

Vention Data Analysis Report

Kallil de Araujo Bezerra

March 12, 2023

Contents

1	Visualization	2
1.1	High Level Sales Analysis	2
1.1.1	Most sold products by Sales Team	5
1.2	Strategic Sales Team	8
1.3	HVS 1	9
2	SQL query	10
2.1	Database creation	10
3	Technical details	14

Chapter 1

Visualization

The data was received in a *.csv* file, and it contains the following columns:

- *accountid* - customer account id
- *amount* - total revenue
- *closedate* - date in which the deal was closed
- *opportunityid* - the id of a possible sales lead
- *opportunity_creation_date__c* - date in which the first contact with a customer was made
- *ownerid* - seller id (the *owner* of the opportunity)
- *primary_application__c* - primary application of the product that is being sold
- *stagename* - in which stage the sale is categorized
- *sales_team__c* - to which sales team the opportunity, seller, and customer belongs

The description of the columns are based solely on the name of the columns itself. In an ideal environment I would get a more accurate definition of each one by asking the person that sent me the file a description of each column, and proceed with the analysis to avoid mistakes and misinterpretations.

1.1 High Level Sales Analysis

Under the column *stagename* it is possible to see 8 different values, they are:

- Closed Lost
- Closed Won
- Prospect
- Project Discovery
- Closing Stage
- Project Quoted
- Design Review
- Awaiting Purchase

To create a chart that shows which sales had a positive result it was considered the registers that have the value *Closed Won*.

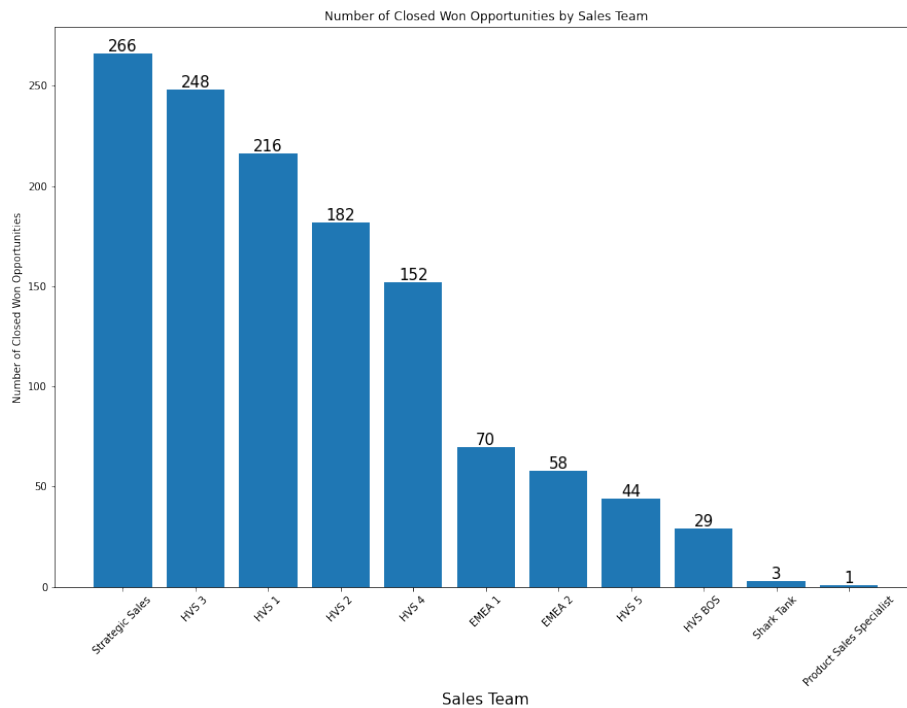


Figure 1: Closed Won versus Sales Team.

As we can see in figure 1 the team with most sales is the *Strategic Sales team*, with 266 closed sales won, followed by *HVS 3* and *HVS 1*, with 248 and 216 sales respectively. From now on the analysis will focus on them.

However, it is also important to see the success rate of each team. Even if a team wins many sales, it doesn't mean that it is the most successful because it may be losing many sales too.

To better evaluate this, it was created a *Sales Index*, which can be calculated using the following formula:

$$Sales\ Index = \frac{\#\ Closed\ Won}{\#\ Closed\ Won + \# \ Closed\ Lost} \quad (1.1)$$

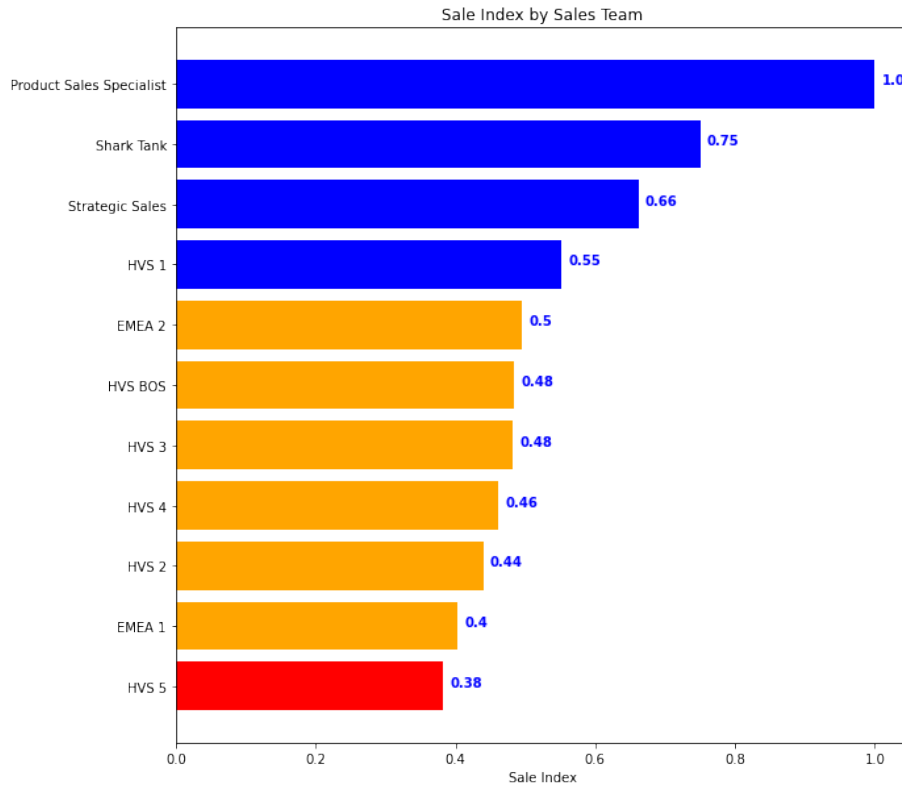


Figure 2: Sales Index.

The color change according to the Sales Index, if it is higher than 0.51 it will be blue, between 0.4 and 0.5 (inclusive) it will be orange, less than 0.4 is red.

From figure 2 it is possible to see that some teams have a high value or even a perfect score. However, those teams have a low number of sales, therefore there is a high chance that when they start to grown their sales number, the *Sales Index* will drop. At the same time, the top team is *Strategic Sales*, with a total of 266 sales and a great 66% Sales Index, or success rate. Knowing which

teams are the most successful may help other teams to identify best practices and areas for improvement.

Finally, it is important to show that the teams that are closing more sales are also doing it quick, as can be seen in figure 3.

sales_team_c	days_to_close
HVS BOS	8.172414
Shark Tank	8.333333
HVS 1	12.662037
Strategic Sales	12.789474
HVS 4	14.657895
HVS 5	14.909091
EMEA 2	15.034483
HVS 3	16.625000
HVS 2	16.835165
EMEA 1	18.314286
Product Sales Specialist	55.000000

Figure 3: Days to close a sale.

The Strategic Sales and HVS 1 are closing sales in less than two weeks, while the HVS 3 team is spending a little over 16 days.

1.1.1 Most sold products by Sales Team

The next analysis will be more focused on a few teams, the objective is to understand what they sell more and how many days they spend between discovering an opportunity and closing a deal.

The pie charts shown in figures 4, 5, and 6 present which products are more relevant to each of the top 3 teams (Strategic Sales, HVS 1, and HVS 3). The *Other Products* are the ones that are not in the top 5, however, it is important to keep them in the visualization so we can see their impact in each team.

From the pie charts it is possible to see that those teams are specialized in different areas. Strategic Sales is strong with Robot Pedestals and Bases, HVS 1 is divided between Safety Enclosures and Workstations, and HVS 3 is experienced with Robot Range Extender. However, all three have at least 10% of sales related to Workstations, therefore, this area is important for all teams.

Top 5 Primary Applications Sold by Strategic Sales

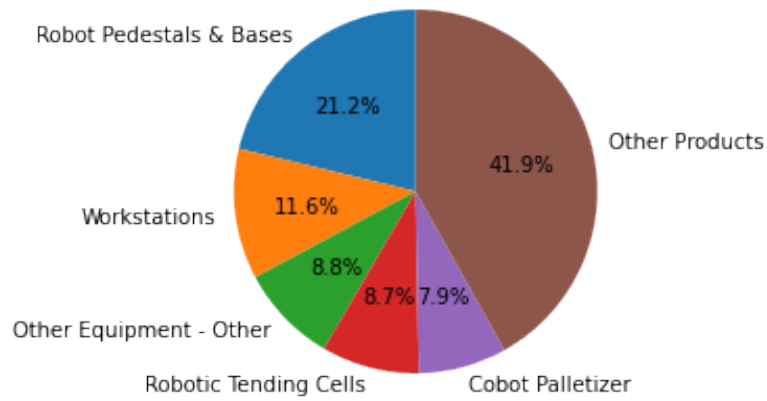


Figure 4: Products sold by Strategic Sales

Top 5 Primary Applications Sold by HVS 1

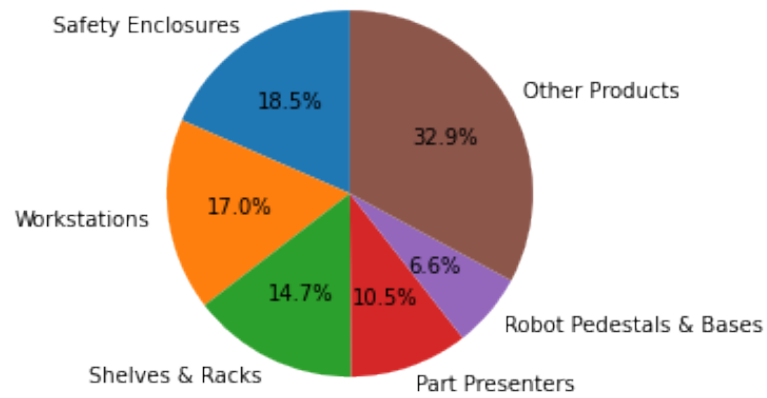


Figure 5: Products sold by HVS 1

Top 5 Primary Applications Sold by HVS 3

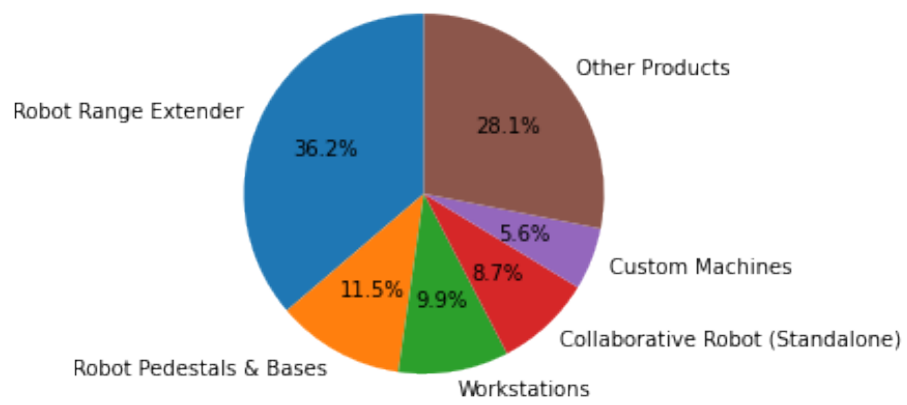


Figure 6: Products sold by HVS 3

1.2 Strategic Sales Team

In this section the Strategic Sales Team will be better analyzed, so we can understand who are the owners of the opportunities and how many sales they did.

stagename ownerid	Closed Won	Closed Lost	sales_index
0056g0000049sqgAAA	80.0	43.0	0.650407
0056g0000049sqhAAA	59.0	27.0	0.686047
0056g0000049sqkAAA	55.0	31.0	0.639535
0056g0000056q5dAAA	43.0	10.0	0.811321
0054v00000E7EDFAA3	23.0	25.0	0.479167
0056g000003L0n7AAG	4.0	0.0	1.000000
0054v00000E65sbAAB	1.0	0.0	1.000000
0054v00000Ef1WvAAJ	1.0	0.0	1.000000

Figure 7: Sellers and their indexes

In figure 7 it is shown the total Closed Won and Closed Lost sales by each of the sellers in the Strategic Sales team, most of them present a good Sales Index. However the seller *0054v00000E7EDFAA3* seems to lose more customers than win new ones. Therefore, it is important to further investigate why this is happening and find a solution.

A possible cause for the low performance of the seller *0054v00000E7EDFAA3* might be a overwhelming number of customers. Therefore lets analyze the how many customers each of the sellers have. The result can be seen in figure 8.

ownerid	
0054v00000E65sbAAB	1
0054v00000E7EDFAA3	21
0054v00000Ef1WvAAJ	7
0056g000003L0n7AAG	3
0056g0000049sqgAAA	83
0056g0000049sqhAAA	58
0056g0000049sqkAAA	68
0056g0000056q5dAAA	37
Name: accountid, dtype: int64	

Figure 8: Amount of clients by sellers

As we can see, the number of costumers that the seller *0054v00000E7EDFAA3* is managing is not high, so it may need further investigation.

1.3 HVS 1

Now we will execute the same analysis with the team HSV 1. Which gives the results from figure 9.

stagename	Closed Won	Closed Lost	sales_index
ownerid			
0054v00000E5wUpAAJ	48.0	28.0	0.631579
0056g00000511ReAAI	42.0	31.0	0.575342
0054v00000EEeP5AAL	42.0	27.0	0.608696
0056g000005720UAAAY	27.0	38.0	0.415385
0056g0000050osIAAA	32.0	15.0	0.680851
0054v00000Edh3HAAR	12.0	16.0	0.428571
0054v00000EeKcBAAV	7.0	12.0	0.368421
0054v00000EfDVZAA3	2.0	5.0	0.285714
0056g00000573Q0AAAY	4.0	2.0	0.666667
0054v00000EIPPCAAP	0.0	1.0	0.000000
0056g0000049sqeAAA	0.0	1.0	0.000000

Figure 9: Sellers and their indexes - HVS 1

As the analysis show, there are many seller with low performance indexes. Despite being one of the top teams in the company, HSV 1 has a high number of low performing sellers. Therefore, further investigation is needed.

ownerid	
0054v00000E5wUpAAJ	66
0054v00000EEeP5AAL	56
0054v00000EIPPCAAP	1
0054v00000Edh3HAAR	37
0054v00000EeKcBAAV	21
0054v00000EfDVZAA3	12
0054v00000Eg1ckAAB	1
0056g0000049sqeAAA	1
0056g0000050osIAAA	30
0056g00000511ReAAI	55
0056g000005720UAAAY	58
0056g00000573Q0AAAY	5
Name: accountid, dtype: int64	

Figure 10: Amount of clients by sellers - HVS 1

The amount of customers that each seller is managing in HVS 1 is higher, therefore it's possible that they are overloaded and may need help.

Chapter 2

SQL query

This chapter will be dedicated to discuss the implementation and query to solve the second task.

2.1 Database creation

To complete this task it was necessary to understand how the tables were related, so a version of what is presented in the task was implemented. The Entity Relation Diagram (ERD) can be seen in figure [1](#).

This diagram was created using the *PGAdmin* tool *ERD for Database*. Moreover, fake data was used to populate the database. Users and products are related to the *Star Wars* universe, and as this database was created only for test purposes, there isn't a high amount of data on it. Some users have a email that ends in *@vention.cc* to add a condition that should be met, according to the Technical Test.

At first the implementation tried to follow a star schema, but the way that the data is related is closer to a snowflake schema. Therefore, a new data model would be needed to use a star schema and the end result would be very different from what was shown.

The query can be seen in table [1](#) in addition to that, images [1](#) and [2](#)

SQL Query

```
SELECT date_trunc('month', o.created_at) AS order_month,
COUNT(DISTINCT o.id) AS count_orders,
SUM(CASE
  WHEN op.li.partner_id IS NULL THEN oli.item_price * oli.quantity
  ELSE op.li.item_price * pf.contract_rate * op.li.quantity
END) AS total_gross_sales,
SUM(CASE
  WHEN op.li.partner_id IS NULL THEN oli.item_cost * oli.quantity
  ELSE op.li.item_price * op.li.quantity
END) AS total_net_sales,
array_agg(o.id) AS list_order
FROM orders o
JOIN order_line_items oli ON o.id = oli.order_id
LEFT JOIN order_partner_line_items op.li
  ON o.id = op.li.order_id AND oli.part_number = op.li.part_number
LEFT JOIN partner_feature pf
  ON op.li.partner_id = pf.id
JOIN users u ON o.user_id = u.id
WHERE u.email NOT LIKE '%@vention.cc'
AND o.created_at >= '2021-01-01'
GROUP BY order_month
ORDER BY order_month;
```

Table 1: Final Query - Exercise 02

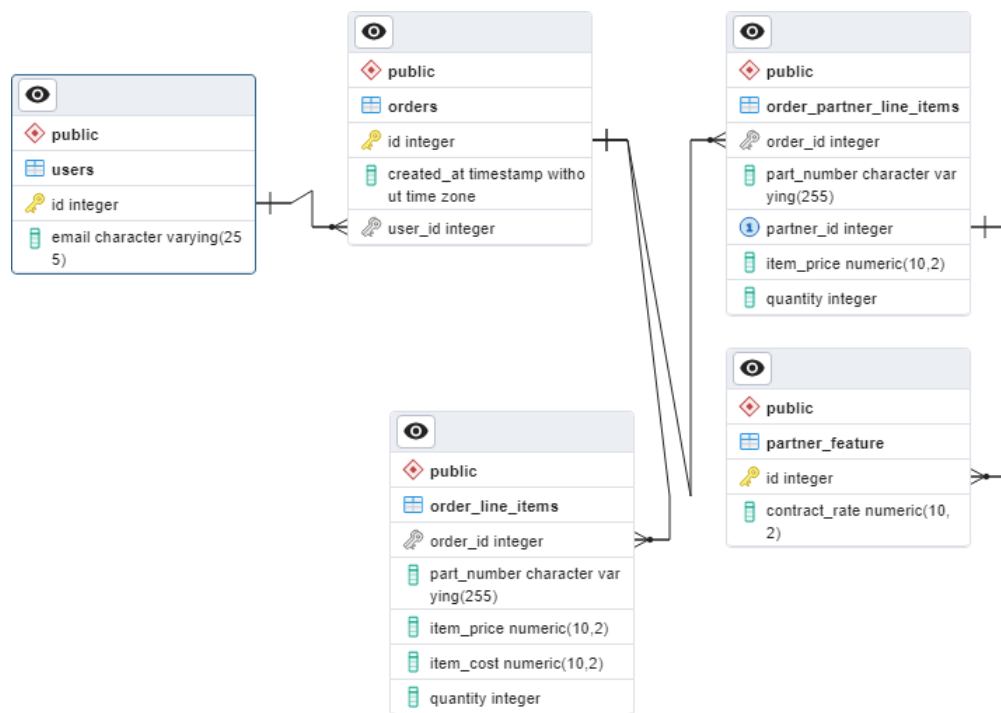


Figure 1: ERD of the implemented database

order_month timestamp without time zone	count_orders bigint	total_gross_sales numeric	total_net_sales numeric	list_order integer[]
2021-01-01 00:00:00	1	1814576.00	2394.00	{961}
2021-02-01 00:00:00	1	1237185.00	988.00	{1058}
2021-04-01 00:00:00	1	1679496.00	5928.00	{983}
2021-06-01 00:00:00	1	188730.00	474.00	{925}
2021-07-01 00:00:00	1	1876959.00	1386.00	{1019}
2021-08-01 00:00:00	1	842860.00	2440.00	{999}
2021-10-01 00:00:00	1	127112.00	1328.00	{1100}
2021-11-01 00:00:00	1	725489.00	5681.00	{1089}
2021-12-01 00:00:00	1	1257825.00	4200.00	{955,955}
2022-01-01 00:00:00	4	1680162.00	5164.00	{1065,1097,922,1045}
2022-02-01 00:00:00	2	983835.00	5100.00	{1080,1008}
2022-04-01 00:00:00	1	635310.00	2889.00	{1096}
2022-05-01 00:00:00	1	484464.00	996.00	{1026}
2022-06-01 00:00:00	1	3783.00	450.00	{1066}
2022-07-01 00:00:00	1	406615.00	405.00	{921}
2022-08-01 00:00:00	1	368100.00	2078.00	{976,976}
2022-11-01 00:00:00	1	702.00	119.00	{937}
2023-01-01 00:00:00	4	3473755.00	11116.00	{1075,958,990,1037}
2023-02-01 00:00:00	1	198270.00	1510.00	{985}
2023-06-01 00:00:00	1	1512544.00	1984.00	{987}
2023-07-01 00:00:00	1	429702.00	6006.00	{1087}
2023-08-01 00:00:00	2	2478993.00	9360.00	{957,957,986}
2023-11-01 00:00:00	1	893991.00	1995.00	{964}
2023-12-01 00:00:00	2	351433.00	3535.00	{982,1021}

Figure 2: Query result

Chapter 3

Technical details