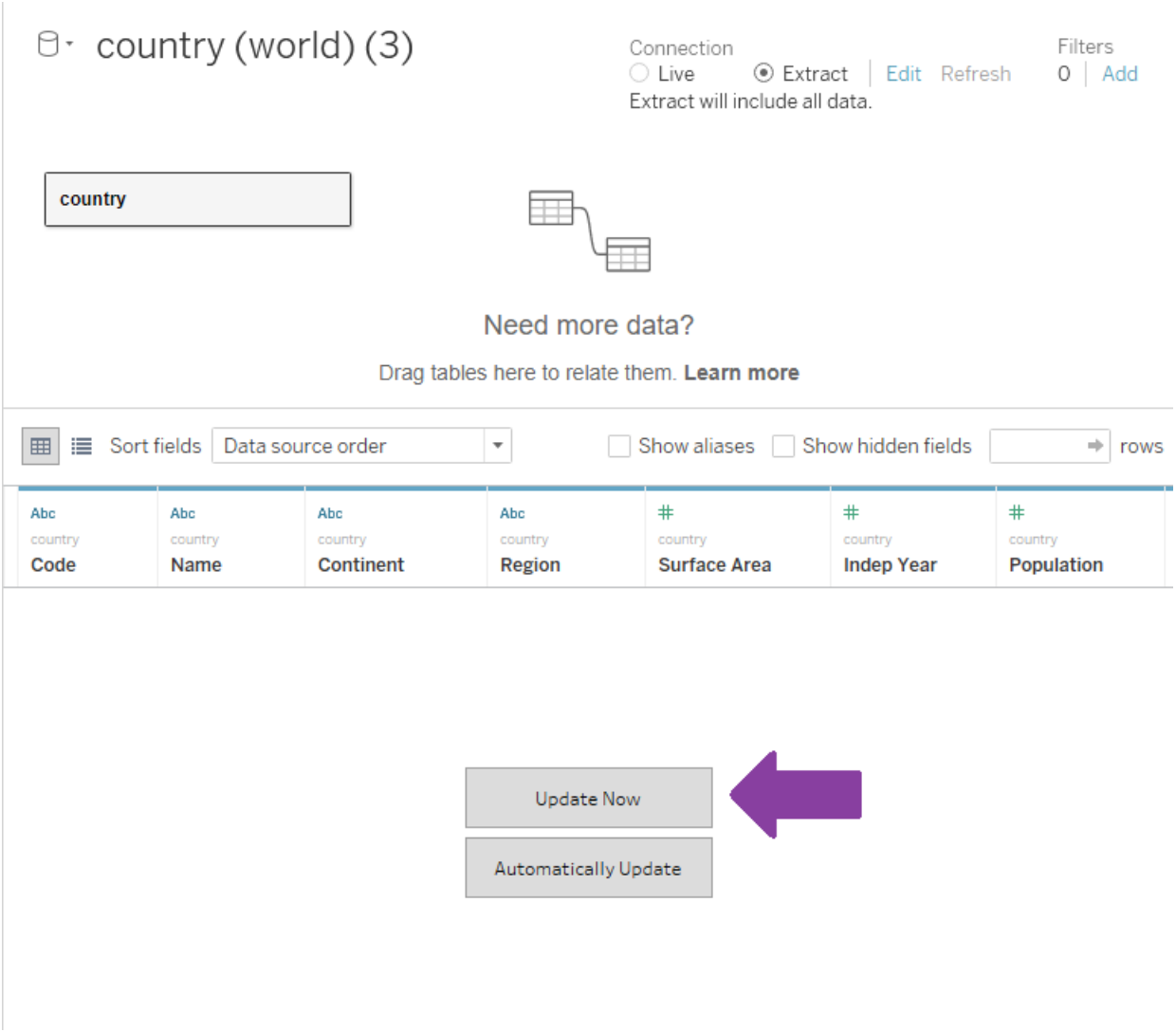

Figure 4. *Update Now* button in data model

you can change the connection type afterwards by pressing on the link-line between *country* and *city* in the data model). Click *Update Now*.

- Skim over the data in the *city* node, as shown underneath the data model. Notice how Tableau recognises city names and country codes as having geographic roles—this is useful for creating map visualisations, as you will see later!

- Click on the *country* node in the data model, then in the data underneath the model, right-click anywhere in the header and choose *Create calculated field*. This allows you to add a custom field, that is derived from other fields (much like you can do with the `SELECT` keyword in SQL).

- Now, enter *Density* as name of the calculated field, and enter [Population]/[Surface Area] as calculation. This will create a new, derived column that Tableau can use.

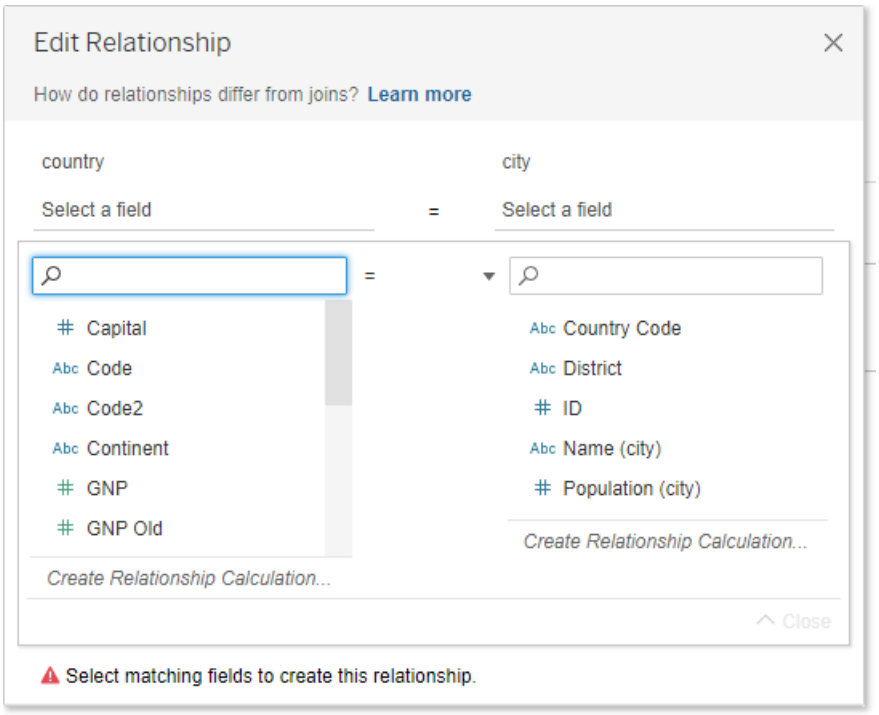

Figure 5. *Edit Relationship* window

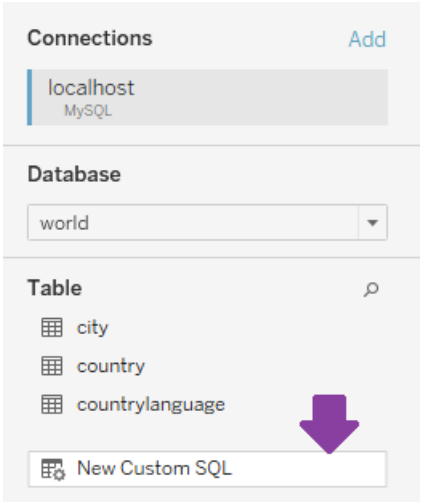

Figure 6. *New Custom SQL* window

- Right next to the new field, *Density*, is a button to sort the data. Sort the data to answer the question which country has the highest population density.
- Let's now include a custom SQL statement! Double-click *New Custom SQL* in the panel on the left (see Figure 6). Next, copy and paste the following statement into the field:

```
SELECT country.Code AS CountryCode,
       country.Name AS CountryName,
       city.Name AS CityName,
       (city.Population/country.Population)*100 AS PercPopInCity
FROM city, country
WHERE city.CountryCode = country.Code
ORDER BY PercPopInCity DESC
```

- Do not add a semicolon when entering a custom SQL statement in Tableau!
- What does this SQL statement do?
- Now the statement has been added, in the data model, click on the Custom SQL Query, and hit the *Update Now* button.
- Use the result of this query to find out which city in which country has the largest part of the country's population residing in that city.

Notice: you could have calculated the percentage of the country's population residing in a certain city without writing any SQL code, but by creating a worksheet instead, e.g. as shown in Figure 7. Such worksheets will be discussed in more detail in the next exercise.

Importantly, sometimes it is just easier to write SQL code directly, especially when data is very big or the calculation is very specific. In fact, the exact ordering seen in the SQL code is, for instance, already quite difficult to replicate using just the drag-and-drop functionalities in Tableau worksheets.

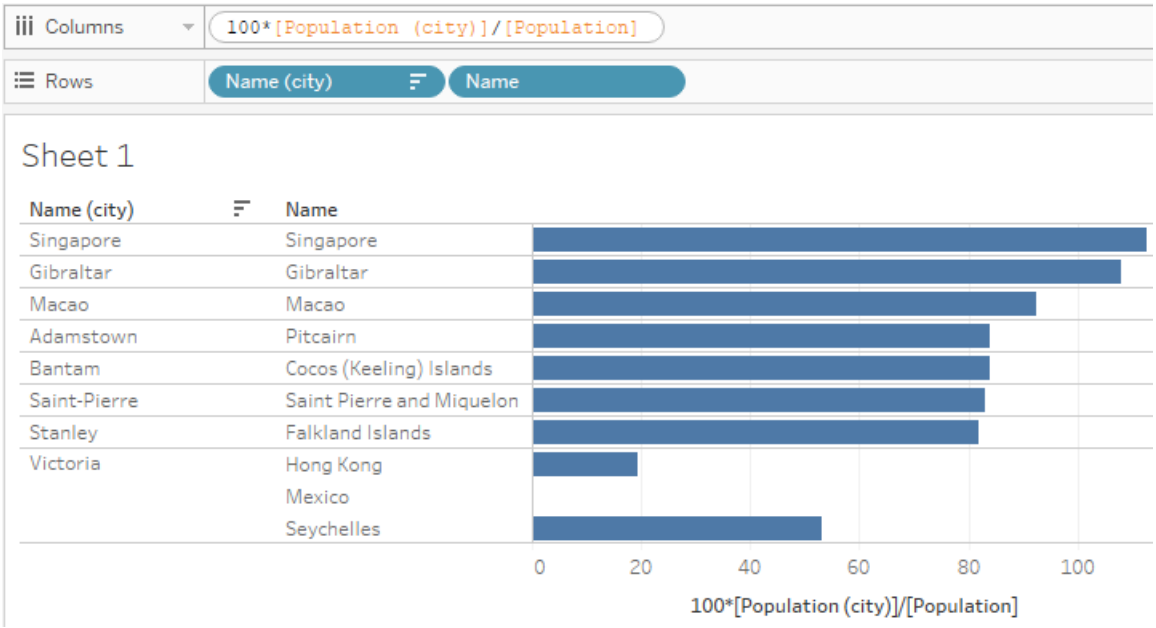

Figure 7. A Tableau sheet showing population percentage per city

Exercise 3. Create a bar visualisation.

- Create a new work sheet by pressing the *New Worksheet* button at the bottom left (highlighted in Figure 8).
- Worksheets are used to create individual visualisations that can later be combined into a dashboard.
- Drag *Population (city)* from *City* to Columns and *Name (city)* from *City* to Rows. Also drag *Name* from *Country* to Rows—after all, city name by itself is not a unique identifier; certain city names appear in multiple countries (e.g. London).
- What do you see now?
- Now open the *Show Me* drop-down menu on the right side of the worksheet, and try all six options highlighted using purple boxes, seen in Figure 9. What does each option represent?
- Switch back to the simple horizontal bar plot (with *Population (city)* in Columns and *Name* and *Name (city)* in Rows).
- Now drag *Name* from the *country* table to the Filters section. Select All and press OK. For now the filter (as seen in Figure 10) does not do anything.
- To make the filter visible, right-click on *Name* in the *Filters* section and choose *Show Filter*. Now a filter should appear on the right.
- In this filter on the right, press on the downwards-pointing triangle (highlighted in purple in Figure 11) and change the filter type to *Single Value (dropdown)*.
- Press on the downwards-pointing triangle again, and under *Apply to Worksheets* choose *All using related data sources*. This step ensures that the filter is applied to all visualisations that we create in the tutorial, after they are combined in a single dashboard.

Figure 8. Opening a new work sheet.

Exercise 4. Map visualisation

- Create another new worksheet.
- Drag *Name (city)* from *city* table to the big white field on the right. That is, drag it straight into the sheet itself. This should automatically create a map visualisation. Next, also drag *Country Code* from *city* to the map visualisation.

Figure 9. Some visualisation options in the Show Me menu.

Figure 10. New filter using *Name* from *country*.

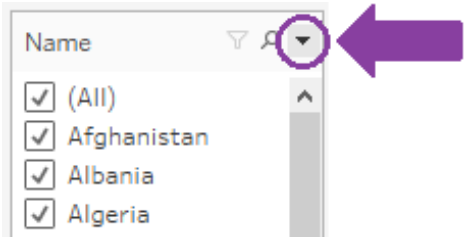

Figure 11. Opening the drop-down menu of the new filter.

- Drag *Population (city)* to *Size* in the *Marks* panel (shown in Figure 12), just left of the sheet itself.
- Drag *Government Form* from the *country* table to *Color*, also in the *Marks* panel.
- What do you see?

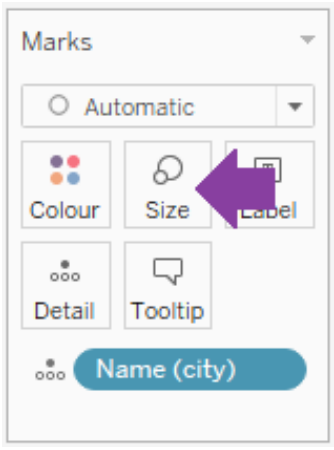

Figure 12. Marks panel, with the *size* field highlighted.

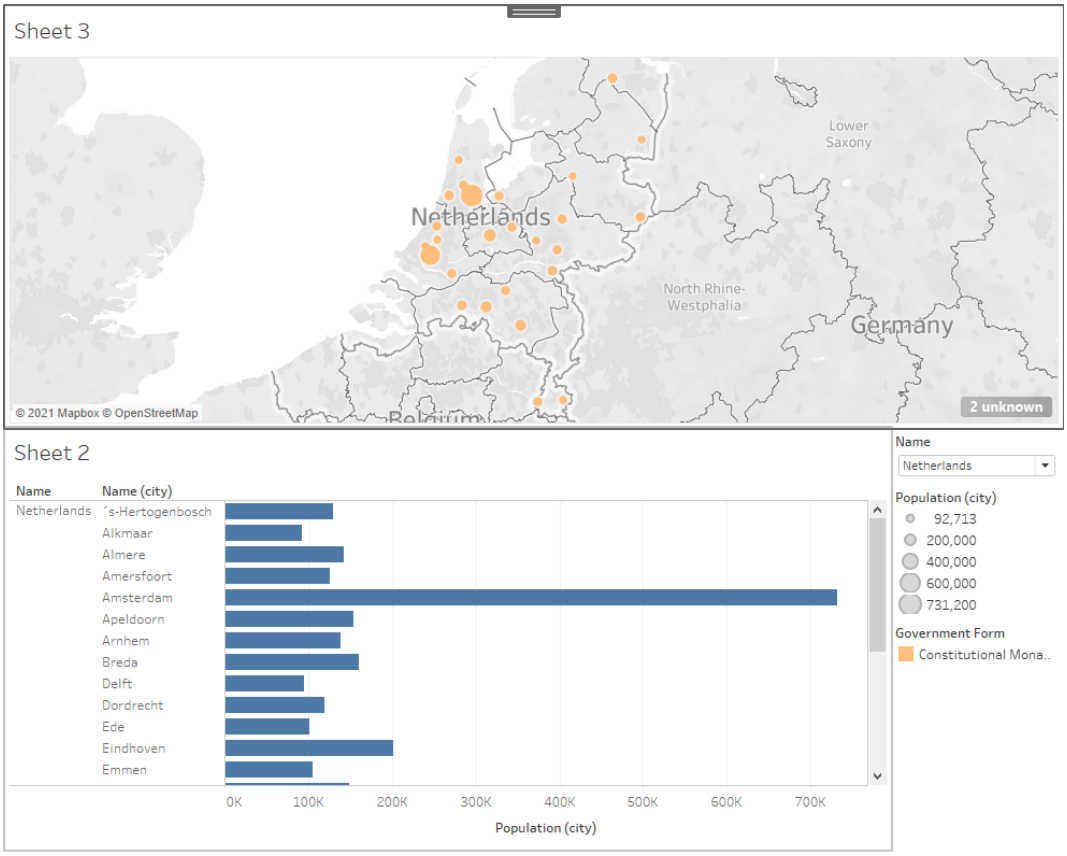

Figure 13. An example of a dashboard, using a filter.

Exercise 5. Creating a dashboard

- Create a new dashboard by clicking *New Dashboard* (right next to *New Worksheet* button).
- From the sheets on the left, drag both sheets that you have created to the big white field in the middle, saying *Drop sheets here*.

- Once these sheets have been dropped, notice that all parts can be moved around easily, and you can for instance apply the previously created filter. For instance, in Figure 13 you see city sizes specifically for the Netherlands.
- Your Tableau Dashboard using data from the "world" database is now complete!
- Feel free to save your work as a **.twbx (Tableau Packaged Workbook)** file and also feel free to show us your dashboard!

This brings us to the end of the tutorial on using Tableau. As you can see, Tableau is a versatile tool for interfacing with a database and for visualising the data therein. Tableau is a tool that is similar to Microsoft Power BI. Both tools are increasingly popular for data visualisation in businesses.

Nevertheless, using tools like MySQL Workbench or programming languages like Python, to put SQL queries to your DBMS directly is often preferable when you have a very specific or complex query and/or when you are working with a very large database.

References

Garcia-Molina, H., Ullman, J.D., and Widom, J. (2013). Database Systems, The Complete Book (2nd international edition; ISBN-10: 1-292-02447-X; ISBN-13: 978-1-292-02447-9). Essex, UK: Pearson Education Limited.