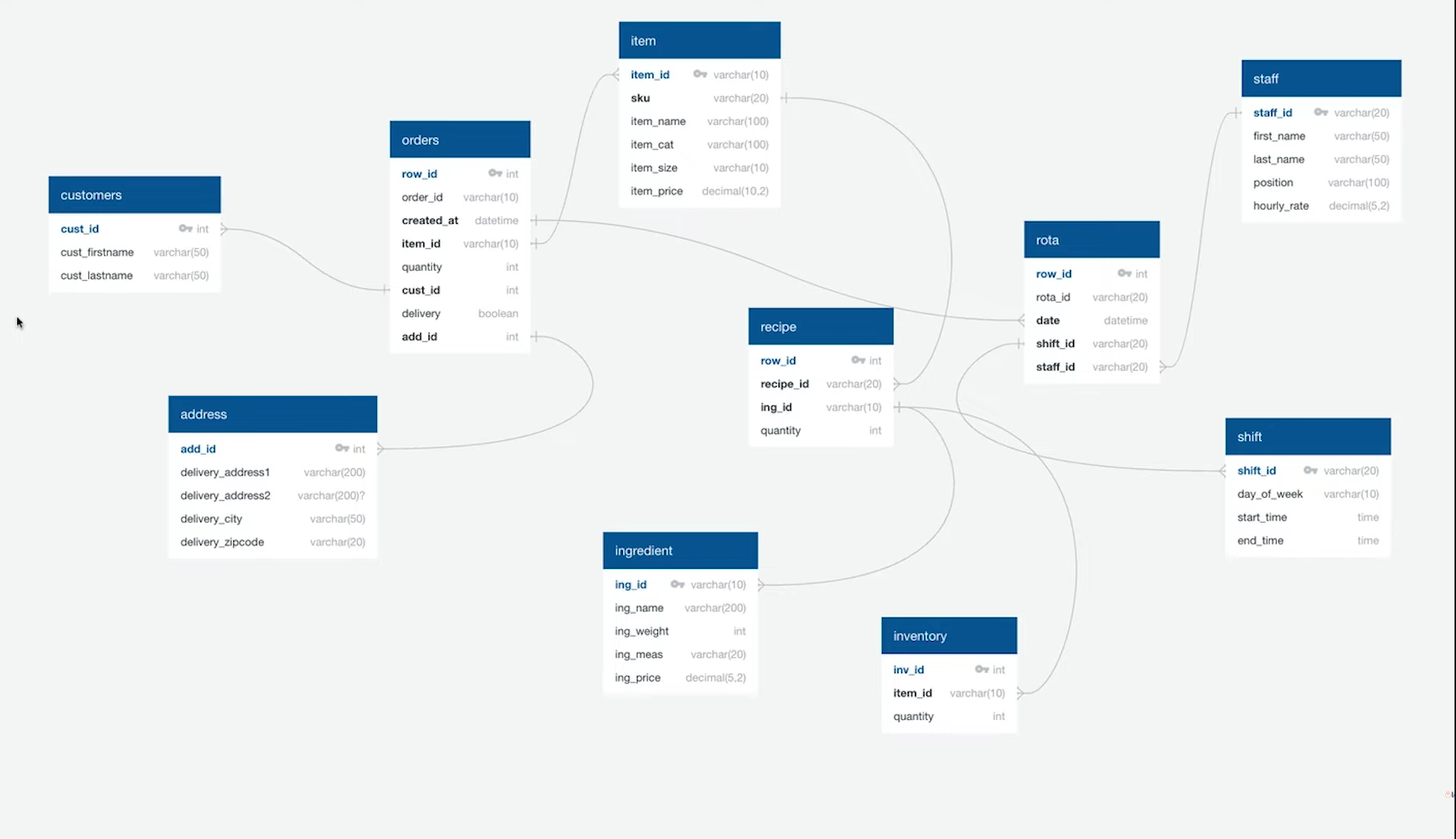
**By: Alonso Calderón Peña  
Project description**The client is opening a new Pizzeria and orders me to build a relational database in order to store the information, focusing on the following main areas:  
  
1.-Customer Orders  
2.- Stock Control  
3.- Staff

This relational database will be then used as a base to generate interactive dashboards that the client will be able to regularly monitor to adjust the staff, re-stock products or run marketing campaigns to specific areas. This project utilized the following tools:  
  
1.-Microsoft Excel  
2.-MySQL  
3.-Navicat  
4.-Google Looker studio  
5.-QuickDBD

**Data Required by the client**Orders:  
- Item name  
-Item price  
-Quantity  
-Customer name  
-Delivery address  
  
Based on these requirements, the client provides a rudimentary table which they have been using to store data, shown in the following screenshot:  

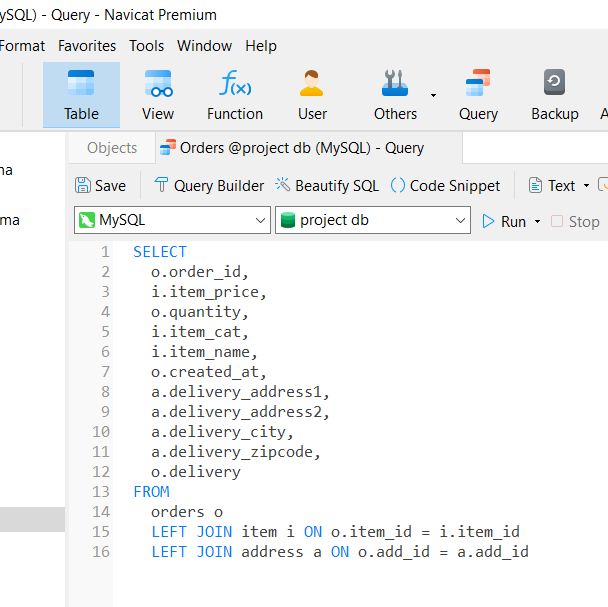

Taking a look at the table we can clearly notice some mistakes, the main one being that the data has many instances of redundance and repetition, hence, the first step to construct our relational data base will be to **normalize** the data.

Due to the scope of this project, not every step of the data normalization process will be explained, however, to provide the logic behind my thought process I will explain briefly on how I started this:  
First we can clearly see the name of the customer is repeated throughout the table, therefore we need to create a separate table for customers, and relate it to our order table using primary and foreign keys. Then I would need another separate table for the addresses, since they are also redundant in the original table. This thought process was applied throughout to generate the following relational database shown:

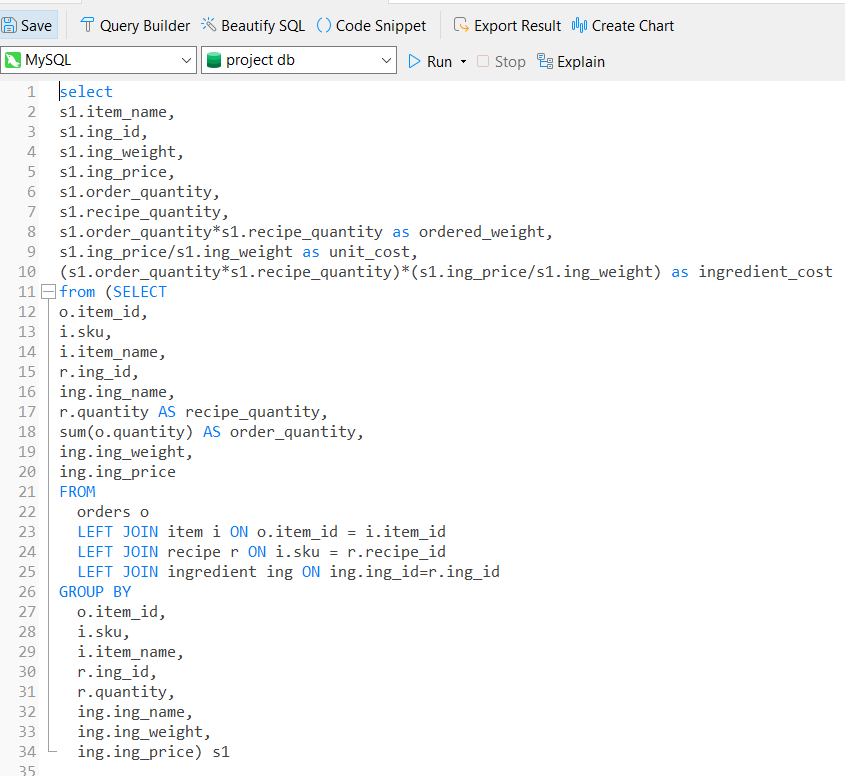


This model was created using the online tool quickDBD, that allows you to create a visualization scheme from SQL code in your browser and export it to different programs, like Navicat, which is the one I will be using for this project. The code is also included in the attached files.  
This relational database model was created considering the client needs and the information provided by them, like the raw initial table and a menu. Examples of the tables included are ingredients, recipes, staff information, rotation, among others.  
  
**Data preparation for Looker studio using SQL queries**

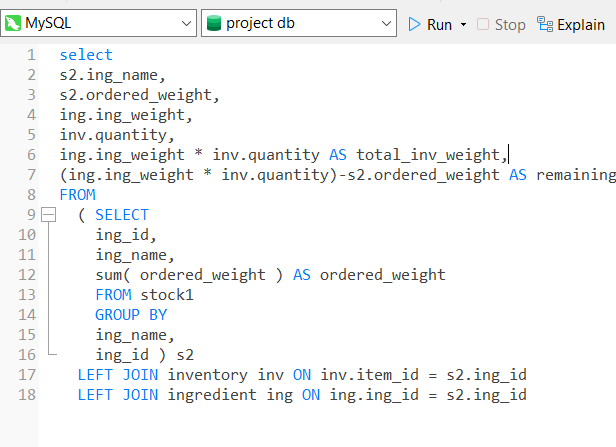
In the next section I will provide the queries I used to prepare the data for looker studio and explain my reasoning and thought process behind each one.  
In order to do effective and efficient queries we need to consider the clients needs and requirements, as well as the general layout of our dashboard and what information we are going to include in it.  
  
**One of the main needs of our client is monitoring the order activity**, in order to do that I considered that we may need a dashboard with the following data:  
1.-Total orders  
2.-Total sales  
3.-Total items  
4.-Average order value  
5.-Sales by category  
6.-Top selling items  
7.-Orders by hour  
8.-Sales by hour  
9.-Orders by address  
10.-Orders by delivery/pick-up

To extract this information from the database I used the simple following query on navicat:  
  
I did a left join in orders and extracted the necessary information from both the address and items tables, as we want to know where the order went and the items ordered, this will allow the client to explore which items are working better and in which areas of the city they should focus on. I used aliases to simplify the code.  
  
**Another need of our client is to monitor the inventory.** In order to do that I needed to construct a dashboard with the following information:  
1.-Total quantity by ingredient  
2.-Total cost of ingredients  
3.-Calculated cost of pizza  
4.-Percentage of stock remaining by ingredient  
5.-List of ingredients to reorder based on current inventory

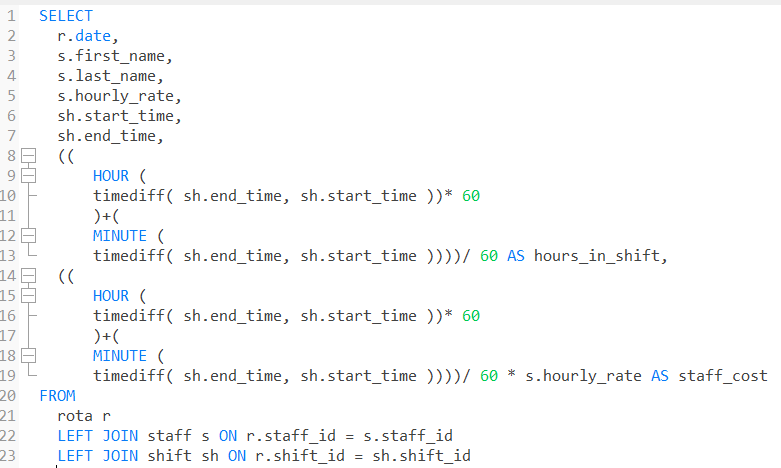
In order to extract this information just one query was not enough, I required the use of sub-queries and views, the SQL code is shown and explained below:



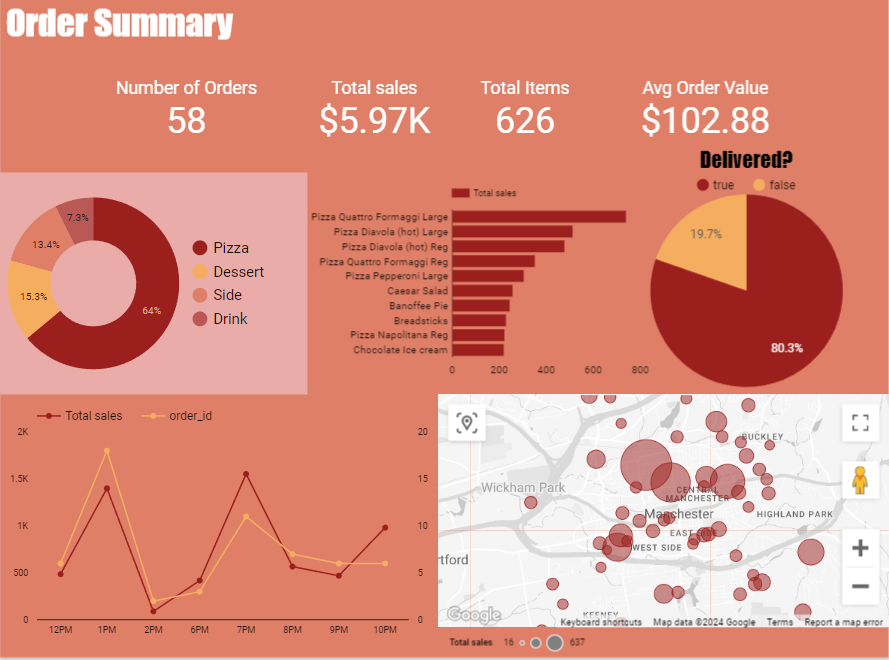
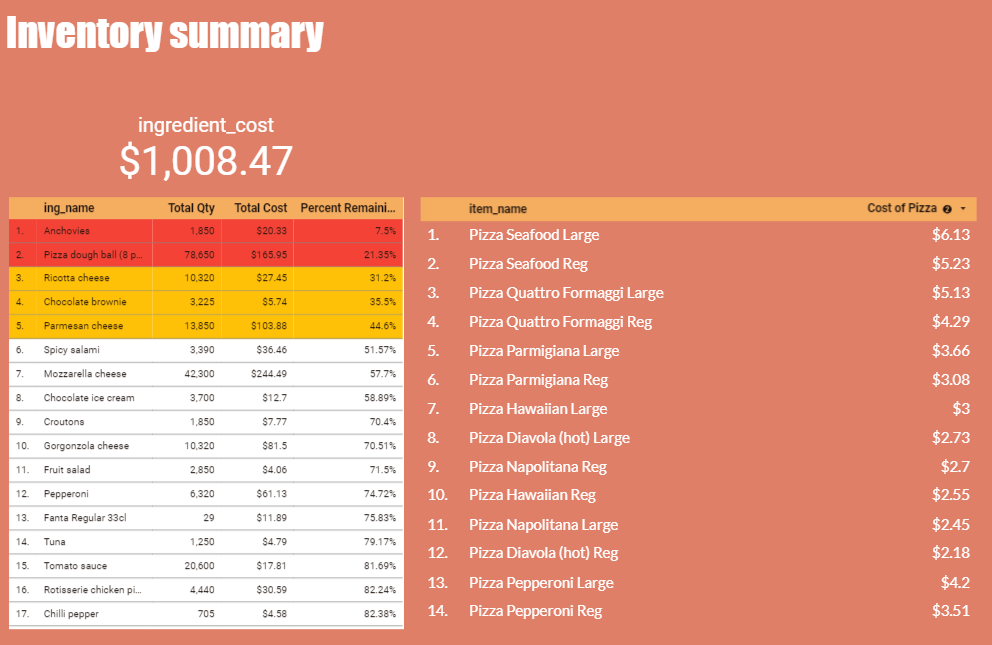
To extract the first 3 items of information I used this query and a sub-query. In this code I had to group by each pizza and obtain the ingredients contained in each from the recipe table, then I grouped and segmented even further by ingredient to obtain the quantity used of each one, then I had to multiply by order quantity, however, as the ordered quantity was already an aggregated field, I had to use a sub-query as a method to allow me to calculate ordered weight and unit cost using the ordered quantity aggregation, therefore avoiding the use of a view. Finally I just had to divide the ordered weight by unit cost to obtain the cost of each ingredient.

To obtain the remaining fields I formulated the following query:  


In order to construct this query I saved my last query as a view, I did this because I found it simpler to just create and additional query and work based on what I already had to the alternative of obtaining all the information in a single query; sometimes adding another query to the process may seem inefficient or an inability of the programmer, but in this case adding an additional query has minimal costs and saves a lot of time that could have been wasted trying to create a single very complex query that did all the work. I created another subquery because I also aggregated the ordered weight, as in the previous example, and I needed that field to calculate the remaining weight of each ingredient in the inventory.

However, I still needed to consider one more cost in order to adequately estimate the cost of producing each pizza, which is the staff cost, to do this I used the following query:  
  


As the client did not give me the duration of each shift and only provided start and end hours of the staff members, the most complicated part of this query becomes calculating the amount of hours of each shift adequately, this can be done aggregating both the hour difference and the minute difference of the starting and ending hours of the shifts, if we divide that by 60, we can obtain the number of hours, by multiplying this by the hourly rate of each staff member, we can successfully obtain the staff cost.

Finally, all I needed to do next was to export the queries into .csv files to upload them to google looker studio to create the dashboards, images of the dashboards are shown below but also included as PDF in the folder and the link is also found below:  
  
  
  
(The table is shown in contrasting white due to the conditional formatting of the inventory levels)



LINK TO THE DASHBOARD: <https://lookerstudio.google.com/reporting/eccc9c73-1b2b-4ab3-b74f-2048e03edbd7>