Project Requirements Document

Fernando Fontoura de Araujo

fontouraaraujo@gmail.com

+1 (778) 997 8877

Vancouver, Brith Columbia - Canada

08/13/2023

Summary

Introduction

This project documentation was developed as a free project, as a data analyst from the fictional company Sell Boardgames would do in his daily work.

In macro needs, the project was divided into 7 major phases.

Phase 01: Construction of the Stage area

Phase 02: Construction of the Datawarehouse

Phase 03: Analysis Services

Phase 04: Reporting Services

Phase 05: Backup Structure

Phase 06: Power BI

Phase 07: Data Analysis and Executive Presentation

For the development of the business intelligence project, Python and tools from the Microsoft data environment were used, namely: SQL Server, Management Studio, Integration Services, Analysis Services, Reporting Services, and Power BI.

All files, packages, and scripts developed and/or used in this project will be sent along with this report.

Project Keywords

SQL Server - Power BI - DAX - Management Studio - SQL - Data Warehouse - ABC Analysis - Common Table Expression (CTE) – Views - Data Analysis - Power Point Presentation - Project Requirements Document - Executive Presentation

Specification

Purpose

After carrying out the development of the main project, I am using these analyzes to demonstrate my analytical side. The focus of this project is to show a little of my ability to compile and analyze data, obtain insights, and prepare a high-level presentation about these insights, supporting each conclusion obtained with data.

Background

SELL is in a leading position as the largest online retailer of boardgames in the world and has used our knowledge of the space to expand into creating our own brand of boardgames. We have gotten to this position following our instinct, and intuitive knowledge of the market, putting customers first, and a simple yet effective strategy of focusing on content for users, SEO, and email marketing.

As a company, we want to be a high-performing, purpose-driven team that is informed by data and to use metrics to get better results helping us achieve our objectives. We know that we can do better for users by validating our insights with data and improving our understanding and performance through analytics and new ways of working.

Assignment

After completing your data organization and KPI construction, create a presentation to communicate to the executive level. The goal is to see what opportunities I uncover in the data and any suggested strategies I can come up with. At Sell Boardgames, we really believe in being curious. I might end up having more questions after reviewing the dataset that I would like to explore if I had more data. One of the goals is to include any of these questions in my presentation and what would be my strategy for digging deeper and finding the answers.

Data Analysis

To perform the data analysis I used views and selects in the database followed by the use of Excel and a Power BI file for analysis. This Power BI file is a raw file with all the information to carry out the analytical approaches, if these approaches become routine, this data goes to dashboards developed for stakeholders, such as those developed in the previous section.
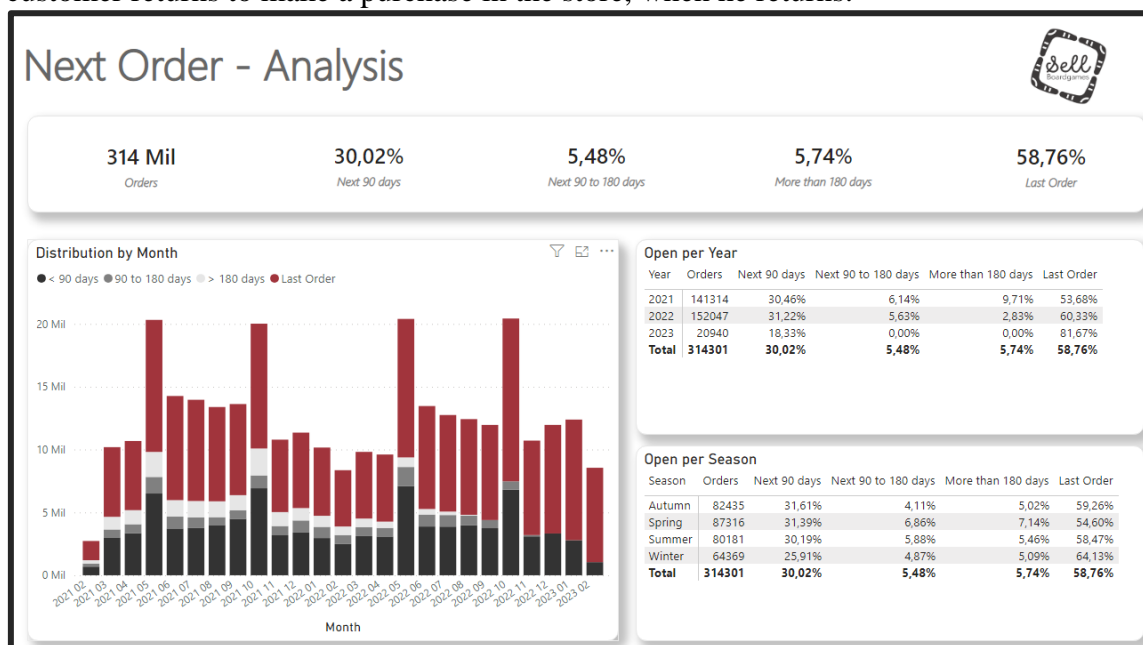
These are the files used for analysis.

DATA_ANALYSIS.sql: Todos os scripts usados para essa construção.

DATA_ANALYSIS_SELL.xls: excel utilizado em análises de trend e organização de quadrantes.

SELL_POWERB_ANALYSIS.pbix: PowerBI com analyses de dados experimentais.

Screen - Next Order

The analysis of the next order was developed to understand the time that a customer returns to make a purchase in the store, when he returns.

Next Order - Analysis

314 Mil	30,02%	5,48%	5,74%	58,76%
Orders	Next 90 days	Next 90 to 180 days	More than 180 days	Last Order

Open per Year

Year	Orders	Next 90 days	Next 90 to 180 days	More than 180 days	Last Order
2021	141314	30,46%	6,14%	9,71%	53,68%
2022	152047	31,22%	5,63%	2,83%	60,33%
2023	20940	18,33%	0,00%	0,00%	81,67%
Total	**314301**	**30,02%**	**5,48%**	**5,74%**	**58,76%**

Open per Season

Season	Orders	Next 90 days	Next 90 to 180 days	More than 180 days	Last Order
Autumn	82435	31,61%	4,11%	5,02%	59,26%
Spring	87316	31,39%	6,86%	7,14%	54,60%
Summer	80181	30,19%	5,88%	5,46%	58,47%
Winter	64369	25,91%	4,87%	5,09%	64,13%
Total	**314301**	**30,02%**	**5,48%**	**5,74%**	**58,76%**

Screen – Dollar per Session

The analysis of dollars per session relates to the two files made available and seeks to understand the proportion of sales generated by the number of sessions.
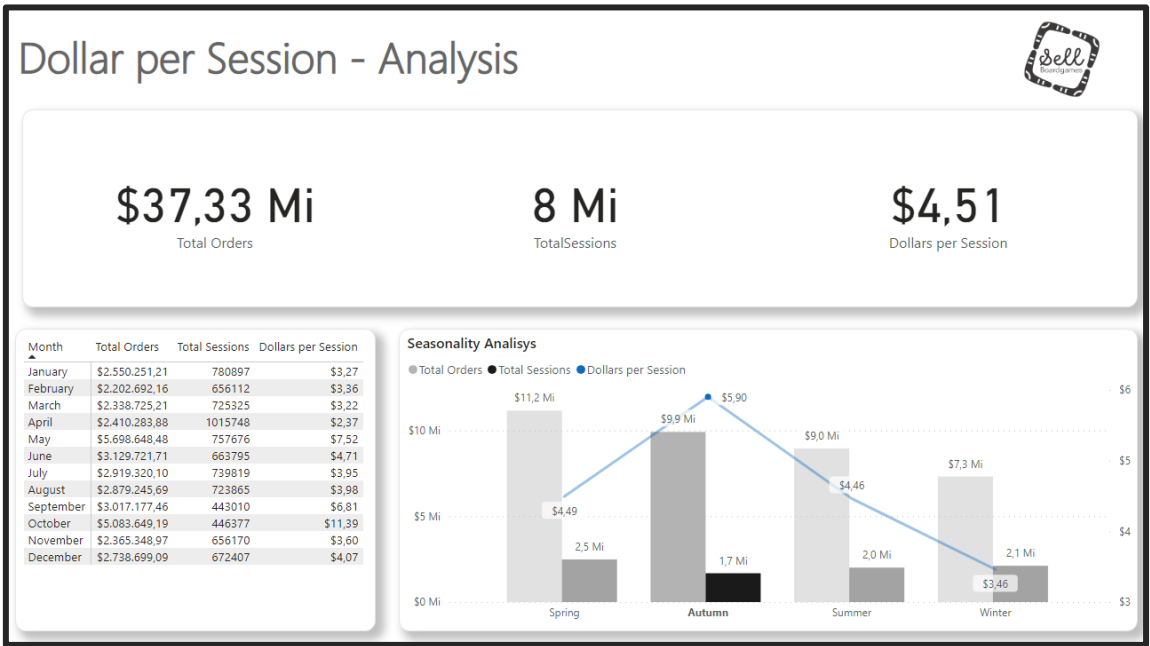
Screen – ABC Analysis by Products

ABC analysis is focused on discovering which products are impactful for the business.
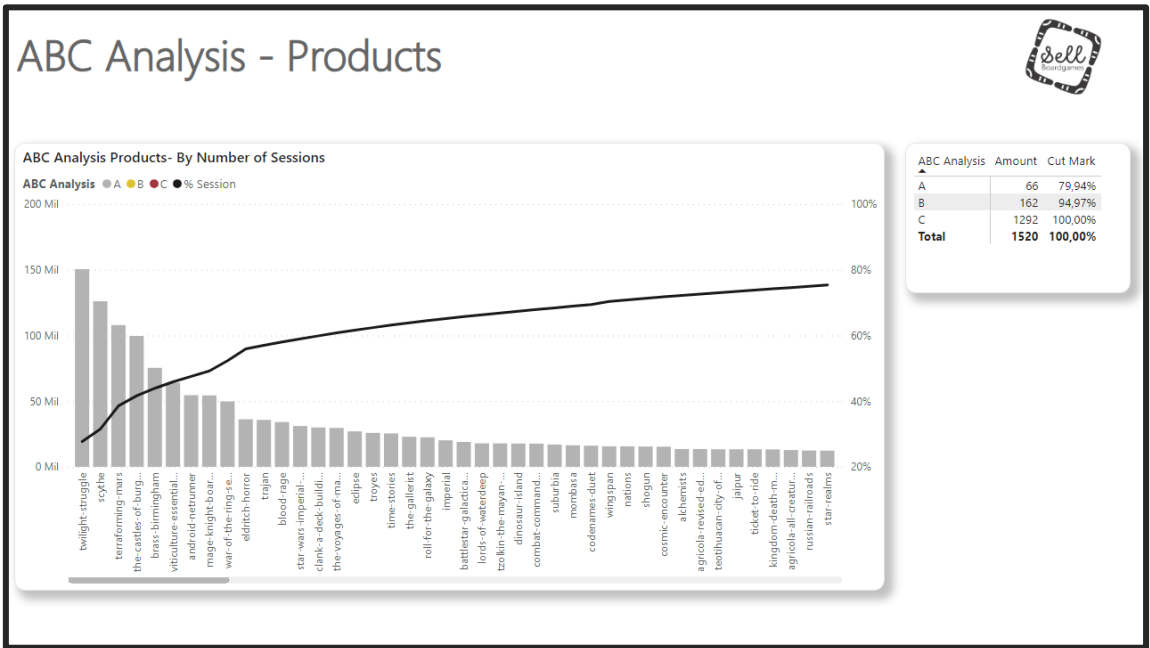
Screen – Basic Numbers and Statistics

Focused on a simple opening of the data to facilitate comparison and develop the first analyses.
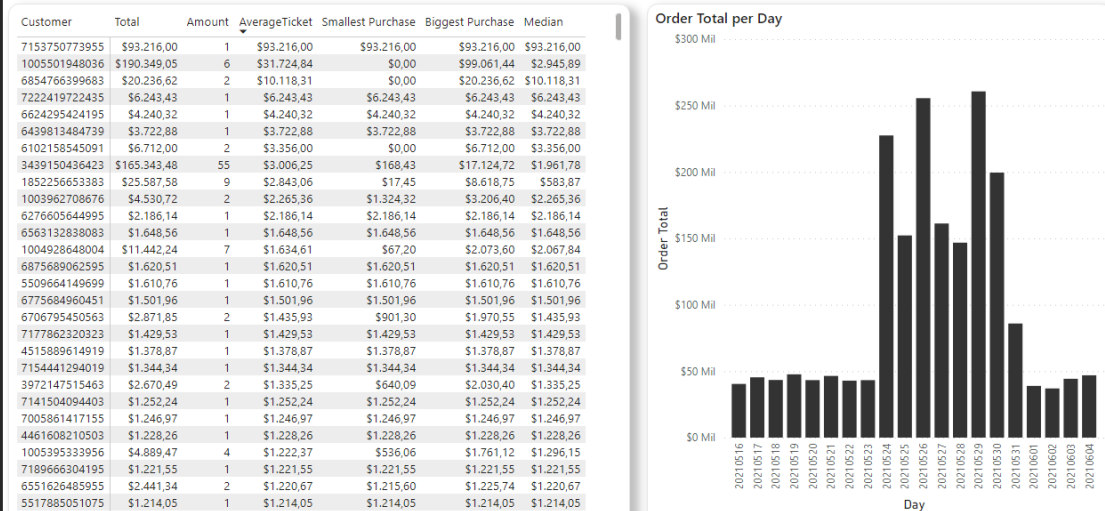
Basic Numbers and Statistics

Customer	Total	Amount	AverageTicket	Smallest Purchase	Biggest Purchase	Median
7153750773955	$93.216,00	1	$93.216,00	$93.216,00	$93.216,00	$93.216,00
1005501948036	$190.349,05	6	$31.724,84	$0,00	$99.061,44	$2.945,89
6854766399683	$20.236,62	2	$10.118,31	$0,00	$20.236,62	$10.118,31
7222419722435	$6.243,43	1	$6.243,43	$6.243,43	$6.243,43	$6.243,43
6624295424195	$4.240,32	1	$4.240,32	$4.240,32	$4.240,32	$4.240,32
6439813484739	$3.722,88	1	$3.722,88	$3.722,88	$3.722,88	$3.722,88
6102158545091	$6.712,00	2	$3.356,00	$0,00	$6.712,00	$3.356,00
3439150436423	$165.343,48	55	$3.006,25	$168,43	$17.124,72	$1.961,78
1852256653383	$25.587,58	9	$2.843,06	$17,45	$8.618,75	$583,87
1003962708676	$4.530,72	2	$2.265,36	$1.324,32	$3.206,40	$2.265,36
6276605644995	$2.186,14	1	$2.186,14	$2.186,14	$2.186,14	$2.186,14
6563132838083	$1.648,56	1	$1.648,56	$1.648,56	$1.648,56	$1.648,56
1004928648004	$11.442,24	7	$1.634,61	$67,20	$2.073,60	$2.067,84
6875689062595	$1.620,51	1	$1.620,51	$1.620,51	$1.620,51	$1.620,51
5509664149699	$1.610,76	1	$1.610,76	$1.610,76	$1.610,76	$1.610,76
6775684960451	$1.501,96	1	$1.501,96	$1.501,96	$1.501,96	$1.501,96
6706795450563	$2.871,85	2	$1.435,93	$901,30	$1.970,55	$1.435,93
7177862320323	$1.429,53	1	$1.429,53	$1.429,53	$1.429,53	$1.429,53
4515889614919	$1.378,87	1	$1.378,87	$1.378,87	$1.378,87	$1.378,87
7154441294019	$1.344,34	1	$1.344,34	$1.344,34	$1.344,34	$1.344,34
3972147515463	$2.670,49	2	$1.335,25	$640,09	$2.030,40	$1.335,25
7141504094403	$1.252,24	1	$1.252,24	$1.252,24	$1.252,24	$1.252,24
7005861417155	$1.246,97	1	$1.246,97	$1.246,97	$1.246,97	$1.246,97
4461608210503	$1.228,26	1	$1.228,26	$1.228,26	$1.228,26	$1.228,26
1005395333956	$4.889,47	4	$1.222,37	$536,06	$1.761,12	$1.296,15
7189666304195	$1.221,55	1	$1.221,55	$1.221,55	$1.221,55	$1.221,55
6551626485955	$2.441,34	2	$1.220,67	$1.215,60	$1.225,74	$1.220,67
5517885051075	$1.214,05	1	$1.214,05	$1.214,05	$1.214,05	$1.214,05

Screen – Quadrant Customers

Important analysis to segment customers according to their purchase pattern and, above all, to find outliers.

Quadrant Customers - Outliers Analysis
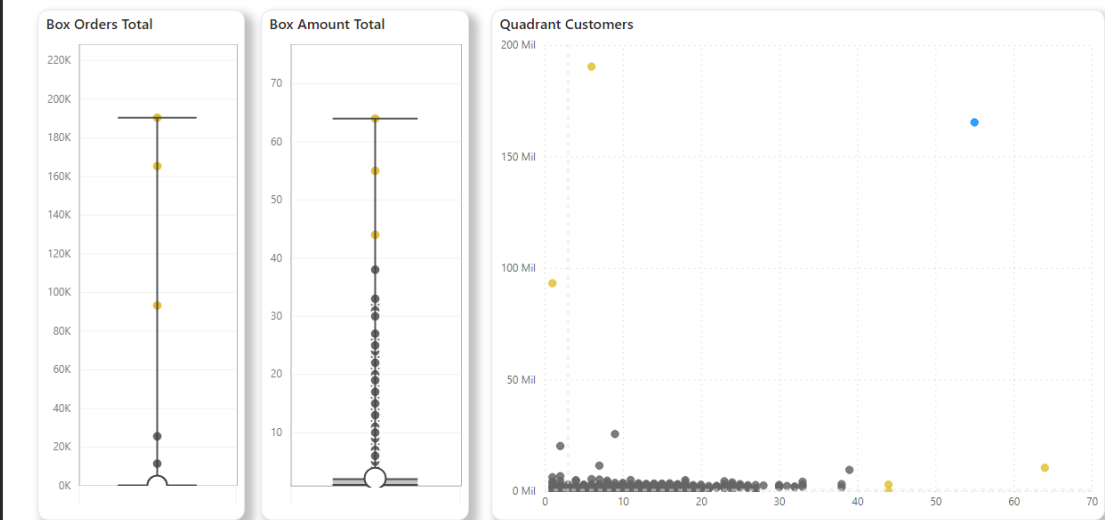
Views for Data Analysis

To carry out the data analyses, I created 10 views and 1 select treating the data still in the DW.

- V_TREND_GA_PRODUCTS – Organizing the already filtered data with filfillness <> of unfulfilled. I created two items focused on its performance.
- V_TREND_DIM_ORDERS - Filtering only customers who made at least 3 purchases in the last 360 days.
- V_QUAD_YEAR – Filtering only sales from customers who made their first Purchase less than 360 days ago.
- V_QUAD - They are used for the 12-month Moving Average and Month over Month comparison.
- V_QUAD_TOTAL_DIST - Performs the aggregation of the Google Analytics file.
- V_QUAD_AMOUNT_DIST - Prepares data for ABC Analysis.
- V_QUAD_FINAL_DIST - Prepares data for ABC Analysis.
- SEASONALITY_ANALYSIS- Prepares data for ABC Analysis.
- V_DOLLARPERSESSION_ANALYSIS - Prepares data for ABC Analysis.
- DASH_BASE_TIME_NEW_ORDERS - Prepares data for ABC Analysis.

```
----------------------------------------
--- VIEW V_TREND_GA_PRODUCTS
----------------------------------------
CREATE VIEW V_TREND_GA_PRODUCTS as
with CTE_GA_PRODUCTS as
(select
        TIME_YEAR_MONTH,
        GA_VALUE,
        sum(GA_SESSIONS) GA_SESSIONS
from
        DW_FACT_GA ga
        inner join DIM_TIME t on ga.GA_DATE = t.TIME_DATE
where
        GA_CATEGORY = 'products'
group by
        TIME_YEAR_MONTH,
        GA_VALUE
),
CTE_GA_PRODUCTS_2 as
(select
        GA_VALUE,
        sum(GA_SESSIONS) GA_SESSIONS,
        row_number() over(order by sum(GA_SESSIONS) desc) LINHA
from
        CTE_GA_PRODUCTS
group by
        GA_VALUE)

select
        GA_VALUE,
```

```
        isnull([202001],0) M_202001,
        isnull([202002],0) M_202002,
        isnull([202003],0) M_202003,
        isnull([202004],0) M_202004,
        isnull([202005],0) M_202005,
        isnull([202006],0) M_202006,
        isnull([202007],0) M_202007,
        isnull([202008],0) M_202008,
        isnull([202009],0) M_202009,
        isnull([202010],0) M_202010,
        isnull([202011],0) M_202011,
        isnull([202012],0) M_202012,
        isnull([202101],0) M_202101,
        isnull([202102],0) M_202102,
        isnull([202103],0) M_202103,
        isnull([202104],0) M_202104,
        isnull([202105],0) M_202105,
        isnull([202106],0) M_202106,
        isnull([202107],0) M_202107,
        isnull([202108],0) M_202108,
        isnull([202109],0) M_202109,
        isnull([202110],0) M_202110,
        isnull([202111],0) M_202111,
        isnull([202112],0) M_202112,
        isnull([202201],0) M_202201,
        isnull([202202],0) M_202202,
        isnull([202203],0) M_202203,
        isnull([202204],0) M_202204,
        isnull([202205],0) M_202205,
        isnull([202206],0) M_202206,
        isnull([202207],0) M_202207,
        isnull([202208],0) M_202208
from
        CTE_GA_PRODUCTS
        PIVOT
        (
        sum(GA_SESSIONS)
        FOR TIME_YEAR_MONTH IN
            (
    [202001],
            [202002],
            [202003],
            [202004],
            [202005],
            [202006],
```

```
            [202007],
            [202008],
            [202009],
            [202010],
            [202011],
            [202012],
            [202101],
            [202102],
            [202103],
            [202104],
            [202105],
            [202106],
            [202107],
            [202108],
            [202109],
            [202110],
            [202111],
            [202112],
            [202201],
            [202202],
            [202203],
            [202204],
            [202205],
            [202206],
            [202207],
            [202208]
            )
        ) AS pivot_table
where GA_VALUE in (select GA_VALUE from CTE_GA_PRODUCTS_2 where
linha <=20)


----------------------------------------
--- VIEW V_TREND_DIM_ORDERS
----------------------------------------
CREATE VIEW V_TREND_DIM_ORDERS as
with CTE_DIM_ORDERS as
(select
        TIME_YEAR_MONTH,
        GA_VALUE,
        sum(GA_SESSIONS) GA_SESSIONS
from
        DW_FACT_GA ga
        inner join DIM_TIME t on ga.GA_DATE = t.TIME_DATE
where
        GA_CATEGORY = 'products'
```

```
group by
        TIME_YEAR_MONTH,
        GA_VALUE
),
CTE_DIM_ORDERS_2 as
(select
        GA_VALUE,
        sum(GA_SESSIONS) GA_SESSIONS,
        row_number() over(order by sum(GA_SESSIONS) desc) LINHA
from
        CTE_GA_PRODUCTS
group by
        GA_VALUE)

select
        GA_VALUE,
        isnull([202001],0) M_202001,
        isnull([202002],0) M_202002,
        isnull([202003],0) M_202003,
        isnull([202004],0) M_202004,
        isnull([202005],0) M_202005,
        isnull([202006],0) M_202006,
        isnull([202007],0) M_202007,
        isnull([202008],0) M_202008,
        isnull([202009],0) M_202009,
        isnull([202010],0) M_202010,
        isnull([202011],0) M_202011,
        isnull([202012],0) M_202012,
        isnull([202101],0) M_202101,
        isnull([202102],0) M_202102,
        isnull([202103],0) M_202103,
        isnull([202104],0) M_202104,
        isnull([202105],0) M_202105,
        isnull([202106],0) M_202106,
        isnull([202107],0) M_202107,
        isnull([202108],0) M_202108,
        isnull([202109],0) M_202109,
        isnull([202110],0) M_202110,
        isnull([202111],0) M_202111,
        isnull([202112],0) M_202112,
        isnull([202201],0) M_202201,
        isnull([202202],0) M_202202,
        isnull([202203],0) M_202203,
        isnull([202204],0) M_202204,
        isnull([202205],0) M_202205,
```

```
        isnull([202206],0) M_202206,
        isnull([202207],0) M_202207,
        isnull([202208],0) M_202208
from
        CTE_GA_PRODUCTS
        PIVOT
        (
        sum(GA_SESSIONS)
        FOR TIME_YEAR_MONTH IN
                (
        [202001],
                [202002],
                [202003],
                [202004],
                [202005],
                [202006],
                [202007],
                [202008],
                [202009],
                [202010],
                [202011],
                [202012],
                [202101],
                [202102],
                [202103],
                [202104],
                [202105],
                [202106],
                [202107],
                [202108],
                [202109],
                [202110],
                [202111],
                [202112],
                [202201],
                [202202],
                [202203],
                [202204],
                [202205],
                [202206],
                [202207],
                [202208]
                )
        ) AS pivot_table
```

```
where GA_VALUE in (select GA_VALUE from CTE_GA_PRODUCTS_2 where
linha <=20)


----------------------------------------
--- VIEW V_QUAD_YEAR
----------------------------------------
CREATE VIEW V_QUAD_YEAR as
with CTE_DIM_ORDERS as
(
select
        TIME_YEAR,
        CUSTOMER_ID,
        sum(ORDER_TOTAL) ORDER_TOTAL,
        count(CUSTOMER_ID) AMOUNT_TOTAL,
        (case
                when sum(ORDER_TOTAL) >= 500 then 'TOTAL_A'
                else 'TOTAL_B'
        end) QUAD_TOTAL,
        (case
                when count(CUSTOMER_ID) >= 3 then 'AMOUNT_A'
                else 'AMOUNT_B'
        end) QUAD_AMOUNT,
        (case
                when sum(ORDER_TOTAL) >= 500 and count(CUSTOMER_ID) >= 3
then 1
                when sum(ORDER_TOTAL) >= 500 and count(CUSTOMER_ID) < 3
then 2
                when sum(ORDER_TOTAL) < 500 and count(CUSTOMER_ID) >= 3
then 3
                else 4
        end) QUAD_YEAR
from
        DIM_ORDERS o
        inner join DIM_TIME t on o.CUSTOMER_CREATED_TIME_IDSK =
t.TIME_IDSK
        inner join DIM_CUSTOMERS c on o.ORDER_CUSTOMER_ID =
c.CUSTOMER_IDSK
group by
        TIME_YEAR,
        CUSTOMER_ID
)

select
        CUSTOMER_ID,
```

```
        isnull([2014],0) Y_2014,
        isnull([2015],0) Y_2015,
        isnull([2016],0) Y_2016,
        isnull([2017],0) Y_2017,
        isnull([2018],0) Y_2018,
        isnull([2019],0) Y_2019,
        isnull([2020],0) Y_2020,
        isnull([2021],0) Y_2021,
        isnull([2022],0) Y_2022
from
        CTE_DIM_ORDERS
        PIVOT
        (
        sum(QUAD_YEAR)
        FOR TIME_YEAR IN
                (
                [2014],
                [2015],
                [2016],
                [2017],
                [2018],
                [2019],
                [2020],
                [2021],
                [2022]
                )
        ) AS pivot_table


----------------------------------------
--- VIEWS V_QUAD
----------------------------------------
CREATE VIEW V_QUAD_FINAL as
select
        CUSTOMER_ID,
        sum(ORDER_TOTAL) ORDER_TOTAL,
        count(CUSTOMER_ID) AMOUNT_TOTAL,
        (case
                when sum(ORDER_TOTAL) >= 500 then 'TOTAL_A'
                else 'TOTAL_B'
        end) QUAD_TOTAL,
        (case
                when count(CUSTOMER_ID) >= 3 then 'AMOUNT_A'
                else 'AMOUNT_B'
        end) QUAD_AMOUNT,
```

```
        (case
                when sum(ORDER_TOTAL) >= 500 and count(CUSTOMER_ID) >= 3
then 1
                when sum(ORDER_TOTAL) >= 500 and count(CUSTOMER_ID) < 3
then 2
                when sum(ORDER_TOTAL) < 500 and count(CUSTOMER_ID) >= 3
then 3
                else 4
        end) QUAD_FINAL
from
        DIM_ORDERS o
        inner join DIM_TIME t on o.CUSTOMER_CREATED_TIME_IDSK =
t.TIME_IDSK
        inner join DIM_CUSTOMERS c on o.ORDER_CUSTOMER_ID =
c.CUSTOMER_IDSK
group by
        TIME_YEAR,
        CUSTOMER_ID

CREATE VIEW V_QUAD_TOTAL_DIST as
select
        QUAD_TOTAL,
        count(QUAD_TOTAL) AMOUNT_QUAD_TOTAL,
        count(QUAD_TOTAL)*1.0/(select count(*) from V_QUAD_FINAL)
PCT_QUAD_TOTAL
from
        V_QUAD_FINAL
group by
        QUAD_TOTAL

CREATE VIEW V_QUAD_AMOUNT_DIST as
select
        QUAD_AMOUNT,
        count(QUAD_AMOUNT) AMOUNT_QUAD_AMOUNT,
        count(QUAD_AMOUNT)*1.0/(select count(*) from V_QUAD_FINAL)
PCT_QUAD_AMOUNT
from
        V_QUAD_FINAL
group by
        QUAD_AMOUNT

CREATE VIEW V_QUAD_FINAL_DIST as
select
        QUAD_FINAL,
        count(QUAD_FINAL) AMOUNT_QUAD,
```

```
        count(QUAD_FINAL)*1.0/(select count(*) from V_QUAD_FINAL)
PCT_QUAD
from
        V_QUAD_FINAL
group by
        QUAD_FINAL

CREATE VIEW SEASONALITY_ANALYSIS AS
select
        t1.TIME_SEASON_YEAR,
        t1.GA_SESSIONS,
        t2.ORDER_TOTAL
from
        (
        select
                TIME_SEASON_YEAR,
                sum(GA_SESSIONS) GA_SESSIONS
        from
                DW_FACT_GA ga
                inner join DIM_TIME t on ga.GA_DATE = t.TIME_DATE
        group by
                TIME_SEASON_YEAR
        ) t1
        inner join
        (
        select
                TIME_SEASON_YEAR,
                sum(ORDER_TOTAL) ORDER_TOTAL
        from
                DIM_ORDERS o
                inner join DIM_TIME t on o.ORDER_CREATED_TIME_IDSK =
t.TIME_IDSK
        group by
                TIME_SEASON_YEAR
        ) t2 on t1.TIME_SEASON_YEAR = t2.TIME_SEASON_YEAR


CREATE VIEW V_DOLLARPERSESSION_ANALYSIS AS
select
        t1.TIME_MONTH_NAME,
        t1.TIME_MONTH,
        t1.GA_SESSIONS,
        t2.ORDER_TOTAL
from
        (
```

```sql
select
        TIME_MONTH_NAME,
        TIME_MONTH,
        sum(GA_SESSIONS) GA_SESSIONS
from
        DW_FACT_GA ga
        inner join DIM_TIME t on ga.GA_DATE = t.TIME_DATE
group by
        TIME_MONTH_NAME,
        TIME_MONTH
) t1
inner join
(
select
        TIME_MONTH_NAME,
        TIME_MONTH,
        sum(ORDER_TOTAL) ORDER_TOTAL
from
        DIM_ORDERS o
        inner join DIM_TIME t on o.ORDER_CREATED_TIME_IDSK =
t.TIME_IDSK
group by
        TIME_MONTH_NAME,
        TIME_MONTH
) t2 on t1.TIME_MONTH_NAME = t2.TIME_MONTH_NAME

select
        *
from
        (
        select
                GA_VALUE PRODUCT,
                sum(GA_SESSIONS) GA_SESSIONS,
                row_number() over(order by sum(GA_SESSIONS) desc) POSITION
        from
                DW_FACT_GA ga
                inner join DIM_TIME t on ga.GA_DATE = t.TIME_DATE
        where
                GA_CATEGORY = 'products'
        group by
                GA_VALUE
        ) tb
where
        PRODUCT like '%bundle%'
order by
```

```
        POSITION asc

CREATE VIEW DASH_BASE_TIME_NEW_ORDERS as
select
        date_amount.TIME_DATE,
        NUMBER_ORDERS,
        NUMBER_ORDERS-isnull(NUMBER_ORDERS_CHURN_90,0)
NUMBER_ORDERS_90_MINUS,
        isnull(NUMBER_ORDERS_CHURN_90,0)-
isnull(NUMBER_ORDERS_CHURN_180,0) NUMBER_ORDERS_90_180,
        isnull(NUMBER_ORDERS_CHURN_180,0)-
isnull(NUMBER_ORDERS_CHURN_LAST,0) NUMBER_ORDERS_180_PLUS,
        isnull(NUMBER_ORDERS_CHURN_LAST,0)
NUMBER_ORDERS_CHURN_LAST
from
        (
        select
                TIME_DATE,
                count(CUSTOMER_ID) NUMBER_ORDERS
        from
                DASH_BASE_CUSTOMERS_ORDERS
        group by
                TIME_DATE
        ) date_amount
        left join
        (
        select
                TIME_DATE,
                count(CUSTOMER_ID) NUMBER_ORDERS_CHURN_90
        from
                DASH_BASE_CUSTOMERS_ORDERS
        where
                datediff(day,TIME_DATE,NEXT_SELL)>=90 or NEXT_CUSTOMER
<> CUSTOMER_ID
        group by
                TIME_DATE
        ) date_amount_churn90 on date_amount.TIME_DATE =
date_amount_churn90.TIME_DATE
        left join
        (
        select
                TIME_DATE,
                count(CUSTOMER_ID) NUMBER_ORDERS_CHURN_180
        from
                DASH_BASE_CUSTOMERS_ORDERS
```

```
        where
                datediff(day,TIME_DATE,NEXT_SELL)>=180 or NEXT_CUSTOMER
<> CUSTOMER_ID
        group by
                TIME_DATE
        ) date_amount_churn180 on date_amount.TIME_DATE =
date_amount_churn180.TIME_DATE
        left join
        (
        select
                TIME_DATE,
                count(CUSTOMER_ID) NUMBER_ORDERS_CHURN_LAST
        from
                DASH_BASE_CUSTOMERS_ORDERS
        where
                NEXT_CUSTOMER <> CUSTOMER_ID
        group by
                TIME_DATE
        ) date_amount_churn_last on date_amount.TIME_DATE =
date_amount_churn_last.TIME_DATE
```

Power Point Presentation

In the Files folder made available to the SELL team is the file ExecutivePresentation_SellBoardgames_FernandoAraujo.ppt with the presentation developed to be presented in the last stage of the process. In this presentation are the insights taken from the analyses, their justifications, and questions that are asked to deepen the understanding of the company's data.

Fernando Araujo | Executive Presentation
Sell Boardgames

Thanks

Thank you for the opportunity, it was an honor to be able to analyze all the data from this fictitious company and to be able to show a little of my work. Thank you for accompanying the whole project and I am at your disposal if you want a presentation or ask any questions.

Warmly Regards,

Fernando Araujo