**FOLDER STRUCTURE**

Apart from this text file, the folder contains three objects:

* a “Code” folder,
* a “Data” folder, and
* the PowerPoint presentation.

The latter should complement this text file here on any explanations missing. In particular, it includes:

* the original database schema and a link to it by CTRL + click on its screenshot [slide 2];
* the adjustments made to it to obtain the ER diagram [slides 3-6];
* the ER diagram itself [slide 7];
* its restructuration to obtain the logical design [slides 8-12];
* the logical design itself [slide 13];
* a summary of the five queries addressed, with their respective sub-queries/developments [slide 14];
* a clearer presentation of the queries code and of what they achieve (which in the code itself is all in one line, hence hard to read) [slides 15-26].

The “Data” folder contains the tables downloaded from the original Kaggle database, in both “xlsx” and “csv” format; this is because the original files where Excel ones, but we then converted them to CSV so as to be able to open them from Python. Opening such files can help in understanding the “data extraction” part of the code, as specific columns of the original tables are referenced to extract the data.

The “Code” folder contains four python files, to be run in the order specified by their names:

* *“1 - Database\_creation”*, used to create the database;
* *“2 - Tables\_creation”*, used to create the tables into the database created in the previous step;
* *“3 - Data\_Extraction\_and\_Insertion”*, used to extract the data from the CSVs and to insert it into the tables created earlier;
* *“4 - Queries”*, which contains the code for running the queries explained in the presentation.

The code files also have some comments of their own, but they are more of a reference than a full explanation. For a complete explanation of how to run the code, follow the instructions on the next page.

**CODE INSTRUCTIONS AND EXPLANATION**

1. **Database creation**

In the “Code” folder, open the *“1 - Database\_creation”* file and change the *“user”* and *“passwd”* fields to your own MySQL user and password, respectively.

Run the file: you should obtain a list of all your databases in the console, among which should be the *“olympics”* one.

1. **Tables creation**

In the same folder, open the *“2 - Tables\_creation”* file and, again, change the *“user”* and *“passwd”* fields to your own MySQL user and password, respectively. The additional field *‘database = “olympics”’* ensures we are operating on the *“olympics”* database.

Next, run the file, and you should obtain 5 blocks of text in the console: the first four are the result of the command *“DESC table”* performed after having created each table, so as to check that it was created correctly; the last block of text, instead, is the result of the *“SHOW TABLES”* command, used at the end to check that all tables were created and are now present in the database.

1. **Data extraction and insertion**

Again, from the same folder, open the *“3 - Data\_Extraction\_and\_Insertion”* file and, once again, change the *“user”* and *“passwd”* fields to your own MySQL user and password, respectively.

Now, run the file and expect it to take a few seconds.

Each table is filled up by extracting the data from the CSVs with a simple “for”loop and inserting it, line by line, into the corresponding table. A few special constructs were used to deal with inconsistencies in the original database:

* in line 35, a Python list of all NOCs is created; it is, then, filled in line 38 and used in lines 56-57.

*This was done to deal with NOCs absent in the “Medals” file, but present in the “Athletes” one; for such NOCs, they were inserted in the NOCs table with 0 medals won, so that they could then be used as an attribute for their athletes, in the ATHLETES table.*

* in line 52, a Python list of all athletes is created, which is then filled in line 68 and used in line 62.

*This served the purpose of dealing with homonym athletes; in such cases, having checked that we had already encountered an athlete before, we skipped it with the “continue” command in line 63, which skips the next lines (65-68) and so avoids inserting such duplicate into the ATHLETES table.*

* in lines 95-101, an ‘if’ statement is used to insert coaches from the *“coaches.csv”* file to the TEAMS table (already created and filled earlier, but up to the fourth column, leaving the fifth column empty for the coaches).

*This ‘if’ statement was used to deal with coaches having a ‘NULL’ event, and it solves the problem by inserting the coach in the TEAMS row where the discipline and the event are the same, as this was the case for all those disciplines with no special event in the “teams.csv” file, which in fact had no ‘NULL’ values.*

1. **Queries**

Open the last file of the folder, *“4 - Queries”*, changing one final time the *“user”* and *“passwd”* fields to your own MySQL user and password, respectively.

Then, feel free to execute the queries one by one, or all together: the results will be displayed in the console.

A brief title is given to each query and can be used to find the corresponding representation in the PowerPoint presentation, which can be especially helpful to read the more complex queries more easily than in just one line as in the code.