

## Exercises 4-5: Creating a Database and Changing the Database Structure

The group project for the **Software 1** course is a flight simulator game that includes a database. When you complete these exercises, you will learn to

- Understand the scheme of the project database.
- Build a database for your own project.

### 1. Instructions

In this exercise you will build the database for the flight simulator game. You will find a summary of the stages for setting up the database below. The tasks are described in more detail below the summary.

### 2. General Description

You will use your MariaDB server and build a database based on the relational model you used in the previous exercises. The relational model diagram is shown in Figure 1.

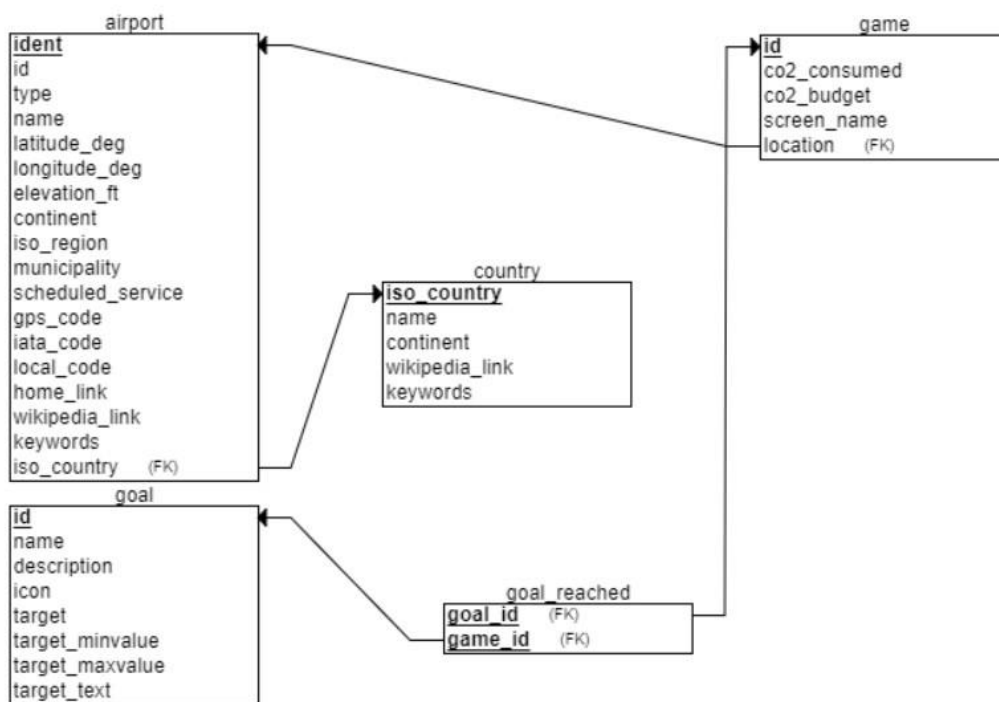


Figure 1 Airport database relational model

### 3. The airport and country tables

- You will get the data from the data files you downloaded before (aiports.csv and country.csv). **Remember not to open the files with Excel! Use Notepad++.**
- First you will create the tables so that they match the CSV files.
  - The **column names** are on the first row of the CSV files.
  - Pay attention to the **number of columns**. For the import to be successful, you must have exactly the same number of columns in your database as there are in the CSV file.
  - The final database will not be exactly the same as the CSV file contents. This will be corrected later after the import.

Skills you will use:

- Create table
- Alter table

### 4. The goal table

- The goal table will be similar to the relational model
- Data will be inserted later.
- Figure 2 shows the contents of the **goal** table in MariaDB.

Skills you will use:

- Create table
- Insert into

```
MariaDB [flight_game]> select * from goal;
```

| id | name   | description                 | icon | target  | target_minvalue | target_maxvalue | target_text |
|----|--------|-----------------------------|------|---------|-----------------|-----------------|-------------|
| 1  | HOT    | Temperature over +25C       | 01d  | TEMP    | 25.00           | 9999.00         | NULL        |
| 2  | COLD   | Temperature under -20C      | 13d  | TEMP    | -9999.00        | -20.00          | NULL        |
| 3  | 0DEG   | Temperature exactly 0C      | 04d  | TEMP    | -0.50           | 0.50            | NULL        |
| 4  | 10DEG  | Temperature exactly +10C    | 04d  | TEMP    | 9.50            | 10.50           | NULL        |
| 5  | 20DEG  | Temperature exactly +20C    | 04d  | TEMP    | 19.50           | 20.50           | NULL        |
| 6  | CLEAR  | Clear skies                 | 01d  | WEATHER | NULL            | NULL            | Clear       |
| 7  | CLOUDS | Cloudy                      | 04d  | WEATHER | NULL            | NULL            | Clouds      |
| 8  | WINDY  | Wind blows more than 10 m/s | 04d  | WIND    | 10.00           | 9999.00         | NULL        |

8 rows in set (0.001 sec)

Figure 2 Data values in the goal table.

### 5. The game and goal\_reached tables

- These tables will follow the relational model.
- As the player plays the game, these tables will be filled with data. The tables are used to store the game state.
- To practice making queries to all tables in the database, we will manually insert dummy text data to the tables. Later these dummy data will be deleted.

Skills you will use:

- Create table
- Insert into
- Delete

## 6. Detailed step-by-step instructions

1. Create a database called **flight\_game**. You can write the script for creating the database on Notepad++ and then copy the finished script to the CLI (Command Line Interface) of MariaDB. This way you will always have a copy of the entire script if something goes wrong. If you run into any major problem, you can easily destroy the entire database, modify the script if you need to, and run it again. This saves a lot of time.
2. Run the use **flight\_game**; command to select the database you are working with.
3. Create tables. Select suitable data types for the columns. Plan the order in which to create the tables. To maintain referential integrity, a referenced table must exist before the corresponding foreign key. You can follow these instructions to maintain a good order.
  - a. Create the **country** table (six columns in the CSV file). Select suitable data types. Do not define the primary key yet. This table will be later altered in term of column count, column names and primary key to match our relational model. However, these modifications cannot be done yet. For now, it is important that the table matches the CSV file exactly.
  - b. Create the **airport** table (18 columns in the CSV file). Select suitable data types. You can already define the ident column as the primary key. Do not define a foreign key yet. It will be handled later.
  - c. Create the **goal** table. It can be an exact match to the relational model (column number and names). Do not forget to define the primary key. (The data type of the icon column is varchar).
  - d. Create the game table. It can be an exact match to the relational model (column number and names). Do not forget to define the primary key. Do not define the foreign key yet. It will be handled later.
  - e. Create the **goal\_reached** table. It can be an exact match to the relational model (column number and names). Do not forget to define the primary key (a composite of two columns). Do not forget to define the foreign keys.
4. Insert data into the **goal** table. The values should look similar to the example in Figure 1.
5. Open HeidiSQL. The session manager of HeidiSQL is seen in Figure 3.

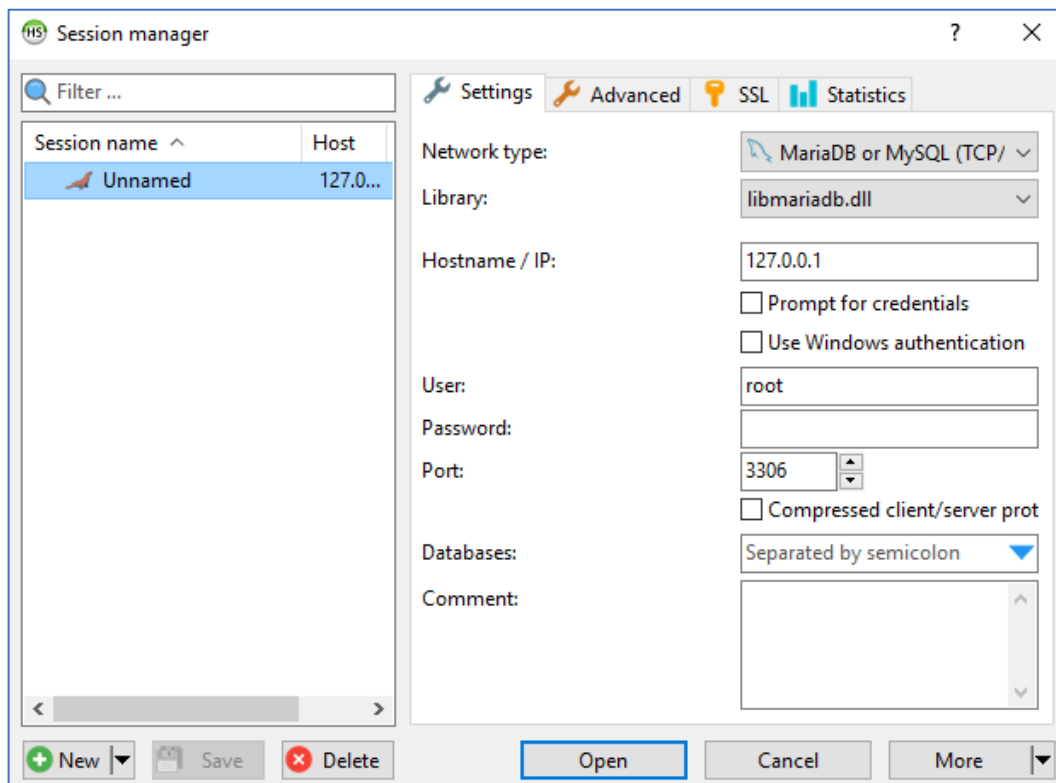


Figure 3 HeidiSQL session manager

6. The password is the same password you selected when you installed MariaDB. If you have forgotten the password, uninstall MariaDB, install it again and this time remember your password! (Also, remember that the password will be visible in the source code and seen by all your group members.)
7. Select your **flight\_game** database from the left panel to activate it.
8. Import data to the country table. Figure 4 shows how to open the import window. Figure 5 shows the correct settings for the import.

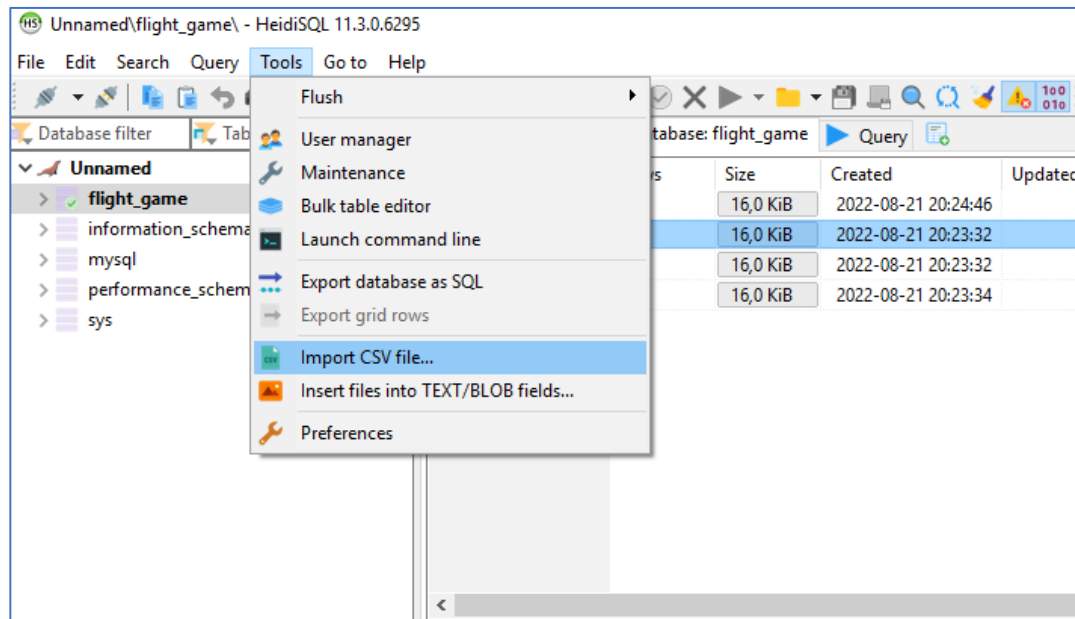


Figure 4 Importing data from CSV

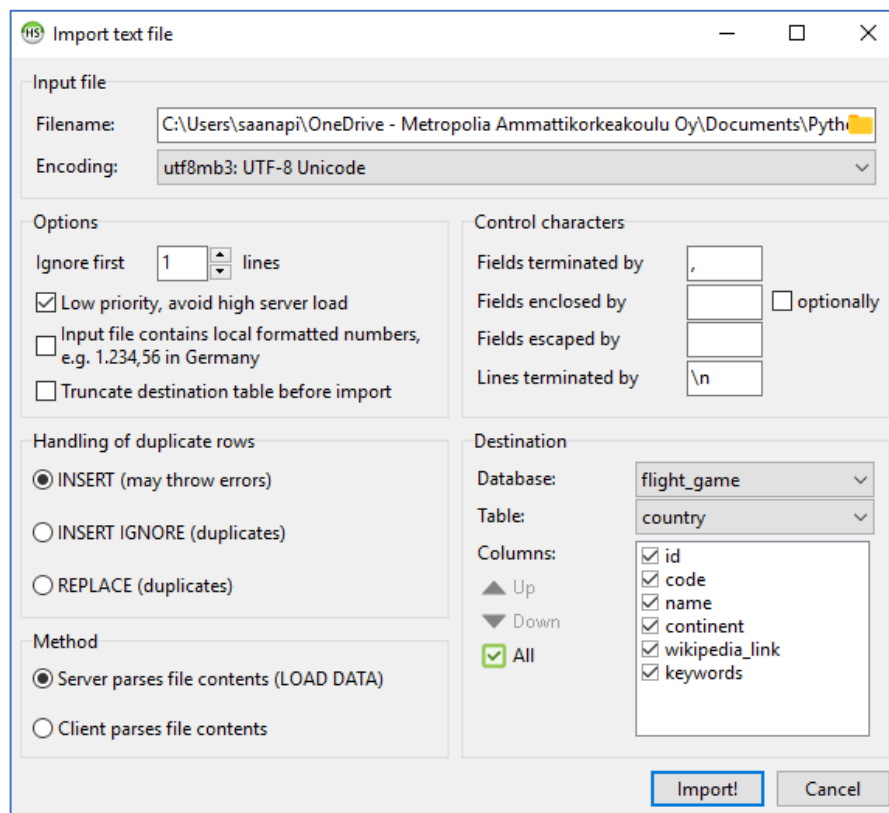


Figure 5 Settings for importing CSV data

- a. In the file name field, find the path to the **countries.csv** file on your computer.
- b. Encoding: **UTF-8 Unicode**.
- c. The terminating symbol for data fields is , (You can see this in the CSV file).
- d. Lines are terminated by \n (This cannot be seen in the file, but it is a common line terminator symbol in IT).

- e. Destination: Database: **flight\_game**
  - f. Destination: Table: **country**
  - g. Destination: Select **All** to import all columns.
  - h. **Press the Import! button once.** Pressing the button multiple times may lead to the data being imported more than once.
9. Import the data to the **airport** table.
- a. Repeat the same steps a-h as with the **country** table. Notice that the file name is **airports.csv** and the name of the table is **airport**.
10. Modify the **country** table.
- a. Remove the **id** column.
  - b. Change the name of the **code** column to **iso\_country**.
  - c. Set the **iso\_country** column as the **primary key** of the table.
11. Modify the airport table.
- a. Set the **iso\_country** column as the primary key that references the **iso\_country** primary key of the **country** table.
12. Modify the **game** table.
- a. Set the **location** column as the primary key that references the **ident** primary key of the **airport** table.
13. Insert dummy text data into the **game** table. Insert the following 3 lines of data:
- 1, 2000, 10 000, "Heini", "EFHK"
  - 2, 3000, 10 000, "Vesa", "EGCC"
  - 3, 8000, 10 000, "Ilkka", "EGKK"
14. Insert dummy text data into the **goal\_reached** table. Insert the following 4 lines of data:
- 1,4
  - 1,7
  - 2,4
  - 3,7