PHP / SQL practical evaluation

Time allowed: 4h00

Marking

At the end of the allocated time, you will have to deliver all of your files, and a mark out of 20 will be awarded according to the following scale:

- **3 points** for exercise 1
- **8 points** for exercise 2
- **5 points** for exercise 3
- **4 points** for exercise 4
- **2 points** on the following criteria:
  - Indentation of the code
  - Comments
  - Relevance of the name of the variables
  - Readability

NB: it is not permitted to copy and use code obtained from other students of this training or previous students of this training - doing so is considered cheating and will be sanctioned.

Bonus : bonus points may be added if the optional functions have time to be carried out. Do it only if you have the time!

Tips

If the task seems overwhelming, don't worry! Each step is detailed to move forward little by little.

- Read the entire statement from the start, to know where you are going.
- Take the time to code, commenting on your code as soon as it is necessary, why not by copying the instructions in comment.
  The proofreader must be able to understand what you have done!
- Keep functions simple, which do only one thing, the better to navigate.
- Focus on PHP rather than CSS/HTML !
- The more your code will be indented and readable, the easier the rest will be!

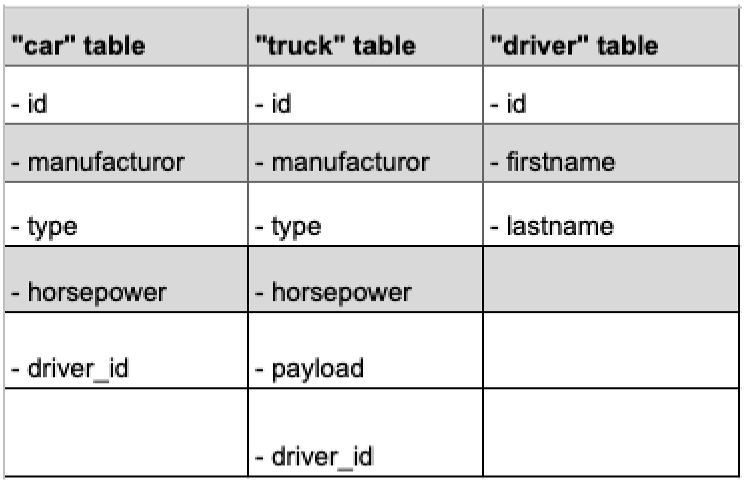
Each exercise is independent, be sure to separate them into different files or folders.

Exercise 1:

Learning objective to be validated:

- PHP / MySQL

Based on this DB structure :



"car" table	"truck" table	"driver" table
- id	- id	- id
- manufacturor	- manufacturor	- firstname
- type	- type	- lastname
- horsepower	- horsepower	
- driver_id	- payload	
	- driver_id	

You need, in a separate SQL file, to write the query to display all the cars along with their respective driver's name.

You will also need to create the DB in PHPMYADMIN and export the SQL file at the end of the evaluation.

BONUS : You also need, in the same file, to display all the vehicles along with their respective driver.

Note: Only write the SQL query, no PHP.

Exercise 2:

Based on the previously created tables and the requisites below, create the corresponding classes.
For this purpose we expect you to use class inheritance.

Required conditions :
Driver's information has to be validated.
Create setters to validate the property type. First name and last name must be string type.
The age must be an integer.

Driver will have a method named 'insert()'. This method should insert the driver in the DB (you have to use the driver's property).

Car and Trucks must be immutable and it will not be possible to change their properties after instantiation.

Exercise 3:

In this exercise, you will have to create a form.

This form will allow you to add a new driver to the database using the Driver class created in Exercise 2.

You have to follow those recommandations :

- The form will be processed in Ajax.
  You can use Ajax in a classic way or with the help of jQuery.
- Validate the input coming from the form.
- Create a Driver object.
- Insert this object into the DB :
  - -- Use the insert() method created previously in exercise 2.
  - -- If you can’t do it, find another way to insert the Driver in the DB.

- A success or error message will be displayed.

Exercise 4:

Display all the drivers on a page.

You will also need to display all the vehicles related to those drivers.

For this exercise, you will need to retrieve information from the database in the form of object(s).

Expected output for one driver :

```
   Driver : John
   Vehicles :
      Car1
      Car2
      Truck1
```