

Data Base Project

Laptop selling store website

Pacuraru Fabian | Data Bases | 28.05.2021

# What was it made for?

This project aims to simulate a website that is connected to a database that could be used to see and manage the specifications, prices and stocks of laptops that a store or multiple stores might sell.

I worked just a little on the styling part so the project may seem not suitable for actual use, but the most important part, in my opinion, is that all the elements work as intended and for a first project using html, css and php, I am happy with its functionality.

# Tables

I started this project by creating 8 tables for the actual website and one for the login page. Then indexes must be created in order to be able to link the tables to one another. I also added a constraint of type UNIQUE on the field Username from table login, which is going to prevent storing multiple accounts with the same username.

## main

“CREATE TABLE main (

Laptop\_Id int(255) NOT NULL,

Display\_Id int(255) NOT NULL,

Processor\_Id int(255) NOT NULL,

Ram\_Id int(255) NOT NULL,

Storage\_Id int(255) NOT NULL,

Graphics\_Card\_Id int(255) NOT NULL

);”

“CREATE INDEX Laptop\_Id

ON main (Laptop\_Id);”

“CREATE INDEX Display\_Id

ON main (Display\_Id);”

“CREATE INDEX Processor\_Id

ON main (Processor\_Id);”

“CREATE INDEX Ram\_Id

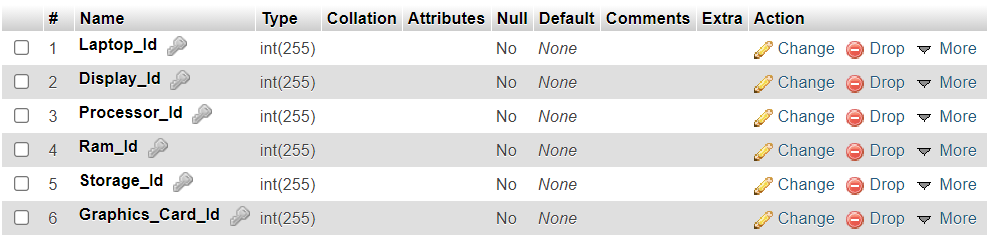
ON main (Ram\_Id);”

“CREATE INDEX Storage\_Id

ON main (Storage\_Id);”

“CREATE INDEX Graphics\_Card\_Id

ON main (Graphics\_Card\_Id);”



## laptop

“CREATE TABLE laptop (

Id\_l int(255) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Model\_l varchar(50),

Width\_l float,

Height\_l float,

Depth\_l float,

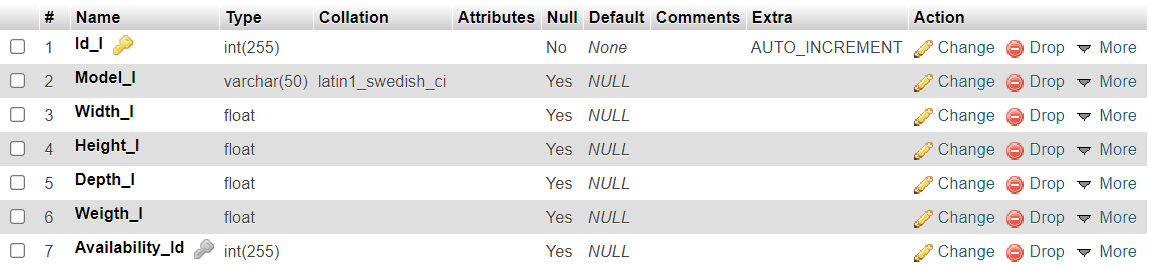
Weigth\_l float,

Availability\_Id int(255)

);”

“CREATE INDEX Availability\_Id

ON laptop (Availability\_Id);”



## processor

“CREATE TABLE processor (

Id\_p int(255) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Model\_p varchar(50),

Core\_Count\_p int(255),

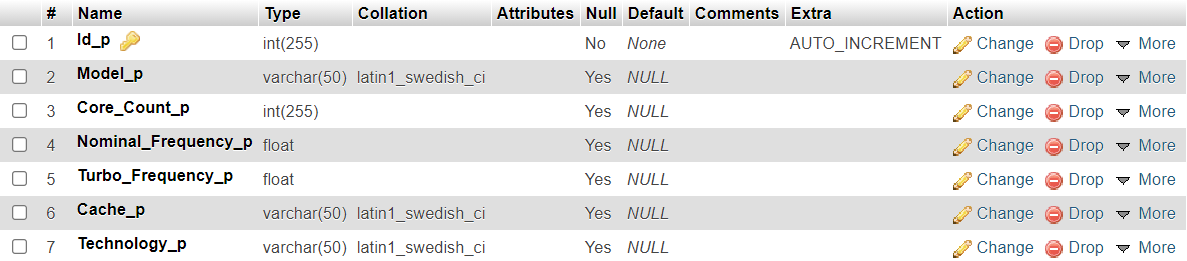
Nominal\_Frequency\_p float,

Turbo\_Frequency \_p float,

Cache\_p varchar(50),

Technology\_p varchar(50)

);”



## ram

“CREATE TABLE ram (

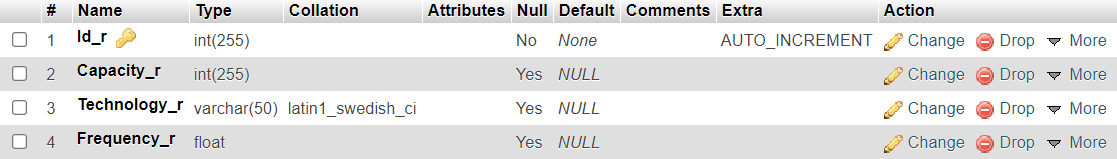
Id\_r int(255) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Capacity\_r int(255),

Technology\_r varchar(50),

Frequency\_r float

);”



## storage

“CREATE TABLE storage (

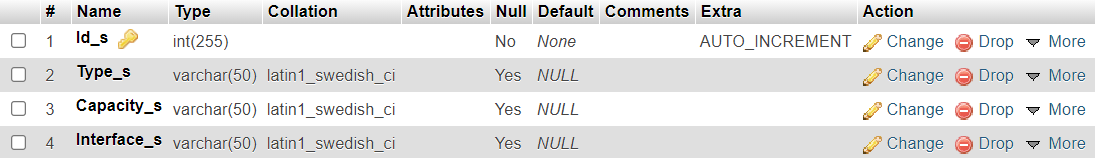
Id\_s int(255) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Type\_s varchar(50),

Capacity\_s varchar(50),

Interface\_s varchar(50)

);”



## display

“CREATE TABLE display (

Id\_d int(255) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

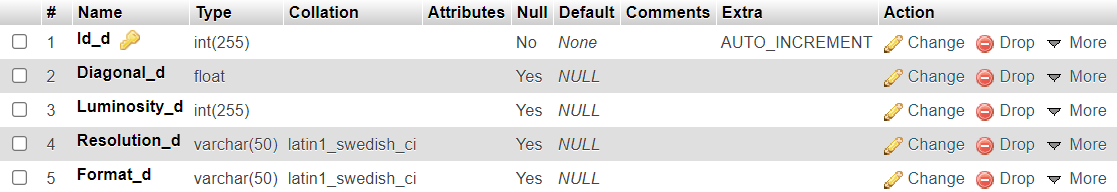
Diagonal\_d float,

Luminosity\_d int(255),

Resolution\_d varchar(50),

Format\_d varchar(50)

);”



## graphics card

“CREATE TABLE graphics\_card (

Id\_g int(255) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

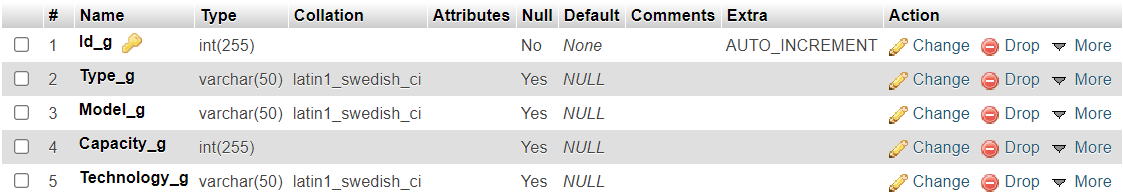
Type\_g varchar(50),

Model\_g varchar(50),

Capacity\_g int(255),

Technology\_g varchar(50)

);”



## availability

“CREATE TABLE availability (

Id int(255),

Store\_name varchar(50),

Deposit\_Id int(255),

Price float,

Stock int(255),

Bigger\_Deposit\_Id int(255)

);”

“CREATE INDEX Id

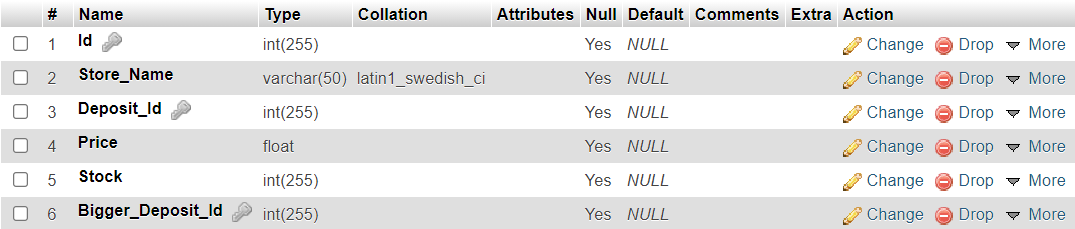
ON availability (Id);”

“CREATE INDEX Deposit\_Id

ON availability (Deposit\_Id);”

“CREATE INDEX Bigger\_Deposit\_Id

ON availability (Bigger\_Deposit\_Id);”



## login

“CREATE TABLE login (

Username varchar(50) NOT NULL,

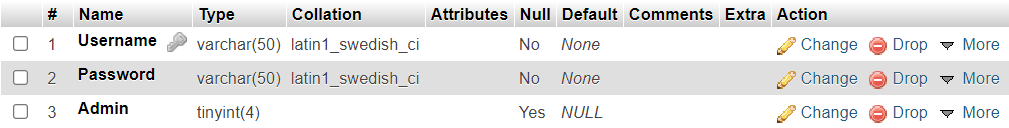
Password varchar(50) NOT NULL,

Admin tinyint

);”

“ALTER TABLE login

ADD CONSTRAINT Username UNIQUE (Username);”

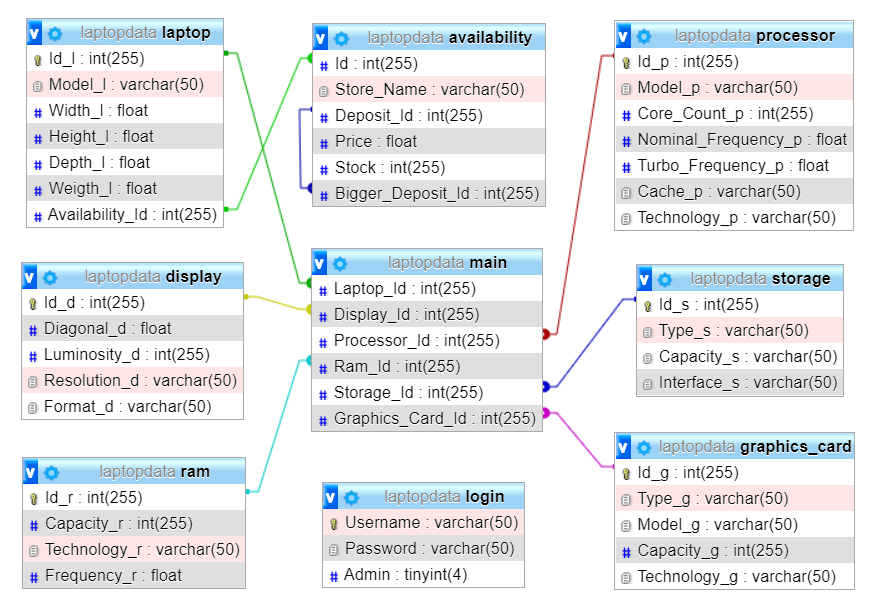


# Relationships between tables

I used phpMyAdmin designer and I created multiple relationships between tables as you can see in the next picture.

I used three types of relationships:

1. One to many
2. Many to many
3. Recursive



## One to many relationships

* From table laptop to table main with Id\_l = Laptop\_Id
* From table processor to table main with Id\_p = Processor\_Id
* From table ram to table main with Id\_r = Ram\_Id
* From table storage to table main with Id\_s = Storage\_Id
* From table display to table main with Id\_d = Display\_Id
* From table graphics\_card to table main with Id\_g = Graphics\_Card\_Id

## Many to many relationships

* From table laptop to table availability with Availability\_Id = Id

## Recursive relationships

* Into table availability with Deposit\_Id = Bigger\_ Deposit\_Id

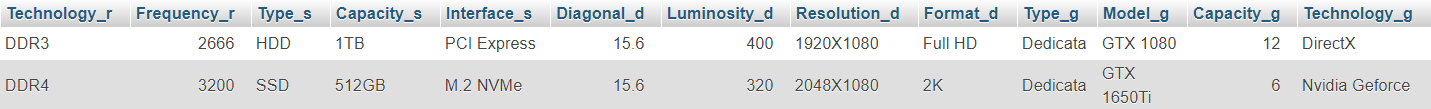
# Views

I created 4 views using some JOIN statements, some aggregate functions. I also ordered and grouped my results using ORDER BY, GROUP BY and LIMIT instead of TOP.

## First View

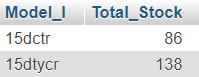
”CREATE VIEW view1 AS SELECT Model\_l,Width\_l,Height\_l,Depth\_l,Weigth\_l,Model\_p,Core\_Count\_p,Nominal\_Frequency\_p,Turbo\_Frequency\_p,Cache\_p,Technology\_p,Capacity\_r,Technology\_r,Frequency\_r,Type\_s,Capacity\_s,Interface\_s,Diagonal\_d,Luminosity\_d,Resolution\_d,Format\_d,Type\_g,Model\_g,Capacity\_g,Technology\_g FROM ((((((main JOIN laptop ON Laptop\_Id = Id\_l) JOIN display ON Display\_Id=Id\_d) JOIN processor ON Processor\_Id=Id\_p) JOIN ram ON Ram\_Id=Id\_r) JOIN storage ON Storage\_Id=Id\_s) JOIN graphics\_card ON Graphics\_Card\_Id=Id\_g)”





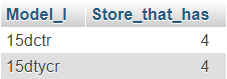
## Second View

“CREATE VIEW view2 AS SELECT Model\_l, SUM(Stock) AS Total\_Stock FROM laptop JOIN availability ON Availability\_Id=Id GROUP BY Model\_l”



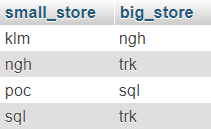
## Third View

“CREATE VIEW view3 AS SELECT Model\_l, COUNT(\*) AS Store\_that\_has FROM laptop JOIN availability ON Availability\_Id=Id GROUP BY Model\_l ORDER BY Store\_that\_has DESC LIMIT 5”



## Fourth View

“CREATE VIEW view4 AS SELECT X.Store\_Name AS small\_store, Y.Store\_Name AS big\_store FROM availability X JOIN availability Y on X.Bigger\_Deposit\_Id=Y.Deposit\_Id WHERE X.Deposit\_Id!=Y.Deposit\_Id GROUP BY small\_store, big\_store”

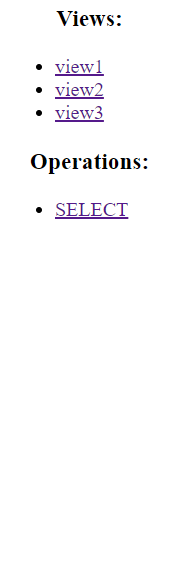


# How to use?

It is true that this website is not too pleasant to the eye, but it is quite simple to use in my opinion. All the actions that can be done into the main webpage, that was split between two sections, can be selected from the left section of the webpage. This is true for both admins and regular users, even though the admins will have much more options.

The users can see the first three views and the select option that will show them exactly what they want from the database, while the admins can see all four views, can select from, insert into, update and delete from the database, regarding first 7 tables. The admins can also add, modify or delete the login table that stores the credentials and whether or not that is a user or an admin and they can also modify the price and stock of products in different stores that are already in the database.

This is how the menu looks like for admins vs users:

 admin: regular user: