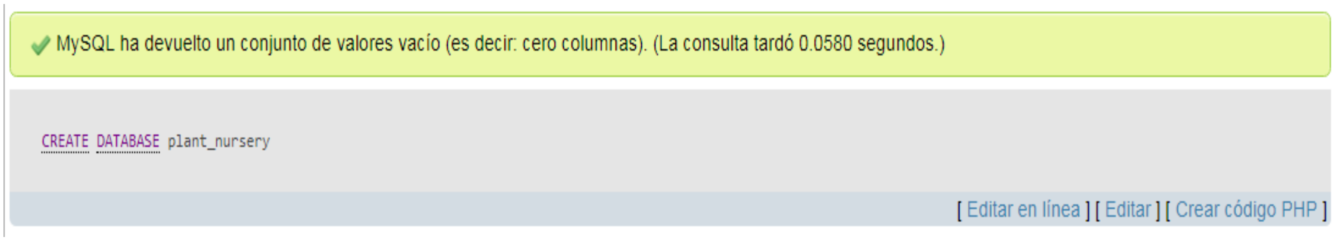
COMMANDS USED

Plant_nursery=dwarded_plants

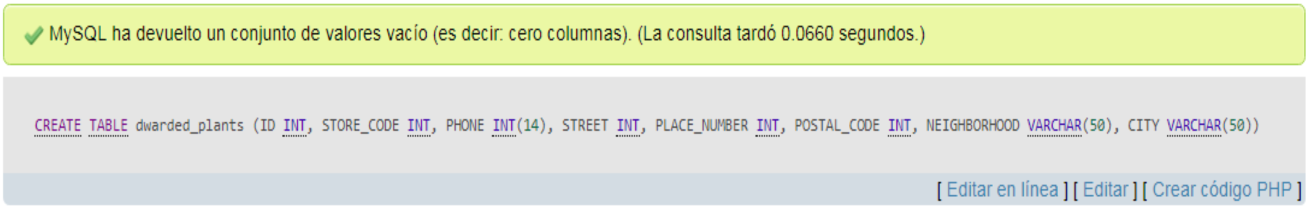
CREATE DATABASE plant_nursery; //These commands create the database



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0580 segundos.)

```
CREATE DATABASE plant_nursery
```

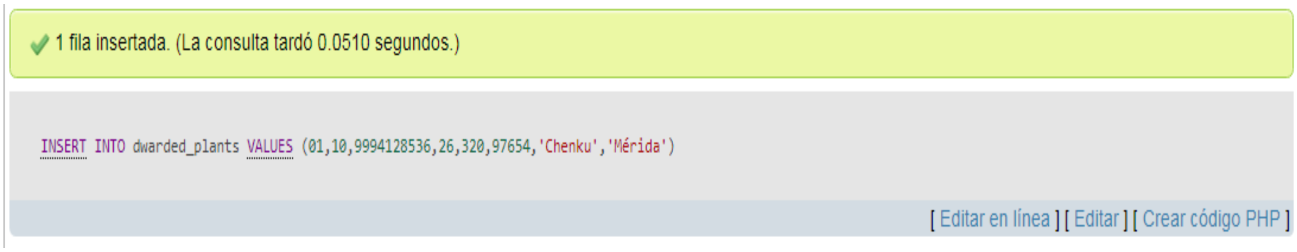
CREATE TABLE dwarded_plants (ID INT, STORE_CODE INT, PHONE INT(14), STREET INT, PLACE_NUMBER INT, POSTAL_CODE INT, NEIGHBORHOOD VARCHAR(50), CITY VARCHAR(50)); //With these commands I create my first column named dwarded_plants, which has all that info.



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0660 segundos.)

```
CREATE TABLE dwarded_plants (ID INT, STORE_CODE INT, PHONE INT(14), STREET INT, PLACE_NUMBER INT, POSTAL_CODE INT, NEIGHBORHOOD VARCHAR(50), CITY VARCHAR(50))
```

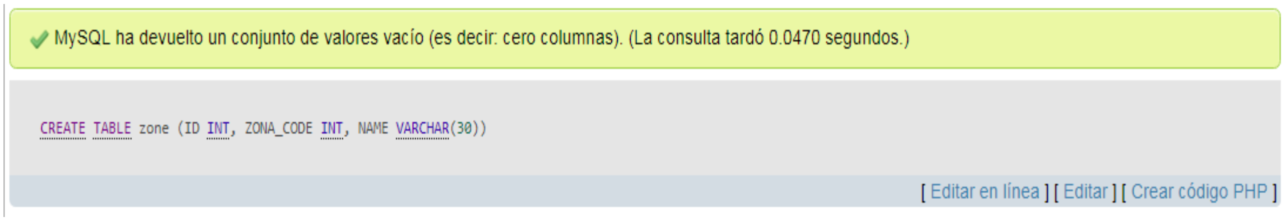
INSERT INTO dwarded_plants VALUES (01,10,9994128536,26,320,97654,'Chenku','Mérida'); //With these commands I add the data.



✓ 1 fila insertada. (La consulta tardó 0.0510 segundos.)

```
INSERT INTO dwarded_plants VALUES (01,10,9994128536,26,320,97654,'Chenku','Mérida')
```

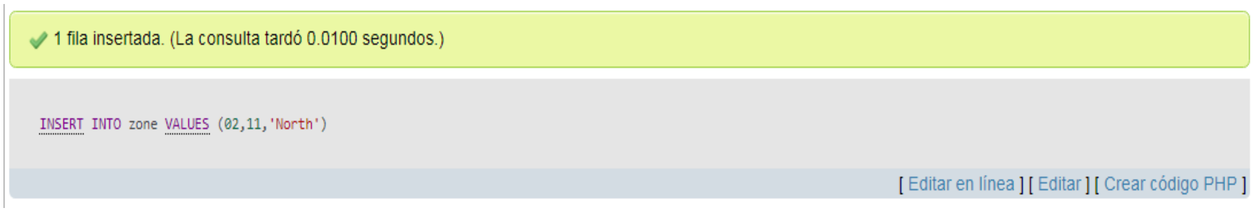
CREATE TABLE zone (ID INT, ZONA_CODE INT, NAME VARCHAR(30)); //With these commands I create my second column named zone, which has all that info.



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0470 segundos.)

```
CREATE TABLE zone (ID INT, ZONA_CODE INT, NAME VARCHAR(30))
```

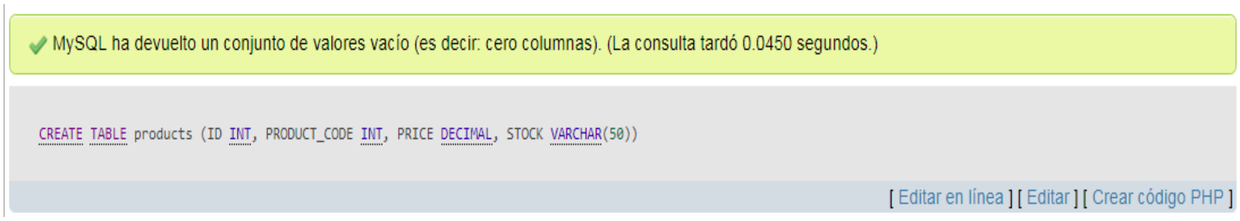
INSERT INTO zone VALUES (02,11,'North'); //With these commands I add the data.



✓ 1 fila insertada. (La consulta tardó 0.0100 segundos.)

```
INSERT INTO zone VALUES (02,11,'North')
```

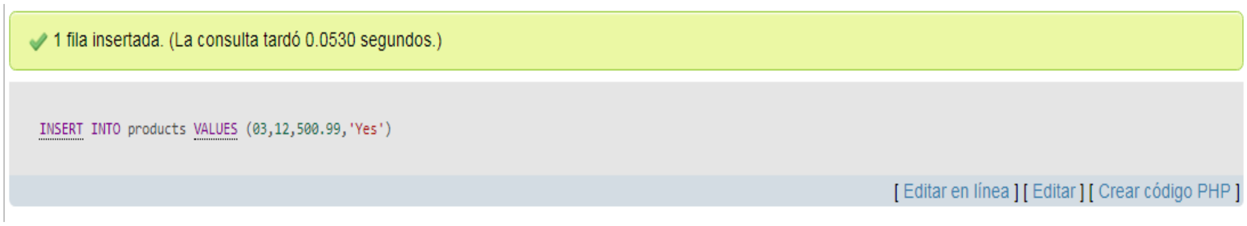
CREATE TABLE products (ID INT, PRODUCT_CODE INT, PRICE DECIMAL, STOCK VARCHAR(50)); //With these commands I create my third column named products, which has all that info.



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0450 segundos.)

```
CREATE TABLE products (ID INT, PRODUCT_CODE INT, PRICE DECIMAL, STOCK VARCHAR(50))
```

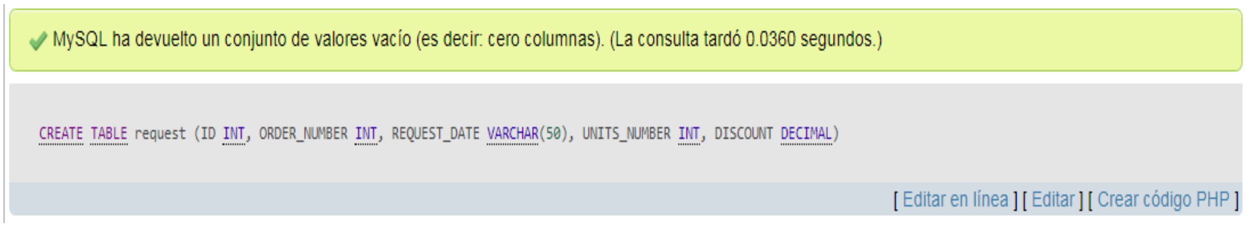
INSERT INTO products VALUES (03,12,500.99,'Yes'); //With these commands I add the data.



✓ 1 fila insertada. (La consulta tardó 0.0530 segundos.)

```
INSERT INTO products VALUES (03,12,500.99,'Yes')
```

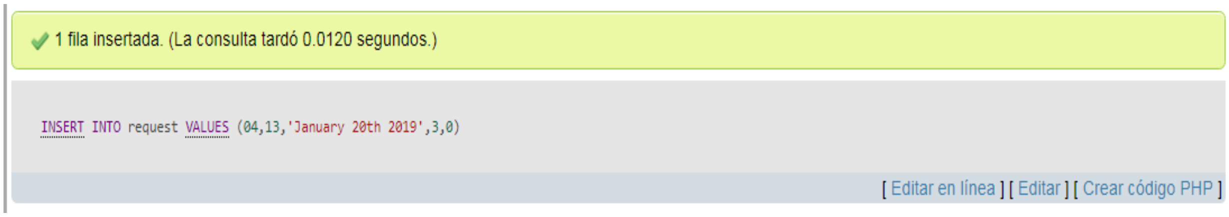
CREATE TABLE request (ID INT, ORDER_NUMBER INT, REQUEST_DATE VARCHAR(50), UNITS_NUMBER INT, DISCOUNT DECIMAL); //With these commands I create my fourth column named request, which has all that info.



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0360 segundos.)

```
CREATE TABLE request (ID INT, ORDER_NUMBER INT, REQUEST_DATE VARCHAR(50), UNITS_NUMBER INT, DISCOUNT DECIMAL)
```

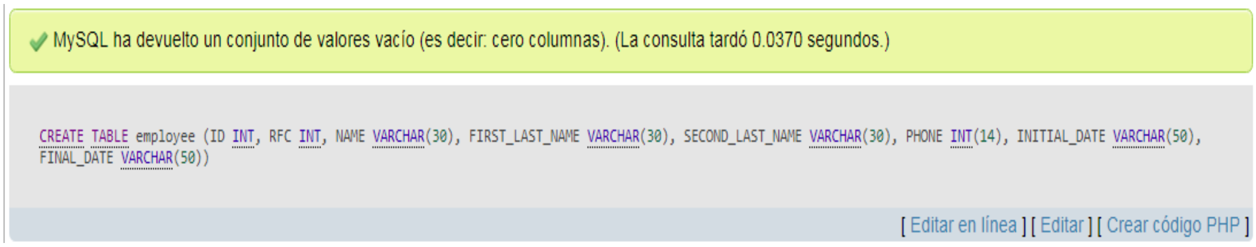
INSERT INTO request VALUES (04,13,'January 20th 2019',3,0); //With these commands I add the data.



✓ 1 fila insertada. (La consulta tardó 0.0120 segundos.)

```
INSERT INTO request VALUES (04,13,'January 20th 2019',3,0)
```

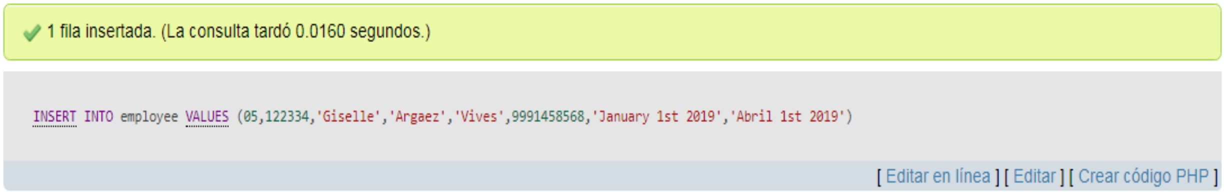
CREATE TABLE employee (ID INT, RFC INT, NAME VARCHAR(30), FIRST_LAST_NAME VARCHAR(30), SECOND_LAST_NAME VARCHAR(30), PHONE INT(14), INITIAL_DATE VARCHAR(50), FINAL_DATE VARCHAR(50)); //With these commands I create my fifth column named employee, which has all that info.



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0370 segundos.)

```
CREATE TABLE employee (ID INT, RFC INT, NAME VARCHAR(30), FIRST_LAST_NAME VARCHAR(30), SECOND_LAST_NAME VARCHAR(30), PHONE INT(14), INITIAL_DATE VARCHAR(50),
FINAL_DATE VARCHAR(50))
```

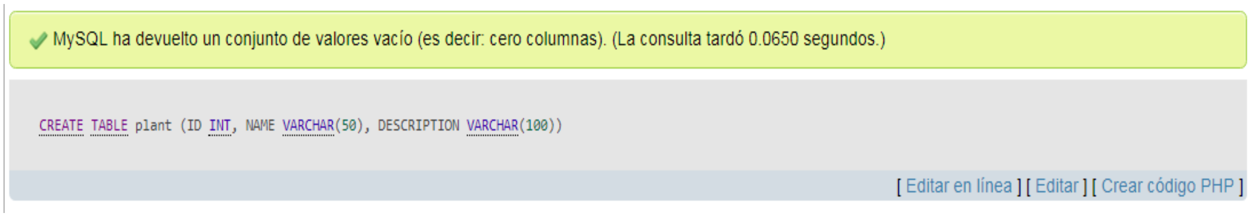
INSERT INTO employee VALUES (05,122334,'Giselle','Argaez','Vives',9991458568,'January 1st 2019','Abril 1st 2019'); //With these commands I add the data.



✓ 1 fila insertada. (La consulta tardó 0.0160 segundos.)

```
INSERT INTO employee VALUES (05,122334,'Giselle','Argaez','Vives',9991458568,'January 1st 2019','Abril 1st 2019')
```

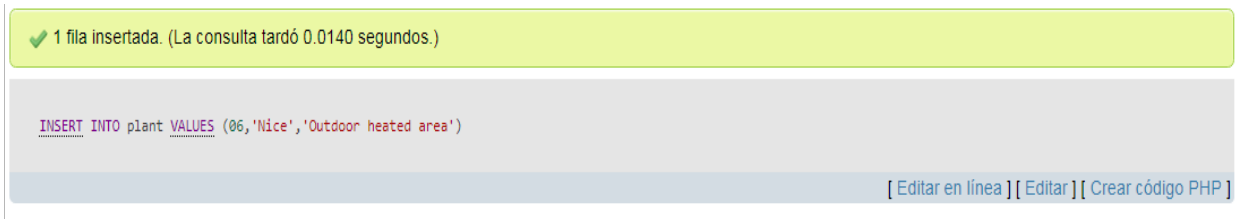
CREATE TABLE plant (ID INT, NAME VARCHAR(50), DESCRIPTION VARCHAR(100)); //With these commands I create my sixth column named plant, which has all that info.



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0650 segundos.)

```
CREATE TABLE plant (ID INT, NAME VARCHAR(50), DESCRIPTION VARCHAR(100))
```

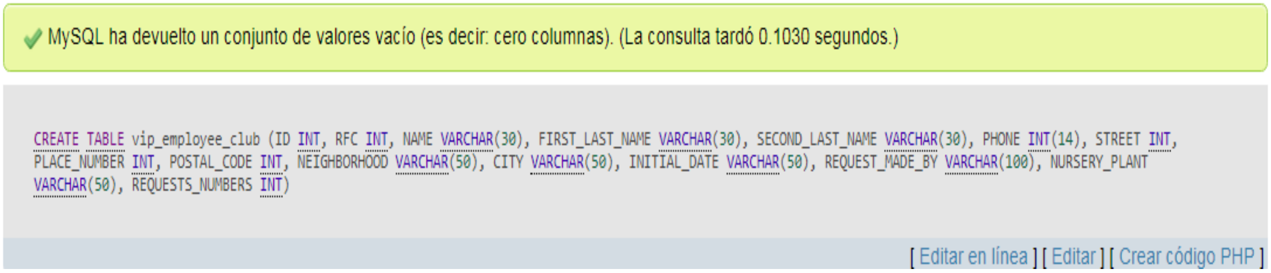
INSERT INTO plant VALUES (06,'Nice','Outdoor heated area'); //With these commands I add the data.



✓ 1 fila insertada. (La consulta tardó 0.0140 segundos.)

```
INSERT INTO plant VALUES (06,'Nice','Outdoor heated area')
```

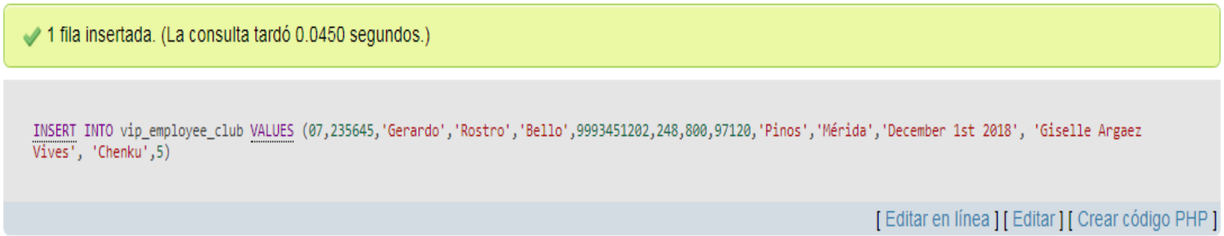
CREATE TABLE vip_employee_club (ID INT, RFC INT, NAME VARCHAR(30), FIRST_LAST_NAME VARCHAR(30), SECOND_LAST_NAME VARCHAR(30), PHONE INT(14), STREET INT, PLACE_NUMBER INT, POSTAL_CODE INT, NEIGHBORHOOD VARCHAR(50), CITY VARCHAR(50), INITIAL_DATE VARCHAR(50), REQUEST_MADE_BY VARCHAR(100), NURSERY_PLANT VARCHAR(50), REQUESTS_NUMBERS INT); //With these commands I create my seventh column named vip_employee_club, which has all that info.



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.1030 segundos.)

```
CREATE TABLE vip_employee_club (ID INT, RFC INT, NAME VARCHAR(30), FIRST_LAST_NAME VARCHAR(30), SECOND_LAST_NAME VARCHAR(30), PHONE INT(14), STREET INT,
PLACE_NUMBER INT, POSTAL_CODE INT, NEIGHBORHOOD VARCHAR(50), CITY VARCHAR(50), INITIAL_DATE VARCHAR(50), REQUEST_MADE_BY VARCHAR(100), NURSERY_PLANT
VARCHAR(50), REQUESTS_NUMBERS INT)
```

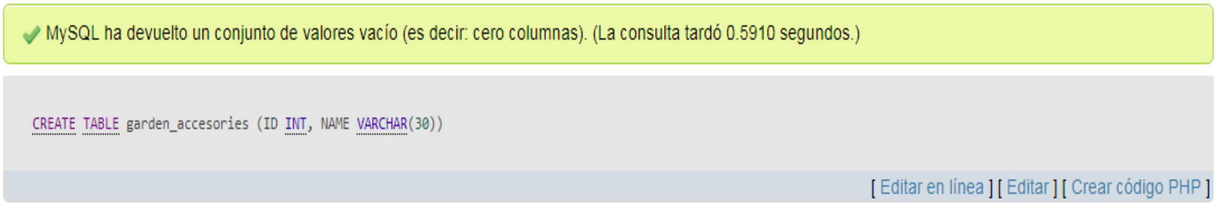
INSERT INTO vip_employee_club VALUES (07,235645,'Gerardo','Rostro','Bello',9993451202,248,800,97120,'Pinos','Mérida','December 1st 2018', 'Giselle Argaez Vives', 'Chenku',5); //With these commands I add the data.



✓ 1 fila insertada. (La consulta tardó 0.0450 segundos.)

```
INSERT INTO vip_employee_club VALUES (07,235645,'Gerardo','Rostro','Bello',9993451202,248,800,97120,'Pinos','Mérida','December 1st 2018', 'Giselle Argaez
Vives', 'Chenku',5)
```

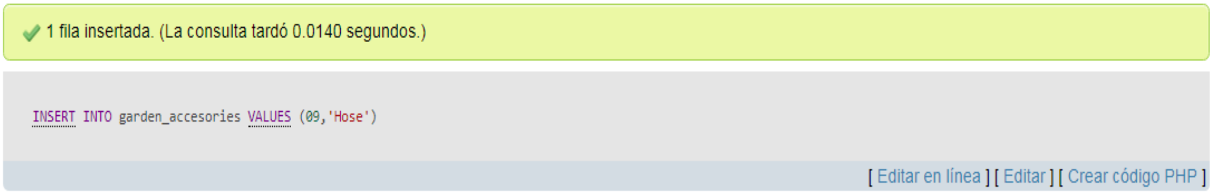
CREATE TABLE garden_accesories (ID INT, NAME VARCHAR(30)); With these commands I create my eighth column named garden_accesories, which has all that info.



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.5910 segundos.)

```
CREATE TABLE garden_accesories (ID INT, NAME VARCHAR(30))
```

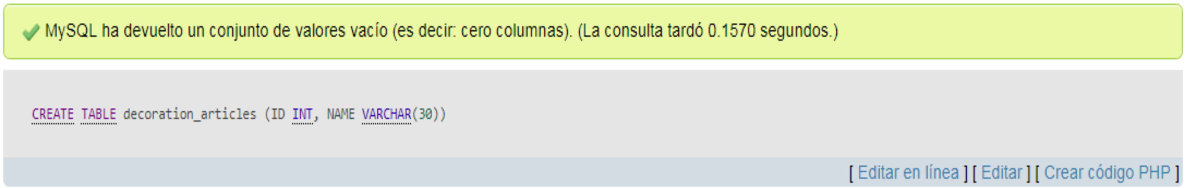
INSERT INTO garden_accesories VALUES (09,'Hose'); //With these commands I add the data.



✓ 1 fila insertada. (La consulta tardó 0.0140 segundos.)

```
INSERT INTO garden_accesories VALUES (09,'Hose')
```

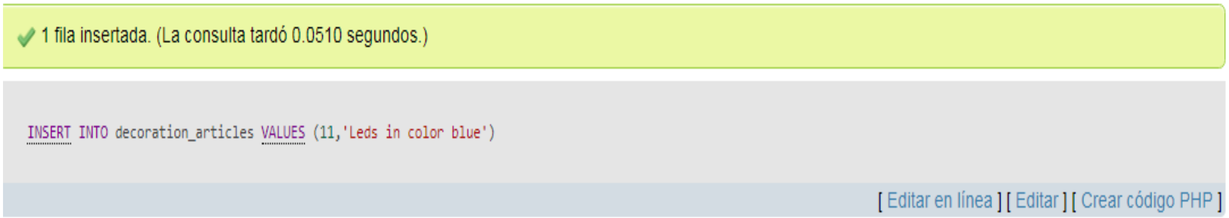
CREATE TABLE decoration_articles (ID INT, NAME VARCHAR(30)); With these commands I create my ninth column named decoration_articles, which has all that info.



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.1570 segundos.)

```
CREATE TABLE decoration_articles (ID INT, NAME VARCHAR(30))
```

INSERT INTO decoration_articles VALUES (11,'Leds in color blue'); //With these commands I add the data.



✓ 1 fila insertada. (La consulta tardó 0.0510 segundos.)

```
INSERT INTO decoration_articles VALUES (11,'Leds in color blue')
```

The types of language that I used in order to make my database are the following:

- The first one is the DDL, which is the one that contains commands such as CREATE, between ALTER, DROP, RENAME and TRUNCATE and this type of language means Data Definition Language, this also means that a database schema is specified by a set of definitions expressed through a special language.
- The second type of language that I used was DML, which means Data Manipulation Language and is a language that allows users to access or manipulate the data organized through the appropriate data model, in this type of language I used a lot the commands INSERT and also this language has other commands such as UPDATE and DELETE.

JUSTIFICATION OF THE CARDINALITY RELATIONSHIP

The dwarded_plants have a cardinality type of many to many with the zone, because there are many dwarded_plants that are located in different zonas. In other words, the dwarded_plants have many zones but a zone has many dwarded_plants.

The zone has a cardinality type of many to many with the products, because as the statement says the products are located in many zones. Otherwise, the zone has many products but the products have many zones.

The products have a cardinality type of many to many with the requests, because when you ask for products they have many requests and vice versa. Otherwise, the products have many requests but the request has many products.

The products also have cardinality types with the plant, garden_accesories and the decoration_articles. In other words, products can have many plants, but plants can have many products. Garden_accesories have many products but products have many garden accessories. Decoration_articles have many products but products can have many decoration_articles.

The dwarded_plants have a cardinality type of one to one with the employee, because a dwarded_plants is in charge of the employee. The dwarded_plants can have only one employee but an employee can have at most one dwarded_plants.

The employee has a cardinality type of one to one with the zone, because the employee can only control one zone at the moment and in a specific time. In other words, the employee can have only one zone but a zone can has at the most one employee.

The employee has a cardinality type of one to many with the request, because the employee can get many requests from the clients. Otherwise, the employee can have many requests but the requests can have at most one employee.

EXPLANATION OF THE ENTITY-RELATIONSHIP DIAGRAM

This is my database of my Project, which I named as plant_nursery. It has 9 entities, which are:

1. Dwarded_plants
2. Zone
3. Products
4. Request

5. Employee
6. Plant
7. Vip_employee_club
8. Garden_accesories
9. Decoration_articles

In this database, the dwarded_plants and the products are located in a zone, the products includes a request. The request is made by the employee and also by the vip_employee_club (which are the clients). The dwarded_plants are in charge of an employee and also the employee is assigned to a zone. The products have a plant, garden_accesories and decoration_articles.

CONCLUSION

We got to the conclusion that in order to make this amazing, beautiful and complex database we spent a good amount of time, because one of the most important and difficult things in order to get this stuff done is the analysis of the problem that is presented at the beginning. Many times, we as data engineers will have a lot of situations like this one, in which, we are going to have to work with clients whose maybe their idea is not very good, or maybe they do not have a clear idea of what they want, so, our task will be to interpret and analyze the given problem.

We also conclude that the most common commands that are used here are CREATE, which belongs to the type of language called DDL (Data Definition Language) and the other command which is the one in which we add or put the data inside our database and that command is: INSERT, which belongs to the type of language DML (Data Manipulation Language).

The other conclusion that we got is that in order to develop a good and better analysis of the situation, we first need and must to use an E-R Diagram (Entity-Relationship Diagram), because when we make this good practice, we can start visualizing in a better way the situation. Despite that also in this part is where we start declaring our entities and after this the attributes, that are one of the important keys in order to make a database. After we got this two important things, we can start thinking in our cardinality type and then start drawing our E-R Diagram and after this process, we can start doing our database with the necessary commands, in this case I used a graphic interface which is called XAMPP, whose also the one that we have been working since the beginning of the quarter.