Lux Pizzeria

## 12/11/2023 by: Imran Boutadghart

## summary

this will serve as the report for my project entitled “Lux Pizzeria”, the next section will present briefly the content of this report.

## Table of content

* summary
* table of content
* introduction
* scope
* environment requirements
* modeling part
* back-end development
* front-end development
* testing
* challenges
* references

**introduction**

For the module called “System d’information” we are required to build a pizzeria management project using one of the following technologies: PHP/MYSQL or MSACCESS/VBA or SWING (Java)/MYSQL.

**scope**

the technology I chose to work with is PHP MYSQL, and the reasons I chose this technology are

* + - * + **ease of learning**: as It is my first project in any of the mentioned topics, I chose what I found the easiest to learn with the steadiest learning curve, also due to my humble experience in HTML CSS, working with php would make the most sense out of my choices
        + **community support**: php has a huge active community second to only to java, which gives me<the developer> many resources to learn from as well as ask for any confusion I might face along the way.
        + **Integration with other technologies:** as mentioned before, I possess a humble experience in web development, so I had to choose the language that I would be most comfortable with considering integrability with technologies I know

so as prementioned, the project’s scope contains:

* + - * + PHP
        + SQL
        + HTML
        + CSS
        + Javascript

I also feel the need to mention my usuage of the git technology to keep track of different versions of the project, also to always have a safe check point toget back to if anything went wrong.

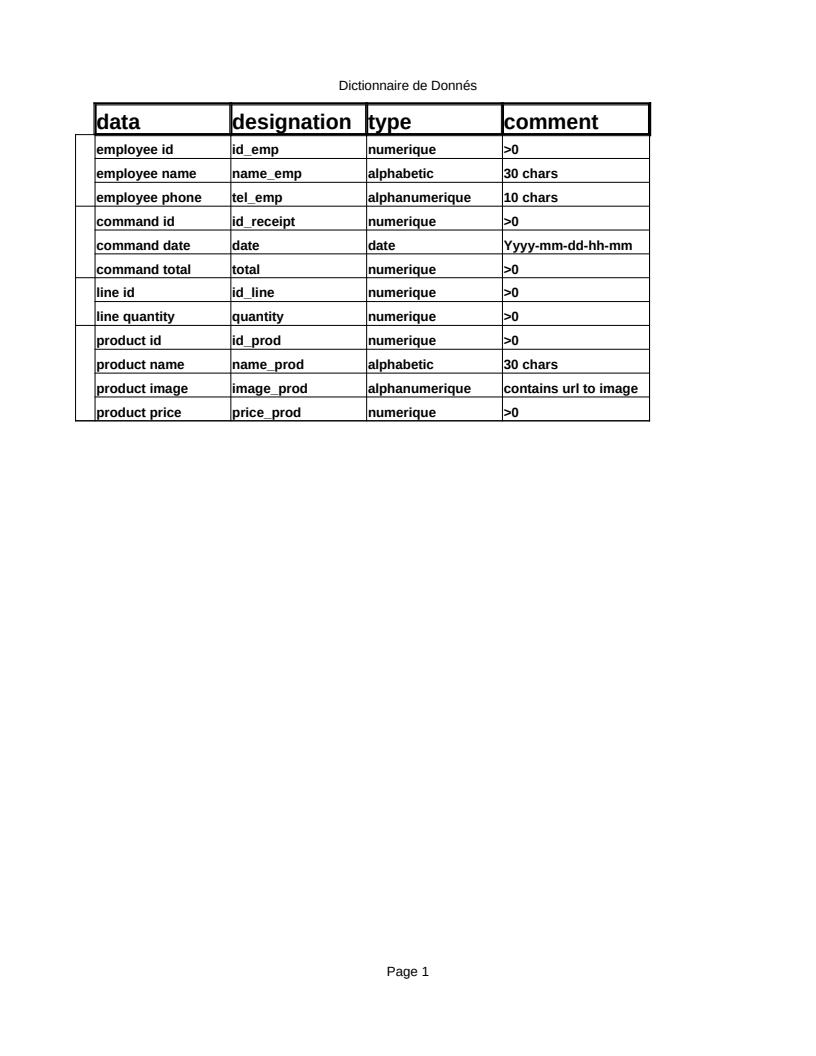
**environment requirements**

the requirements needed for the project to run is a mini web server running locally on your machine, a browser to render the php/html code, and having mysql server running in localhost.

**modeling part**

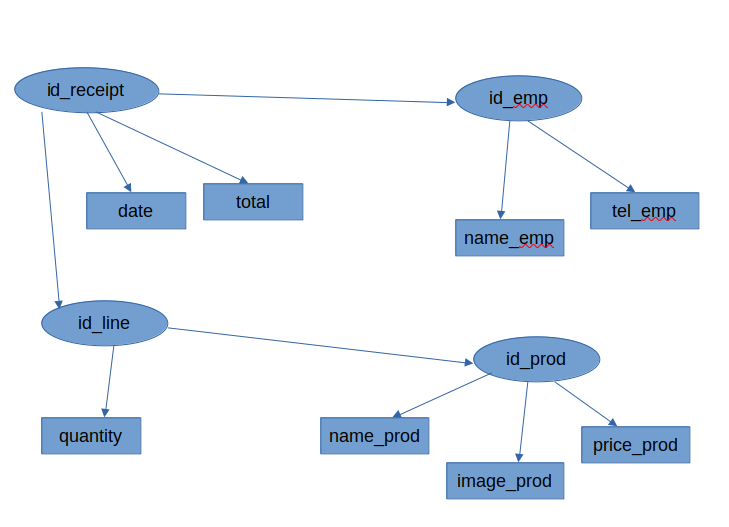
to begin my journey, i used MERISE method to make a general purpose model of my database.

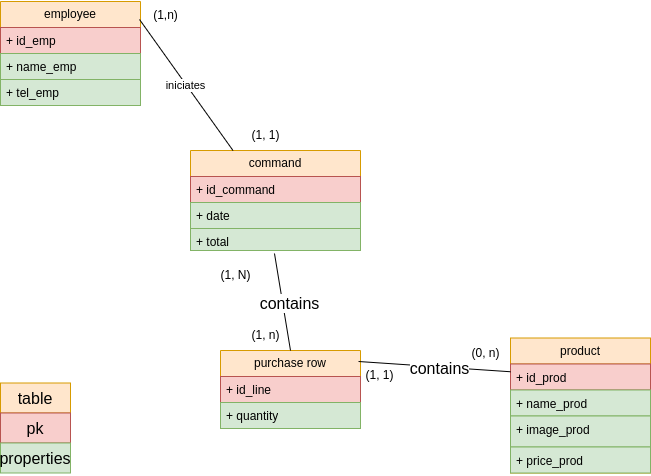
I started with putting down a data dictionary that went as follows:



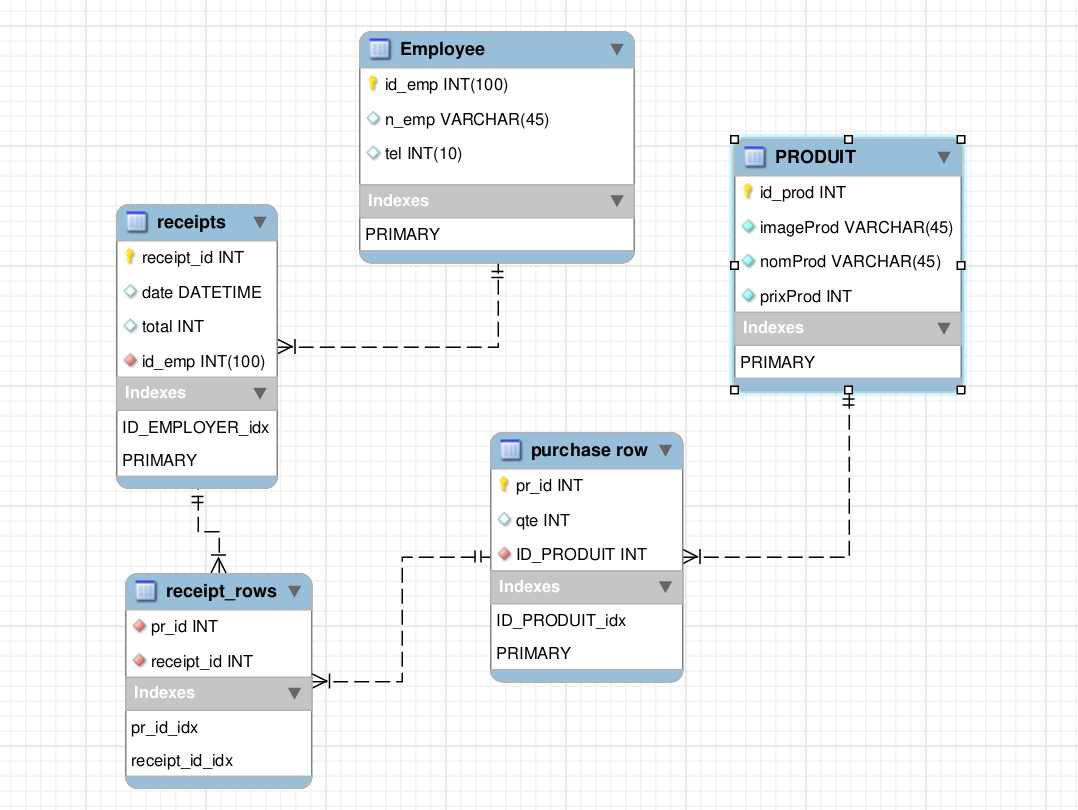
Then I proceeded to make a functional dependencies graph (GDF in french)

which is shown here:

as it may be shown, all my data is related in a somewhat tree pattern, which helped me build the MCD, a conceptual data model, I have to note that there is a performance issue in my MCD noted by my professor



it may not be very clear but the relation between purchase row and command does not have to be (1,n)(1,n) that is due to the ability to include purchase rows in the command table, this mistake will lead me to create an unnecessary table in my MLD (logical data model), speaking of which, here is the MLD that will be the foundation of my database:

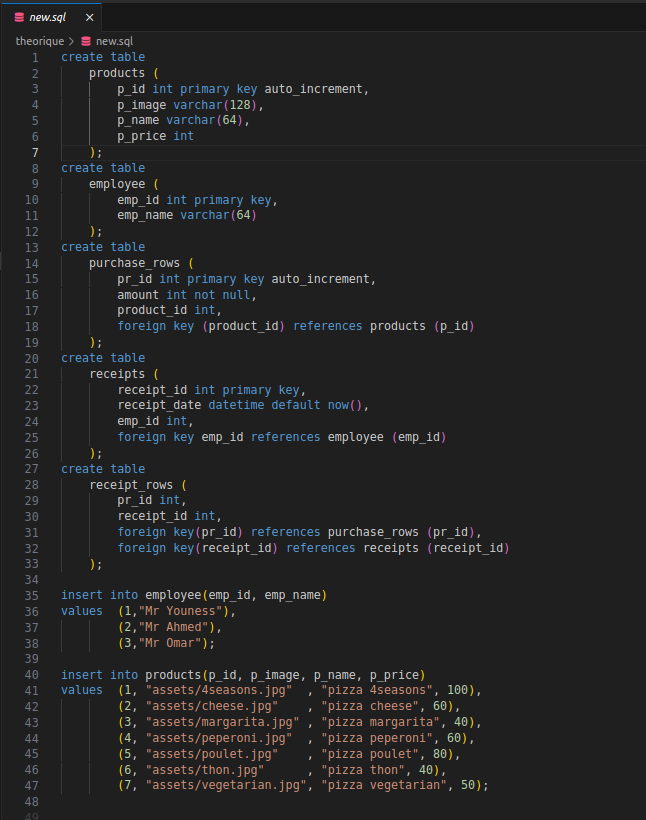


as you might have predicted, a 5th table was added due to that simple mistake I made in the MCD, which shows how important the MCD to build a database.

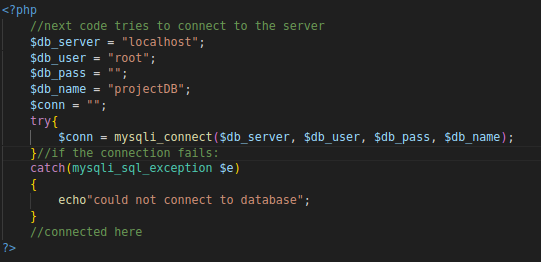
At this point I was ready to start building the front-back-end of my project, which I started right the way.

**back-end development**

to start of the back-end journey I started by creating the base tables which were employee, Product tables because they did not have any foreign keys in them, I then went on to build the other 3 tables, the process of building the tables was all saved in a file named “sql.sql”, and the reason is to have a backup code to build my back-end from scratch if something were to go wrong.

Here is the backup script I made.

after building the database, I used ‘mysqli’ set of functions, ‘mysqli’ is a set of functions in php that creates an interface to connect the user to the database, I also had to surround every ‘mysqli’ function in a try-catch block to protect my code from any exceptions that may occur during the connection to the database.

Heres an example of mysqli function use

After some front-end development I faced an issue, which was because the data I received from the user of the website was stored in javascript variables, meaning I could not use php directly to save my data in the database, which was the reason I built a function named “get\_receipt” the function uses an http request to a webpage named “receipt\_handler.php” with the data stored in the request, the request then gets processed by another connection that saves the data into the server.

The latter mentioned function uses the method print on my webPage when the user presses the get\_receipt button, which prints the receipt in a pdf form, the pdf is styled in a css print query to be well structured as requested in the subject of the project.

**front-end development**

the front end consisted of building an html page that holds items with the same naming and classing as my php page so that I test the css on it, which I did right the way, the pages consist of a header and a body, the header is used to navigate through different pages of my project, while the body (depending on the page), holds the items that are the main topic of the page, here are the pages my project have:

**testing**

**challenges**

**references**