Costa Rica Institute of Technology

San Carlos Campus

School of Computer Engineering

Databases, group. 50

Second programmed project

“Database of Straightening and painting workshop”

Students:

Raschell Jarquín Quesada

Fabricio Porras Morera

Carlos Solís Mora

Professor:

Efren Jiménez Delgado

Delivery date:

November 23rd, 2022.

Second semester of 2022.

Table of contents:

Executive Summary

The following content of this project, is mainly based on the creation of a database management system, generated from the PostgreSQL programming language. The central focus of the second part of this project consists of the design of a business logic of an application, taking as reference the database created in the first project.

In the following project, the design and the logic of the application will be reflected so that the client is able to understand more clearly the database that he requested, so the work that is carried out is related to what was learned in the course, using the tools provided in the same

Some special features of this database management system that are mentioned in the first part of this project too are that the workshop does not offer a product for sale as such, like spare parts or oil, since after making an evaluation, a price is agreed between the customer and the workshop, and the products are all bought from a supplier. This database management system also explores the solution to create a correct relationship between the different entities, to generate invoices for the different payments that can be made.

Several of the requirements that are requested in this second part, are the various functions to delete, update and insert, which are necessary to be able to modify the tables that exist, clearly along the way, tables will be found which will violate certain situations, for which will have to be solved in a more complex way. And at the same time, it is also necessary to validate that the data that is modified is appropriate, and that it meets certain requirements. This will be resolved with the triggers, which will validate the incoming information in the tables regarding these functions. The development in 3 layers of the application is also extremely important, therefore work will be carried out in an appropriate way so that this is fulfilled in an adequate way, and thus be able to have a clear and concise database.

Objectives

General Objective

To create a business logic of an application for the database created in the first project.

Specific Objectives

To Develop the application in 3 layers: view layer, logical layer and data access layer.

To create a pleasant view for the customer, who can feel familiar with the application.

To carry out all the points that are requested in the project so that they all relate to each other and work in a good way.

To put into practice all the topics seen in the course so that they adapt to what the job requests, as well as other knowledge that arises through extra research and that can be of great help to carry out various points.

Introduction

The following project, besides focusing on the creation of a database for the client's straightening and painting shop, seeks to make the client understand how their information is managed in a simpler and more pleasant way, but without losing its true function for the order in their company, keeping their data secure and constantly updated.

To understand the importance of this project, it is important to know that for a company, having its merchandise in order, its customers, its products that come and go, is something that must always be maintained, if you do not have the proper order, it is very likely that sales are reflected with disorder, and this causes sales to fall and in the worst case, they cannot continue to maintain it and have to close it.

Then, in order for the business managers to be able to make a good management of the trends, losses and profits of the business. Good data integrity is needed to ensure the quality and accuracy of the data. Because data is really valuable when making decisions and if a poor management of the data it’s present. It can cause losses to the company. Based on that the project is focus on how a database can maintain this integrity at the time of receiving data

Additionally, will be displayed on the 'database view' part connecting the database to a graphical interface, where it is necessary to be intuitive for the user and the administrator to avoid any kind of confusion. This GUI is able to create, edit and delete database tables.

Problem Description

In the case of this project, the challenge lies in the correctly carry out of the different necessary elements, to create a simulation of how the correct functioning of the databases would occur in the businesses that use them. In order to achieve this objective, 3 main elements are needed, which are the ones that will generate the application in its entirety, which are: The creation of useful functions for data processing, the creation of triggers to maintain data integrity, and finally, the creation of the graphical interface with its respective functionalities for creating, reading, updating and deleting data (CRUD).

In addition to this, some other functionalities are required to be created for the correct completion of the project: 5 views with at least 6 columns and 2 joins must be generated to confirm the correct functioning of the database; 10 SARGABLE type queries to generate the reports and finally an administrator user (with super user permissions) must be created in the database, a normal user who will only be able to access the tables and functions, and finally, a backup user who can only create database backups.

Finally, it is important to make new schemas within the database to organize the previously created tables in a much more efficient way, and then remove the public schema.

Development

Data processing Functions

For the creation of the data processing functions, it was necessary to know the schema of every table, we know that we only need to make data processing functions for delete, update, and insert, so we need to make these three functions for every table in the schemas of the project.

In the case of insert functions, it was really easy to make these functions because we don´t have to make changes, but for example, if one table has in its data a foreign key of another table, we need to know if the foreign key it doesn´t can be use more than one time in the table, like a relationship of all to all, its necessary to make the relation of the id´s first, but if it can be use more than one time, we don´t have problem to insert the new data.

In the case of update functions, we make the functions with the option to update all the data, except for the id, and like I said in the case of insert, if we have a foreign key that we can´t use more than one time we need to see if there is not one that is not being used in the table, so we can use this one for update the data with this.

For the delete functions, we needed to be more cautious, because in some tables we have foreign keys that are really necessary, and this causes certain rules to be violated, so, in this case we can be use two options: make neutral data, or make a bigger function that removes the data for another tables, to finally remove the data that we want to. But this last maybe makes that the other tables run out of information, because we need foreign keys to make a lot of relationships that are so important for the company.

Data integrity triggers

For the development of the data integrity triggers of the project, it was necessary first at all, analyze what can be done on each table to keep the integrity considering the different types of variables and constraints.

Afterwards, it is created a trigger linked to a unique function that returns a trigger. One trigger is for one table and it is ejected before do a commit of a insertion, update or delete of the columns.

> In connection with the above, when the user try to do a insert, update or delete, the trigger before do the commit, execute the function where the integrity of send data is analyzed and see if eject the commit in case the data received is correct or a rollback otherwise.

In order to explain the functions that return triggers. As mentioned above, it is called in all 3 cases. But the function needs to perform different actions in the case of insert, update or delete. This is why conditionals and triggers procedures are used to separate in blocks when the call is for the specific action.

In the case of insert call, it validates that the insert of data received have logic and integrity with conditionals, for example; that the dates comply with a certain possible range, that some values cannot be less than other sub-values or that some foreign keys exist in the database.

Also, the update call, it is simple because its only function is to verify, by means of the new and old procedures of the triggers, that the values of the new and the old are not the same so as not to perform an identical update and waste executions.

Finally, in the delete call, it checks that the id to be deleted exists in the database so as not to perform an empty deletion.

Graphic interface

For the creation of the graphical interface of the project, the C# programming language will be used, and to work in this totally different environment, you will need to install the "npgsql" functionality that has the C# language built in to be able to access data from a postgreSQL database.

Next, it is needed to create at least 4 main folders where the project files that will make up the database in the C# language will be stored:

The Entity folder will be in charge of creating a class with the same attributes that each of the tables contained in the database have, so that when the values of the different tables are taken, identical entities can be created in the project context generated in C# for the graphical interface.

Next, a folder is also needed that is in charge of storing the different functionalities of creating, reading, updating and deleting for each of the tables in the database, the "model" folder will be in charge of storing these functionalities to be able to use them each time. that the user needs it.

In turn, a "controller" folder will store a file that is responsible for initializing its corresponding "model" file. This for each of the existing tables within the previously created database.

Finally, the most vital folder when it comes to understanding the processes that other folders are involved in is the "view" folder; since it is, after all, where the form-type files with which a given user will really interact will be saved; so, it can be said that the previous folders and files are part of the process and data management, while this folder will be in charge mostly of the visual part and initialize the other components of the other folders. For each of the tables in the database, 2 window forms are needed, one that shows the current values stored in the table that corresponds to the data reading, and the different buttons to create, delete, and update the data. The second form will be used when you want to create a new value within the table or you want to edit any of its values, and it has

spaces so that the user can enter or modify each of the attributes of a specific value in a table.

In addition to this, it is required to create a file capable of generating the connection with the database with which it will work, using the functionalities of "npgsql", this connection requires the name of the database, its corresponding port as well as the password used. This file will be used every time you want to access the data, as well as add new ones and even modify them, so it is essential that it exists since without it you could not work on the database.

Conclusions

It is concluded that the fact of accessing and processing the data from a database, as well as the business logic applied to it, is as important as the visual section with which the user will interact. Since although it is true that programmers usually focus on logic and data management, what users see of the application made must also be understood and for that, programmers must pay attention to how simple or difficult it can be to understand what the accessed data shows, and work based on it to reduce problems as much as possible.

It is also concluded that the creation and modification of the data within a table must be carried out in the most controlled way possible so that its integrity is not affected by the margin of human error that is always present in all kinds of places, without no matter the context or place in the world. For this, the correct use of the triggers is essential, so that if any data has been tried to be modified or entered in an erroneous way, the database is capable of detecting it by itself, and not making that creation or modification, and then notify the user of this error.

Finally, it is concluded that the use of functions that handle the correct creation, reading, updating and deletion of data for each table are extremely important tools for any work carried out in relational databases of any kind. By creating functions that know what the structure of the table is like, its relationships, and that take into account different possibilities such as the elimination of values that are still being referenced and work based on it, they achieve that later, when one of the instructions included in the (CRUD) of the table is going to be executed, you can go to the corresponding function and use it, and with that, save time for the programmers involved in the project, as well as to avoid errors of their part.

Recommendations

It is recommended that when creating a graphical interface for a database of a specific business, it is taken into account that, what the user sees and understands is vital for his judgment of the database as a whole to be good or bad, that is, for example, if the database has a value that is of the NULL type and is one of the attributes of a specific table, the programmer must display messages or values with which the user using the graphical interface is familiar, since values such as the NULL or Boolean types are not usually common for a person who is not a programmer, and the fact of understanding or not what the graphical interface shows, will determine if the delivered product is considered "quality" by the client or business.

Secondly , it is recommended that each time the structure of a table has been established, and its relationships and different restrictions are very clear, the functions that execute the instructions to create, delete, update and read the data entered in them be developed immediately, adapting to its context, and so that in this way, extremely useful tools are available with which the workers involved in that particular project can use them, saving time and effort that can be focused on other things at that time.

Finally, it is recommended that the integrity of the data entered in the tables be kept as clean as possible; for this, triggers must be used, which are capable of detecting different situations or contexts that the programmer or the business considers a violation of data integrity, and once this is done, do not modify or create it and maintain the table clean of junk data. After this, it is important to notify the user that there has been an error of this type, and they must be told what exactly the problem was, so that they can correct it and now whether to make the change correctly.

Bibliography

Ks, P. (s/f). PostgreSQL 9.1: integridad referencial y consultas por uniones. Net.ve. Recuperado el 29 de noviembre de 2022, de https://www.ks7000.net.ve/2015/02/16/postgresql-9-1-integridad-referencial-y-consultas-por-uniones/

Maldonado, S. (2013, noviembre 12). PostgreSQL - Funciones como Triggers. Solvetic. https://www.solvetic.com/tutoriales/article/368-postgresql-funciones-como-triggers/

Multi-column indices. (2012, enero 1). PostgreSQL Documentation. https://www.postgresql.org/docs/7.1/indices-multicolumn.html

PostgreSQL - Operadores. (2020, diciembre 10). Stack. https://isolution.pro/es/t/postgresql/postgresql-operators/postgresql-operadores

Annexes

Annex 1.

Activity number	Date and hour	Place	Participants	Matters to be discussed	Agreements and responsibilities	Pending issues
1	Monday, October 31 at 4:10 p.m.	Instituto Tecnológico de Costa Rica.	Fabricio Porras Morera Carlos Solís Mora Raschell Jarquín	Division of labor	It was agreed that all work individually and advance. Have an advance for the next meet.	An important advance of the corresponding part of the Completed? Yes() No(X)
2	Saturday, November 5 at 6 p.m.	Discord and Whatsapp.	Fabricio Porras Morera Carlos Solís Mora Raschell Jarquin.	Check the advances of each one and reassigning work in case of termination	Fabricio Porras end his work, so another part of the project was assigned. Carlos Solis Raschell Jarquin show the advances of the assigned work for each one.	An important advance of the corresponding part of the Completed? Yes(X) No() Complete the assigned work no later than friday, november 18. Completed?

						Yes() No(X)
3	Friday, November 16 at 4:30 p.m.	Instituto Tecnológico de Costa Rica.	Fabricio Porras, Carlos Solís Mora and Raschell Jarquín.	display completed assignments	there were inconveniences with the work, so we decide to give more time	Complete the assigned work no later than friday, november 18. Completed? Yes() No(X)
4	Monday, November 21 at 8:00 p.m.	Discord and Whatsapp.	Fabricio Porras, Carlos Solís. And Raschell Jarquin	help each other to finish the little work that was left to do	It was agreed that the people in charge Fabricio Porras is going to help Carlos Solis with de GUI and Raschell is going to advance in the written part of the project	Completion of each script of the project and the GUI Completed? Yes (X) No()
5	Friday, November 25 at 5:00 pm	Discord.	Fabricio Porras, Carlos Solis And Raschell Jarquin	Perform tests and verify the correct execution of the project.	It was agreed that before defend the project, all the members are going to do tests	Completion of each technical part of the project

					for check the correct execution of the project	Completed? Yes (X) No()
6	Sunday, November 27 at 4:30 p.m.	Discord and Whatsapp	Fabricio Porras, Carlos Solís Mora, y Raschell Jarquín.	Check the project and end the written part.	It was agreed that those responsible Fabricio Porras, Raschell Jarquín and Carlos Solís, together, would be in charge of finalizing the written part	The total completion of the project Without include written part Completed? Yes (X) No()
7	Monday, November 28 at 8:00 p.m.	Discord and Whatsapp.	Fabricio Porras, Carlos Solís Mora, y Raschell Jarquín.	Separate which part of the project defend each one	It was accorded that all talk in general but in the specific part is in charge the the member that correspon ded to work on it.	The total completion of the project including its written part Completed? Yes (X) No()
8	Tuesday, November 29 at 11:00 A.m.	Zoom	Fabricio Porras, Carlos Solís Mora, y Raschell Jarquín.	Project defense	Those responsible Fabricio Porras, Raschell Jarquín and Carlos Solís defend the	There is nothing pending. Completed? YesX) No()

					completed project to the teacher	