National Autonomous University of Mexico

Engineering Faculty

Databases

Data Masters

Professor:
Ing. Fernando Arreola Franco

Team integrants:

Aguilera Ferrusca Patricio
Damian Isidor Angel
Pérez Morales Daniela
Ramírez Ortiz Alexis Giovanny
Sierra García Mariana
Velázquez Villegas Fabricio

Group:6
Semester: 2026-1

Due date:
December 6, 2025

Contents

1 Introduction

As an introduction to this project is fundamental understand the assignment and the specific needs of the stationery store. In this case our team collaborates with a stationery store chain aiming to innovate the ways to manage all their operative information.

We were hired to design and implement a solution by creating a database using PostgreSQL language. Throughout this document, we will present the complete development cycle of a database, starting from the conceptual and logical modeling to the physic implementation and data visualization.

2 Work Plan

This section of the document describes each part that each team member contributed to this work. The activities to be carried out are divided into three phases: the design phase (proposals for an extended entity-relationship model, relational model, contribution to the document, and presentation), the coding phase (everything related to SQL) and the implementation phase (putting into practice everything developed in the previous phase and implementing the Powervi optional requirement).

Design phase:

Aguilera Ferrusca Patricio, Sierra García Mariana, Pérez Morales Daniela and Velázquez Villegas Fabricio.

Coding phase:

Ramírez Ortiz Alexis Giovanny and Damian Isidor Angel

Aplication phase:

Pérez Morales Daniela.

3 Design

Every database goes through a design phase, there are many different ways to propose a database desing but we got focused on two fundamental types of designs as we considered them to be the best methodologies to represent this stationery store problem.

First we used the Extended Entity Relationship model using the Peter Chen's notation to visualize entities, attributes, relationships and the specialization hierarchy. This type of model allows us to classify products into their disjoint categories while maintaining centralized stock control.

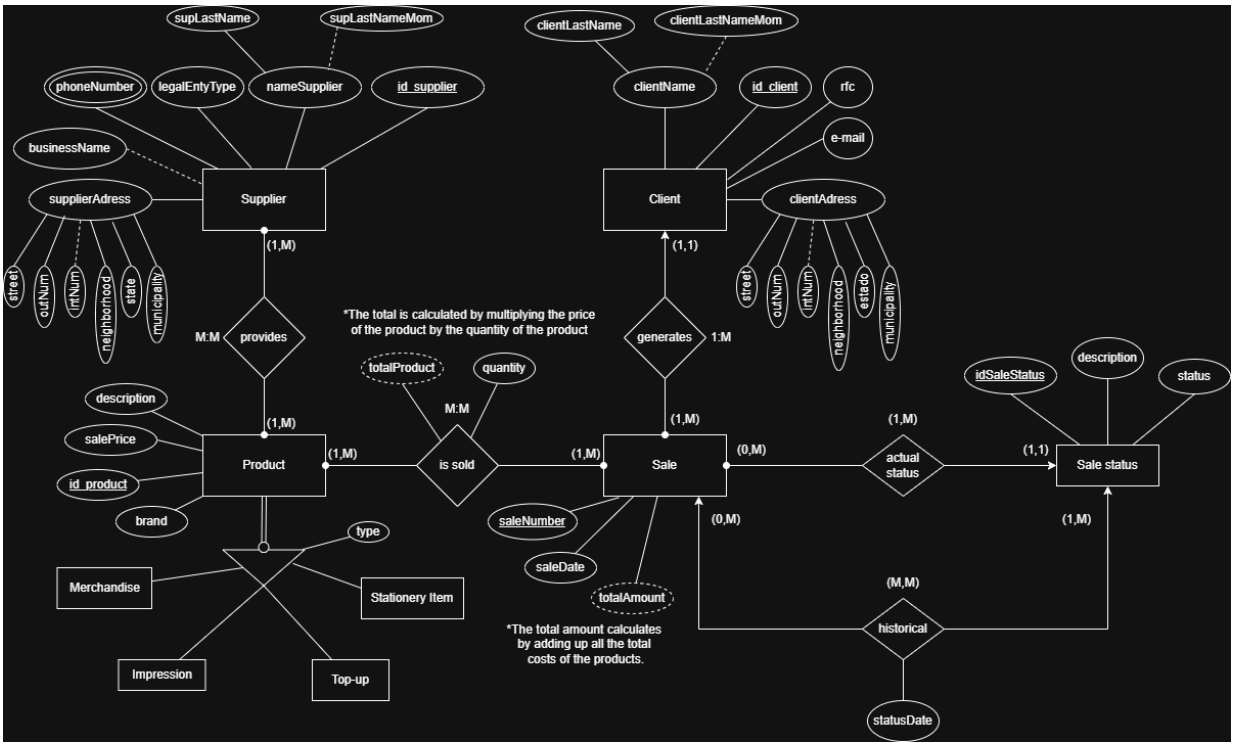

Figure 1: Extended Entity Relationship Model

In this model we express four entities that are key when modeling these requirements: the supplier, the customer, the product and finally the sales. We includes specialization with full participation and historical managment for sales. We conclude that this model satisfies the basic needs of the problem.

Second we developed the Relational Model which is based from the Extended Entity Relationship model. This model introduces many essential concepts such as keys, assocciative tables for M:N relationships, specific data types to represent information and a very important thing, the constraints.

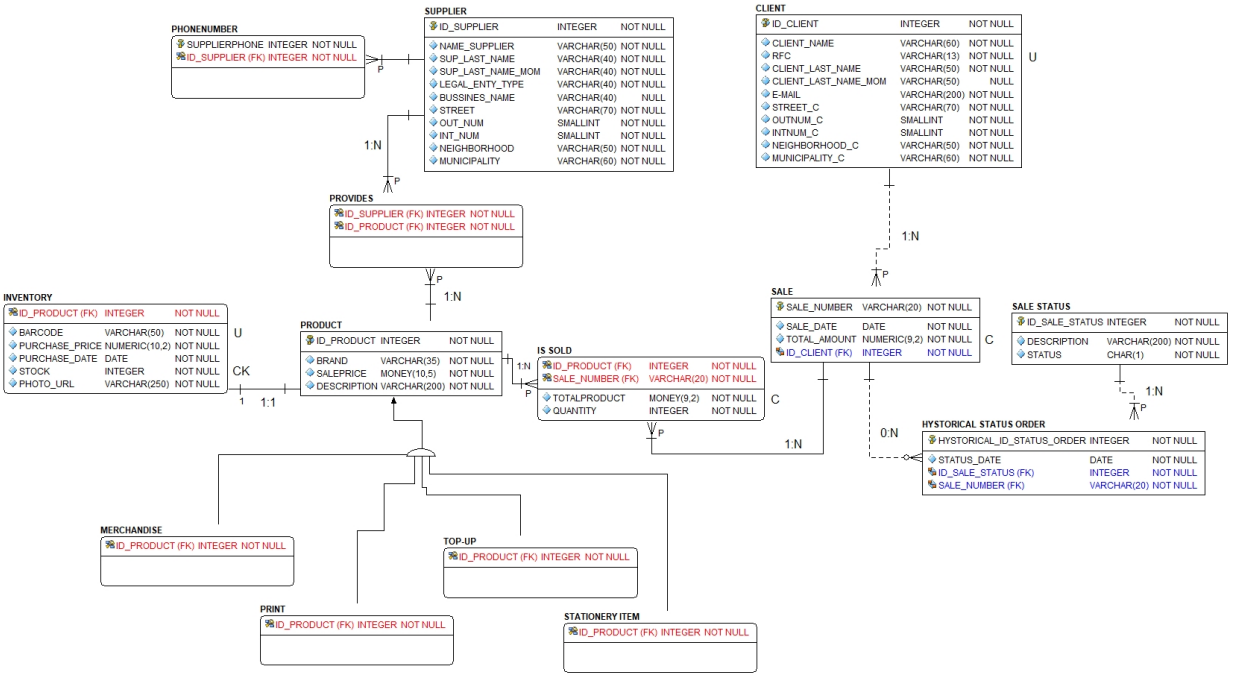

Figure 2: Relational Model

In this model we apply concepts of both foreign and primary keys, using associative tables to represent our M:N relationships, all in crow's feet notation. As can be seen, this model handles some constraint terminology and many data types for our information. We also considered a new table called 'Inventory' in addition to a more modelling of what a history is.

4 Implementation

For this phase we discuss the detailed explanation of what the code implemented as a solution to this problem does.

Main tables and structures

In this part we perform the creation of all the tables that make up the database of our system for this we define entities such as suppliers, clients, products, sales, inventory and specialized categories, each one with its attributes, primary keys and corresponding constraints, we also established the relationships through foreign keys, ensuring the correct connection between sales, clients, products and suppliers. We include the structure to handle supplier phone numbers, the sales status history, the classification of products by type (merchandise, prints, recharges and stationery items) and an inventory table with its own index to optimize the search by barcode.

INSERTS

In this section we implemented the inserts that are used to load initial data into the main tables of our system. These records allow establishing suppliers, clients, products, inventory, sales and their associated lines as well as the specific classification of each product. Their function is to provide sufficient information to put the relational schema into operation, verify the relationships between

tables and check the behavior of integrity rules. It allows evaluating the operation of triggers and the system process, such as inventory control, total calculation and handling of physical products and services. With this data the database can simulate real transactions and validate that the system responds correctly to insertions, updates and deletions.

Part 1.1

We implemented a view that simulates the format of an invoice integrating customer information, sold items, and corresponding amounts per line. It consolidates data from sale, client, is_sold, and product, allowing a complete purchase to be queried through a single logical structure. With this view the system can generate receipts or visual representations of a sale without needing to perform multiple queries.

Part 1.2

This function generates a report that summarizes the total items sold the profit obtained, and the number of sales recorded on a specific date or within an interval. If an end date is not provided the calculation is performed only for the indicated day. To do this it combines information from sale, is_sold, and inventory, applying aggregations that allow obtaining both the sales volume and the profit generated during the evaluated period.

Part 1.3

Trigger 1
The trigger executes shortly before entering is_sold, first it verifies if the product is a service in that case it does not check inventory nor deducts stock so it only calculates the item total. If the product is physical it validates the existence in inventory verifies that there are sufficient units, deducts the quantity, calculates the total and issues an alert when stock drops below three units.

Trigger 2
This trigger executes before updating in is_sold and adapts the inventory according to the product type, in case the change involves services it only avoids stock adjustments since inventory is not managed for these. If there is a change between physical products it returns or deducts units as appropriate, if only the quantity changes it updates the inventory based on the difference. In all cases it recalculates the total and generates an alert if the final stock is less than three units.

Trigger 3
This trigger executes before deleting in "is_sold", if the product is physical it returns to inventory the deleted quantity, if it is a service it does not make adjustments because it has no associated stock. In this way inventory consistency is maintained when deleting items from a sale.

Trigger 4
This trigger executes after inserting, updating, or deleting a record and its purpose is to keep the overall total of each sale updated depending on the operation performed, it identifies the corresponding sale number and recalculates the sum of the subtotals of all associated products. Once the new

total is obtained it updates the sale table ensuring that the final sale amount always correctly reflects the registered ítems, the total calculation remains consistent and requires no additional intervention.

Part 1.4

In this part we implemented a view that directly displays all products whose inventory is below three units. The view contains information from product and inventory presenting the identifier, brand, description, and current stock level. Items that require restocking can be quickly queried.

Part 1.5

This function calculates the profit generated by a product based on its barcode subtracting the purchase price from the sale price, it joins the product and inventory tables and obtains the necessary values. If the code does not exist in the inventory, an exception is generated that prevents incorrect results.

Part 2.1

In this part the v_invoice_lines view is queried filtering by a specific sale number, the view integrates customer information, sold items and the corresponding amounts for each line which allows visualizing a complete sale with a single command. We can see that the view functions as a logical representation of an invoice and that the data related between tables is displayed in a coherent manner.

Part 2.2

In this part the function is executed in two modalities: for a single day and for a date interval. It calculates the total items sold, the profit generated and the number of sales recorded within the requested period, with this the system's capability to generate aggregated reports that allow analyzing sales behavior in different time ranges is validated.

Part 2.3

En esta parte se consulta inicialmente el stock de un producto para conocer su existencia actual, posteriormente se registra una nueva venta y se insertan productos en "is_sold", lo cual activa los triggers encargados de validar el inventario, actualizar subtotales y mantener coherencia en el stock, finalmente se consulta la tabla is_sold para verificar que el registro se haya realizado correctamente.

Part 2.4

Here we can query the view that shows all products in inventory when it is below three units, this view allows immediately identifying items that require restocking facilitating operational stock control and ensuring compliance with the alert rules established in the system.

Part 2.5

In this part we use the barcode_utility function to calculate the profit of a product based on its barcode, we also test the is_service function to know if an item is a service and finally a new record is inserted in inventory with its barcode, purchase price, acquisition date and initial stock.

5 Application

Figure 3: Dashboard

The dashboard that was created with Power BI presents a visual analysis of the stationery store's sales behavior, it was organized through three main charts that allow interpreting in an intuitive way the revenue, profit and investment during the different months analyzed. The first chart "Revenue

by Year, Month and id_product" allows showing the revenue generated by each product throughout the months studied, each bar represents the amount sold per product in October, November and December which allows identifying the items with highest demand and comparing their monthly evolution.

Then we have "Monthly Income" which is a pie chart that summarizes the total percentage of revenue per month, thanks to this representation it is possible to identify which period contributed most to total sales showing clearly the proportion of revenue corresponding to October, November and December, this helps to evaluate the relative weight of each month within the annual total.

And finally we present "Investment and Profit per Month" which compares the investment made and the profit obtained during each month of the year, with this visualization we can observe the

relationship between the acquisition cost of products and the effective benefit generated by sales. The comparative design allows us to easily determine which months were most profitable, which ones presented higher profit margin and in which periods it is necessary to adjust purchase strategies or prices.

6 Conclusions

Aguilera Ferrusca Patricio:
This project was a good exercise for keeping fresh most of the concepts covered in the theory. Defining and outlining the conceptual and logical model for the case study was simple; creating a design for how the information would be stored in a stationery store's database was not very complex. The difficult part, in my judgment, was the programming aspect, defining the constraints set out in the case study; however, the desired result was achieved. Finally, I can say that the project objectives were successfully met.

Damian Isidor Angel:
Throughout this project, I was able to develop and refine my SQL coding skills by building the entire database from scratch. Designing tables, relationships, and constraints helped me understand how each decision affects the behavior of the system. When I started creating the functions and triggers, the real challenges appeared—especially when handling the difference between physical products and services. Physical products require stock, purchase data, and barcodes, while services do not, so I had to design logic that treated them differently without breaking the consistency of the database. It took several attempts and adjustments to get the triggers to update inventory correctly and avoid invalid operations. Even though it was complex, solving these problems helped me understand database behavior at a deeper level.

Pérez Morales Daniela:
The process of this project allowed me to acquire new skills. I learned about using Power BI. Teamwork allowed me to see different perspectives on the problem we had to solve. I was able to put into practice the knowledge acquired in theory. Improve my knowledge and understanding some code concepts. In general, this project allowed me to see a real application of the use of databases. I think that the work developed is very important, since we provided a solution to a real scenario.

In some parts of the design I had difficulties assimilating what was required storing stock and carrying out an inventory could be understood in many ways. After exchanging ideas we arrived at a common solution that meets the request.

For the design of the presentation it was interesting and fun to give it a sales focus. Seeing it this way we can experience what it will be like in the world of work, when we have to sell our work.

Ramírez Ortiz Alexis Giovanny:
Working on this project taught me not only SQL, but also how to approach problems that appear during implementation. Writing the code for the tables was straightforward, but creating functions and triggers brought new difficulties, especially when defining the rules for physical products and services. I had to carefully think about how the system should react when a sale involved stock, or when a product classified as a service should not be allowed to have inventory. These situations required debugging, rewriting logic, and testing multiple cases to make sure everything worked cor-

rectly. Even with the challenges, the process helped me improve my reasoning and problem-solving skills, and as a team we learned how to communicate and coordinate our work to deliver a complete and functional database system.

Sierra García Mariana:
This project allowed me to reinforce my skills in database design with all the theory seen in classes, especially when building the extended entity-relationship model and its respective relational model, the design phase was fundamental, since it involved analyzing in detail the requirements of the case and making decisions about entities, relationships, keys, constraints and the general structure of the system.

Although the implementation also presented difficulties, I consider that the design was what allowed the rest of the project to advance correctly understanding that it is fundamental, completing this project helped me better understand the importance of planning the structure of a database before programming it and develop a more complete criterion to identify the real needs of a system.

Velázquez Villegas Fabricio:
This project has been one of the most time-consuming and labor-intensive ones I had to work on, as the concept of databases is still a quite new to me. Some of the challenges I faced doing this project were in the design phase, the models specially the relational model, as in some cases I was confused when propagating keys between entities, so I asked for constant feedback to my work team. I also encountered some challenges in using SQL, as I was quite accustomed to using SQLPlus. On the other hand, when it came to giving the presentation and contributing to the document, it was easy, but there were exceptions such as the grammatical use of English and how to make sense of the slides. Those were the most significant challenges when putting all together.