Nicolas Gonzalez

SQL Project

**1.**

Entity Relationship Diagram

Graphical user interface

Description automatically generated with low confidence

**2.1.**

Which 5 employees should the company offer a bonus to?

The query that I used to solve this question is the following:

SELECT EMP\_FNAME,EMP\_LNAME ,LGEMPLOYEE.EMP\_NUM , EMP\_HIREDATE AS HIRE\_DATE, DEPT\_NAME, SUM(INV\_TOTAL) AS TOTAL\_SALES, COUNT(INV\_TOTAL) AS NUMBER\_SALES , AVG(INV\_TOTAL) AS AVERAGE\_PRICE\_PER\_SALE

FROM LGINVOICE

JOIN LGEMPLOYEE

ON LGINVOICE.EMPLOYEE\_ID = LGEMPLOYEE.EMP\_NUM

JOIN LGDEPARTMENT

ON LGEMPLOYEE.DEPT\_NUM = LGDEPARTMENT.DEPT\_NUM

WHERE INV\_DATE BETWEEN '2015-01-16' AND '2016-01-16'

GROUP BY LGEMPLOYEE.EMP\_NUM

ORDER BY SUM(INV\_TOTAL) DESC LIMIT 5;

With this query, in the beginning, I selected the data that I wanted to display in the new table. First, I wrote EMP\_FNAME and EMP\_LNAME to know the employee’s first and last name to know the personal information of the person in the table. After that, I wrote LGEMPLOYEE.EMP\_NUM to know the number of each employee which is retrieved from the LGEMPLOYEE table. Then, I selected EMP\_HIREDATE to know when the employee started working with the company since making big sales with more experience is easy for the employees, but it is also a disadvantage because the employees get used to the monotony in the workplace, so have a great volume of sales it is a symptom of the employee’s motivation. Then, I selected the DEPT\_NAME to know in which department is working each employee. After that, SUM(INV\_TOTAL) to know the total bill of sales and I name the column Total Sales. Then, I wrote COUNT(INV\_TOTAL) to know how many items the person sold during the year. The next one is AVG(INV\_TOTAL) to know the average price per sale, and the column is called Average price per sale. Then, how the data are from different tables, I had to join the tables to create the new one, and for that I followed the query with a multijoin, first I selected the LGINOVICE and LGEMPLOYEE to join with the attribute of EMPLOYEE\_ID in LGINVOICE table which is equal to the EMP\_NUM in LGEMPLOYEE table. The second join was to join LGEMPLOYEE and LGDEPARTMENT with DEPT\_NUM. Then, I wrote WHERE INV\_DATE BETWEEN '2015-01-16' AND '2016-01-16' to select only the sales that were closed between these dates. To end my query, I selected GROUP BY LGEMPLOYEE.EMP\_NUM to display the total of sales per employee number, and finally, I wrote ORDER BY SUM(INV\_TOTAL) DESC LIMIT 5; to order the table with the biggest number of total sales at the top, and limit to the top 5 sellers within the company.

A screenshot of a computer

Description automatically generated with medium confidence

In this table, we are seeing that Herb Manning, Gwen Easley, Rosalind Villarreal, Linnie Goldman, and Diego Erwin are the top 5 sellers in the company. Herb Manning who was hired on 04/13/1997 with an employee number of 83677, and who is working in the sales department made $262019.30 in sales, he sold 115 items with an average price of $226.25 per item. Also in the query, I have written that the sales have been made in the period from 01-16-2015 to 01-16-2016, which is the last day for which we have data in the LGINOVICE table. The table shows the same information for the top 5 sellers, and according to that data, they deserve a bonus. The company should protect its best sellers, but not only that, with that bonus, the company as I already said, is implementing a competitive environment which is going to make an improvement in the number of sales. In addition, they have demonstrated that they can sell a big quantity of items, so the company should give them the items with more profit for the company.

**2.2.**

What is the company's worst product they sell?

My answer to this question is the following:

SELECT LGLINE.PROD\_SKU, SUM(LINE\_QTY) AS TOTAL\_OF\_SALES , PROD\_PRICE,kz SUM(LINE\_QTY) \* PROD\_PRICE AS TOTAL\_SALES

FROM LGLINE

JOIN LGPRODUCT

ON LGLINE.PROD\_SKU = LGPRODUCT.PROD\_SKU

GROUP BY LGLINE.PROD\_SKU

ORDER BY SUM(LINE\_QTY) \* PROD\_PRICE ASC LIMIT 1;

With this query I wanted to answer the question, First I selected all the data needed in order to display the table LGLINE.PROD\_SKU, SUM(LINE\_QTY) AS TOTAL\_OF\_SALES these two columns are retrieved from the LGLINE table after PROD\_PRICE is retrieved from LGPRODUCT table, then SUM(LINE\_QTY) \* PROD\_PRICE AS TOTAL­\_SALES is retrieved from LGLINE table, and is the multiplication of SUM(LINE\_QTY) times PROD\_PRICE and named as TOTAL\_SALES. You need to know that I used ‘AS’ to name the column as I wanted to do it. For example, I did not want to name the column SUM(LINE\_QTY) \* PROD\_PRICE and for that reason, I used AS to name the table TOTAL\_SALES. After selecting what I want to have at the table, was time to join the tables through the same types of data. The join was FROM LGPRODUCT JOIN LGLINE ON LGLINE.PROD\_SKU = LGPRODUCT.PROD\_SKU uses the PROD\_SKU as a connector, and as you can see, they do not have to be the same names to be able to join, they only have to have the same type, which in this case is Varchar (15). Finally, I selected GROUP BY LGLINE.PROD\_SKU to know the identifier of the product, and ORDER BY SUM(LINE.QTY) ASC LIMIT; is used to order the table by the number of times that the product has been sold. The worst product in the company in my opinion is the product that has fewer total sales in dollars made. Also, the worst product in the company could be the product with the smallest number of sales, but there is a problem with these tables, and it is that we do not know the cost of producing the product, therefore with only the knowledge about the price of the product, my decision was choosing the product with the lower total sales in dollars.

Timeline

Description automatically generated

The table shows us the following data:

In the first column, we can see that the PROD-SKU is ‘8937-WVA’, in the next column we have the TOTAL\_OF\_SALES which is the total number of sales per the PROD\_SKU ‘8937-WVA’. The next column is the PROD\_PRICE which is the price of the product. Finally, the TOTAL\_SALES which is a multiplication between the total of sales and the price of the product makes a total of $355.32 dollars. Thank to this data, we can say that the worst product of the company is the product identified as ‘8973-VWA’ since the total sales are the lowest in the company. As I already said, we do not know the marginal profit of each product, and without that information, my answer to the question is this one. The company should improve the sales of this product with the best sellers within the company, but sometimes it is not enough to make a product become successful, for that reason, if the product is not successful, maybe the company should retire it from the product portfolio.

**2.3.**

Who are the 10 best customers?

The query that best answers the question is the following:

SELECT LGCUSTOMER.CUST\_CODE, CUST\_FNAME AS FIRST\_NAME, CUST\_LNAME AS SECOND\_NAME, SUM(INV\_TOTAL) AS TOTAL\_PURCHASES, COUNT(LGCUSTOMER.CUST\_CODE) AS FIDELITY

FROM LGCUSTOMER

JOIN LGINVOICE

ON LGCUSTOMER.CUST\_CODE = LGINVOICE.CUST\_CODE

GROUP BY LGCUSTOMER.CUST\_CODE

ORDER BY COUNT(LGCUSTOMER.CUST\_CODE) DESC, SUM(INV\_TOTAL) DESC

LIMIT 10;

First, I wanted to select the data to display in the new table and to do that I wrote SELECT LGCUSTOMER.CUST\_CODE, CUST\_FNAME AS FIRST\_NAME, CUST\_LNAME AS SECOND\_NAME from LGCUSTOMER table, since when the field (CUST\_CODE) is the same in more than one table, you must write first the table from which you wanted to retrieve it. Then, from LGINVOICE table I selected SUM(INV\_TOTAL) AS TOTAL\_PURCHASES because when you write SUM () the table displays the sum of all the data. To end the selection, I wrote COUNT(LGCUSTOMER.CUST\_CODE) AS FIDELITY to count how many times the customer code appeared in the table to know how many purchases the customer did in our company, therefore you know who was the most loyal customer. COUNT () is a function that counts the data, and if one data appears 10 times, you have a 10 in your table. For example, if you are a professor and you want to know class attendance, you can count the attendance of your students.

After the selection, I did the join which combines data from two different tables and display the information that you want. FROM LGCUSTOMER JOIN LGINOVICE ON LGCUSTOMER.CUST\_CODE = LGINVOICE.CUST\_CODE is the join from LGCUSTOMER table join LGINVOICE table since the field CUST\_CODE is the same in both tables, and they are connected between them. For that reason, as I already said at the beginning, sometimes you have to say from which table you want to retrieve the data and not only the field since when the field is the same you must clarify everything. To end the query, I wrote ORDER BY COUNT(LGCUSTOMER.CUST\_CODE), SUM(INV\_TOTAL) DESC LIMIT 10; since ORDER BY is used to order the table with your preference, and I ordered like the table because I wanted to have the most loyal customer at the top, with the most times of purchases, and I also put the second order to differentiate two customers with the same number in the FIDELITY column. LIMIT 10 is used to say to the table that I only want to display 10 rows.

This query is the best option to question this question since a retail company needs to keep loyal customers in order to try to improve its profit and try to keep the sales more secure. You can think that the company can use the total purchases as it indicator to know who was the best customer, but you have to think that the best data for a company is the lineal data, I want to say that the company would rather have a customer who is loyal and buy the different product during one year than a customer who buys the same than the customer 1 but only in one time since the customer 2 the next year is going to change the company if he finds better prices. For that reason, this query is the best solution to display the new table.

A screenshot of a computer

Description automatically generated with low confidence

The table shows the information that I wanted to display, and now I am going to explain it.

The first column shows the CUT\_CODE which is the customer code since all the customers have their own code. Then, we have two columns FIRST\_NAME and SECOND\_NAME to know the information about the customer. After, we can the TOTAL\_PURCHASES to know the quantity in dollars of the customer has been purchased in the last year, but as I already said, this data is not the most important for the company. Finally, the FIDELITY column shows the number of times that the customer made a purchase in the year, which is the most important data for the company since it shows fidelity. When the FIDELITY columns are the same quantity, we chose the second decision which is the TOTAL\_PURCHASES, but only when the times of purchases are the same.

After seeing the table, I confirm my idea of the best customers for the company are those who make the most number of purchases, since they show their loyalty to the company, also the company has a less volatile customer portfolio which makes more secure company’s future.   
I believe that the company should implement an idea or process where the customer with several purchases in a year would have a discount, for example, a company card. The card will accumulate points with different purchases, and with more money spent in the company, the customer would have more points and different kinds of discounts. With this implementation, the company should engage more loyalty in their customers which in the future would be key to the company’s success.

**2.4.**

What is your top suggestion for decreasing costs for the company?

SELECT LGEMPLOYEE.EMP\_NUM, EMP\_TITLE AS ORGANIZATION\_STATUS, LGDEPARTMENT.DEPT\_NUM, DEPT\_NAME, EMP\_COMM AS COMMISSION, SUM(INV\_TOTAL) AS TOTAL\_SALES, MAX(SAL\_FROM) AS LAST\_CONTRACT, SAL\_AMOUNT AS SALARY

FROM LGEMPLOYEE

JOIN LGDEPARTMENT

ON LGEMPLOYEE.DEPT\_NUM = LGDEPARTMENT.DEPT\_NUM

JOIN LGINVOICE

ON LGEMPLOYEE.EMP\_NUM = LGINVOICE.EMPLOYEE\_ID

JOIN LGSALARY\_HISTORY

ON LGEMPLOYEE.EMP\_NUM = LGSALARY\_HISTORY.EMP\_NUM

WHERE EMP\_COMM > 0 AND INV\_DATE LIKE '2015%' AND SAL\_END IS NULL

GROUP BY LGEMPLOYEE.EMP\_NUM

ORDER BY SUM(INV\_TOTAL) DESC ;

The query that I used to best answer the question is that one, and I am going to explain it now.

First, is the selection of the data that I want to display in the table. I selected LGEMPLOYEE.EMP\_NUM, EMP\_TITLE AS ORGANIZATION\_STATUS to select the employee number from the LGEMPLOYEE, and employee title, but I renamed the column as ORGANIZATION\_STATUS, and to be able to rename a column you have to use AS and put the new name. Later, I chose LGDEPARTMENT.DEPT\_NUM, DEPT\_NAME from LGDEPARTMENT table to know the number of the department and the name too. When you select data from different tables, and they have the same Field you need to specify from what table you want to retrieve it, or the query is going to have errors. For example, here with LGDEPARTMENT.DEPT\_NUM, I specify that DEPT\_NUM is displayed from the LGDEPARTMENT table. Also, I selected EMP\_COMM AS COMMISSION from the LGEMPLOYEE table to know the commission that they have, and I renamed the column using the command AS. Then, I wrote SUM(INV\_TOTAL) to sum the INV\_TOTAL and I renamed it as TOTAL\_SALES. Later, I selected MAX(SAL\_FROM) AS LAST\_CONTRACT, SAL\_AMOUNT AS SALARY from LGSALARY\_HISTORY table, and MAX(SAL\_FROM) to know if the employees are still working in the company since it displays when their last contract started, and everyone is still working within the company. Also, SAL\_AMOUNT to know their salary.

After selecting all the data that I wanted to display I had to join the different tables and to did that I used join. The first join was FROM LGEMPLOYEE JOIN LGDEPARTMENT ON LGEMPLOYEE.DEPT\_NUM = LGDEPARTMENT.DEPT\_NUM since the DEPT.NUM has the same type. The next join was JOIN LGINVOICE

ON LGEMPLOYEE.EMP\_NUM = LGINVOICE.EMPLOYEE\_ID which connected the tables LGEMPLOYEE with LGINVOICE through the fields EMP\_NUM and EMPLOYEE\_ID, although they do not have the same name, they have the same type (decimal 6.0). To end with the joins, JOIN LGSALARY\_HISTORY ON LGEMPLOYEE.EMP\_NUM = LGSALARY\_HISTORY.EMP\_NUM and this one connected the table of LGEMPLOYEE and LGSALARY\_HISTORY thanks to the EMP.\_NUM. The next step in my query was, WHERE EMP\_COMM > 0 AND INV\_DATE LIKE '2015%' AND SAL\_END IS NULL and my explanation for that is the following: When you wrote WHERE you are specializing a condition, so WHERE EMP\_COMM > 0 means that I only wanted the employees who had any commission, later INV\_DATE LIKE ‘2015%’ means that I only wanted to display the sales from 2015 onwards, since when you put the symbol ‘%’ in this query means that you want to display the year 2015 and all the months and days in that year, and to end, SAL\_END is NULL is to verify that the employees are still working, since if the end date is null (without data) means that they are still working. Then GROUP BY LGEMPLOYEE.EMP\_NUM, to know the data associated with each employee. Finally, ORDER BY SUM(INV\_TOTAL) DESC to order the table by the total sales in order descent to have the best seller at the top.

This query I explained before, is the best solution to answer the question, since my suggestion to reduce the cost is related to the first question, I would give commission only to the top 5 sellers in the company since the department would increase their sales because everybody would want to earn the commission, and that would create a competitiveness that would help the company in the future to make more profit, with two advantages. The company would make more sales, due to the competitiveness among the employees, and the costs in this department would be reduced because only the top 5 sellers will receive the commissions. I also believe that all the employees would work harder to be in the top 5 and earn the commission. In addition, I want to add that the query is the better to answer this question, since when I wrote the conditions with the WHERE I think that I was precise since I selected only the people with commission superior to 0, I selected only the sales from 2015 and forward and I selected the salary end when is null, which it is the most accurate way to see the employees who are still working in the company.

A picture containing text

Description automatically generated

The table shows us the information that I wanted to select with my query, and the explanation is the following:

The first column is EMP\_NUM and it says the employee number which we use to identify the employee. The next column is the ORGANIZATION\_STATUS which is the status of the employee within the company, and with that information, we can understand better the salaries. The next ones are DEPT\_NUM and DEPT\_NAME whose function is to show us that the employees are working in the same department SALES and the number of it is 200. The next column is very important since it shows the percentage of commission that each employee has, and as you can see everyone has commission thanks WHERE EMP\_COMM > 0. The following column is the total sales in dollars per employee, which is the most important column in this query because we can measure the employees with more sales and make a decision to give the commission only to the top 5. This table shows that the employees that deserve the commission are numbers 83677,84163,84276,84334 and 84078. The next column is the LAST\_CONTRACT which shows us when was the date that each employee signed the contract. Finally, the last column is SALARY, which shows us the salary of each employee. Based on the results, we can see that not the employees with the highest salaries have the highest sales, which indicates and supports my idea of giving a bonus to the top 5 sellers, and for the rest of the employees there are no commissions.

I believe that the company should implement the strategy of commission only for the top 5 sellers, there are so many advantages, and only one disadvantage which is the discontent of people who have commissions right now, and who would lose them for not being among the top 5 sellers, although more people would be happy and motivated because people who did not have commissions would be able to have it if they are in the top 5 sellers in the next year. As I already said, this strategy would reduce the cost of this department and at the same time would increase the profitability of the company, thanks to the competitiveness and the possibility of all the sale department employees being part of the top 5 sellers. Also, the company could give an extra bonus as an all-inclusive trip to a paradise site to increment “prize”.

**3.**

REPORT

After asking all the questions related to the tables and the information that they contain, there are several points that I would like to discuss with the company's management to make an improvement in the tables and in the operation of the company by applying data-driving strategies. In the tables, we have the data of the city and state from where our products are sold, and the location data of the customers as well. With these data, the company could carry out a study to find out from which part of the United States more sales are acquired, and if we have any vendor in that place, the company would have to take a strategy to increase the shipment of products since due to the high demand would get a greater profit, on the other hand, if we do not have any vendor in that location, the company should carry out a study to find out if it is profitable in the long term to create its own distributor, or directly contact a vendor. Also, the company could improve its knowledge about its customer target by gathering data such as age, social status…and with that information select the best products for the customer target. For example, the company could ask me to display a table with the first name and second name of the customer. Then, the zip code, city, state, and total purchases in times order the table by the total purchases in times descent to display the customer information with more purchases at the top. After the display, analyze the data and try to find relations between customers and states in order to know where the company could improve its profit.

The next and most important data that the company should gather is the cost of the product since without that data the company and the people who are working with data are not able to display the most accurate information. I want to explain it with an example: Imagine that the company wants to know what the most profitable product is, without that information for the company is impossible since we cannot choose the product with more sales in dollars or with more sales in time because maybe the product with more sales in dollars is also one of the most expensive in its production, and the same happens with the product with more sale times. As I said in my project, it is an urgent update that the company must do within the next month to be more precise and be able to do successful data-driven strategies.

Thank you.

Nicolas Gonzalez.