



# Piscine datascience - 0

## Creation of a DB

*Summary: Today, you will discover the creation of a DB*

*Version: 1.00*

# Contents

<b>I</b>	<b>General rules</b>	<b>2</b>
<b>II</b>	<b>Introduction</b>	<b>3</b>
<b>III</b>	<b>Exercise 00</b>	<b>4</b>
<b>IV</b>	<b>Exercise 01</b>	<b>5</b>
<b>V</b>	<b>Exercise 02</b>	<b>6</b>
<b>VI</b>	<b>Exercise 03</b>	<b>7</b>
<b>VII</b>	<b>Exercise 04</b>	<b>8</b>
<b>VIII</b>	<b>Submission and peer-evaluation</b>	<b>9</b>

# Chapter I

## General rules

- You have to render your modules from a computer in the cluster either using a virtual machine:
  - You can choose the operating system to use for your virtual machine
  - Your virtual machine must have all the necessary software to realize your project. This software must be configured and installed.
- Or you can use the computer directly in case the tools are available.
  - Make sure you have the space on your session to install what you need for all the modules (use the goinfre if your campus has it)
  - You must have everything installed before the evaluations
- Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors. If this happens, your project will be considered non functional and will receive a 0 during the evaluation.
- We encourage you to create test programs for your project even though this work **won't have to be submitted and won't be graded**. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.
- Submit your work to your assigned git repository. Only the work in the git repository will be graded. If Deepthought is assigned to grade your work, it will be done after your peer-evaluations. If an error happens in any section of your work during Deepthought's grading, the evaluation will stop.
- By Odin, by Thor ! Use your brain !!!

# Chapter II

## Introduction

In the next two modules, you will see the role of a data engineer.

This second step is important to understand. The data engineer "cleans" the data and transforms it in order to have data ready to be analyzed by analysts/data scientists.

The next module involves data cleansing. This second step is important to understand the data engineer "cleans" the data and transforms it. The objective is to have data ready to be analyzed by analysts/data scientists.


We are at the end of February 2022, it's your first day in a company selling items on the Internet. Before leaving on a trip your boss gives you the sales of the last 4 months. You will have to exploit them and propose solutions to increase the turnover of the company.



**Be careful** with this "piscine". Even if you manage to validate a module, you may be stuck later if you haven't cleaned up or stored your data properly.

# Chapter III

## Exercise 00

	Exercise 00
Exercise 00 : Create postgres	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <b>Makefile</b> , <b>Dockerfiles</b> , <b>docker-compose.yml</b>	
Allowed functions : <b>All</b>	

You have to use docker compose. You also have to write your own Dockerfiles. The Dockerfiles must be called in your docker-compose.yml by your Makefile.


- A Docker container that contain postgres
- The name of the image is piscineds
- The password is "mysecretpassword"
- Configure the ports to connect to your postgres image

We must be able to see your image with this command:

```
$>docker-compose ps -all
CONTAINER ID   IMAGE          COMMAND                  SERVICE   CREATED        STATUS          PORTS
XXXXXXXXXXXX   piscineds     "docker-entrypoint.s."   foo       6 seconds ago  Up 6 seconds    XXXXX
```

# Chapter IV

## Exercise 01


	Exercise 01
Exercise 01 : Show me your DB	
Turn-in directory : <i>ex01/</i>	
Files to turn in :	
Allowed functions : <code>pgadmin</code> , <code>Postico</code> , <code>dbeaver</code> or what you want to see the db easily	

- Find a way to visualize the db easily with a software
- The chosen software must help you to easily find and manipulate data using its own corresponding ID

event_time	event_type	product_id	category_id	category_code	brand	price	user_id	user_session
2022-10-01 00:00:00.000000	cart	5773203	1487580005134238464	<null>	runail	2.62	463248011	2eddd6ee-4dac-4778-8d2c-92e149dab885
2022-10-01 00:00:03.000000	cart	5773353	1487580005134238464	<null>	runail	2.62	463248011	2eddd6ee-4dac-4778-8d2c-92e149dab885
2022-10-01 00:00:07.000000	cart	5773498	1487580005134238464	<null>	runail	2.62	463248011	2eddd6ee-4dac-4778-8d2c-92e149dab885
2022-10-01 00:00:07.000000	cart	5881589	2151191071051219712	<null>	lovely	13.48	429681838	49e8d843-adf3-428b-a2c3-fe8bcaa307c9
2022-10-01 00:00:15.000000	cart	5881449	1487580013522845952	<null>	lovely	8.56	429681838	49e8d843-adf3-428b-a2c3-fe8bcaa307c9
2022-10-01 00:00:16.000000	cart	5857269	1487580005134238464	<null>	runail	2.62	430174032	73deale7-664e-43f4-8b38-d32b9d5af04f

# Chapter V

## Exercise 02

	Exercise 02
Exercise 02 : First table	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <b>table.*</b>	
Allowed functions : All	


- Create a postgres table using the data from a CSV from the 'customer' folder. Name the tables according to the CSV's name but without the file extension, for example : "data\_2022\_oct"
- The name of the columns must be the same as the one in the CSV files and have the appropriate type, beware you should have at least 6 different data types
- A DATETIME as the first column is mandatory



Be careful, the typings are not quite the same as under Maria DB

# Chapter VI

## Exercise 03

	Exercise 03
Exercise 03 : items table	
Turn-in directory : <i>ex03/</i>	
Files to turn in : <b>items_table.*</b>	
Allowed functions : All	

- You have to create the table "items" with the same columns as in the "item.csv" file
- You have to create at least 3 data types in the table
- You have to write a script/program that does it automatically because everything is deleted at the beginning of the evaluation

Below is an example of the expected directory structure:

```
$> ls -alR
total XX
drwxrwxr-x 2 eagle eagle 4096 Feb 42 20:42 .
drwxrwxr-x 5 eagle eagle 4096 Feb 42 20:42 ..
drwxrwxr-x 2 eagle eagle 4096 Jan 42 20:42 customer
drwxrwxr-x 2 eagle eagle 4096 Jan 42 20:42 items


./customer:
...

./items:
total XX
drwxrwxr-x 2 eagle eagle 4096 Feb 42 20:42 .
drwxrwxr-x 5 eagle eagle 4096 Feb 42 20:42 ..
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 items.csv
```



# Chapter VII

## Exercise 04

	Exercise 04
Exercise 04 : automatic table	
Turn-in directory : <i>ex04/</i>	
Files to turn in : <b>automatic_table.*</b>	
Allowed functions : All	

- We are at the end of February 2022, you should be able to create tables with data extracted from a CSV.
- Now, in addition, retrieve all the CSV from the 'customer' folder automatically and name the tables according to the CSV's name but without the file extension, for example : "data\_2022\_oct"

Below is an example of the expected directory structure:

```
$> ls -alR
total XX
drwxrwxr-x 2 eagle eagle 4096 Feb 42 20:42 .
drwxrwxr-x 5 eagle eagle 4096 Feb 42 20:42 ..
drwxrwxr-x 2 eagle eagle 4096 Jan 42 20:42 customer
drwxrwxr-x 2 eagle eagle 4096 Jan 42 20:42 items

./customer:
total XX
drwxrwxr-x 2 eagle eagle 4096 Feb 42 20:42 .
drwxrwxr-x 5 eagle eagle 4096 Feb 42 20:42 ..
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 data_2022_dec.csv
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 data_2022_nov.csv
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 data_2022_oct.csv
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 data_2023_jan.csv

./items:
...
```

# Chapter VIII

## Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.



The evaluation process will happen on the computer of the evaluated group.