Maven Amazonas

A Data Analysis & Business Intelligence Project

by Supreetha Hanasoge

Key Objectives

Understand	As a data analyst, understand the database to better help with the business.
Ensure	Ensure database administration best practices so that the company is set to continue to scale.
Perform	Perform EDA to find key patterns in the data.
Analyze	Analyze the results to answer some of the key business questions to the target audience i.e. the company **leadership team.**

Data & Approach

- Maven Amazonas is a (fictional) Brazilian eCommerce firm that sells a wide-range of consumables.
- The data consists of 8 different tables to capturing data on customers, sellers, products, orders and reviews, ranging from September 2016 to September 2018.
- The raw data was present in the form of a .sql file which was uploaded into the MySQL
- The Tables were examined individually in MySQL to ensure:
  - Clean data
  - Consistency & Coherency in the values
- Exploratory data analysis was performed using MySQL
- Assembled the findings with visuals in Microsoft Power BI dashboards that address various business use-cases.

Key Business Questions

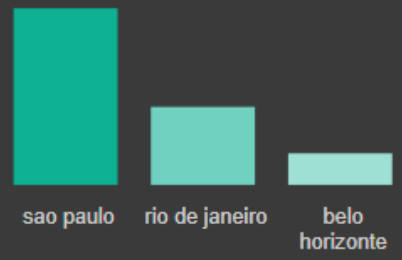
- What is the trend of monthly revenue from items sold ?
- What are the top product categories that are driving the revenue from items sold for 2018? How does it compare with the revenue they brought in from 2017, calculate the YoY increase.
- How is the monthly trend of order volume, orders delivered to customer late and the share of orders that were late? Is on-time delivery improving ?
- What is the monthly trend of order review volume and the share of 5 star reviews ?

Summary

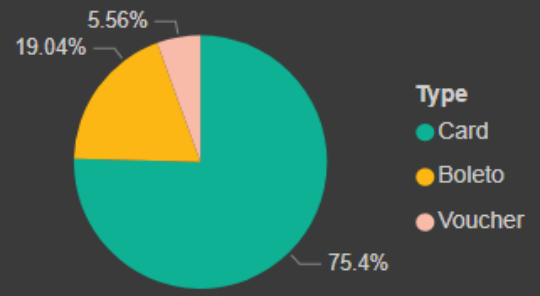
Customers	Total Orders	Total Revenue	Unique Products	Unique Sellers
96K	99K	$16M	32.95K	3095

Maven Amazonas

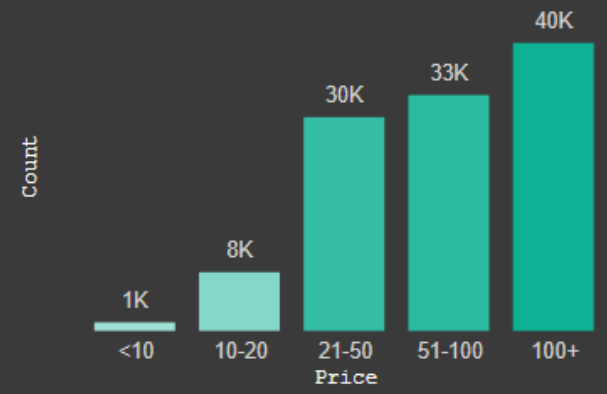
Where are most customers from ?



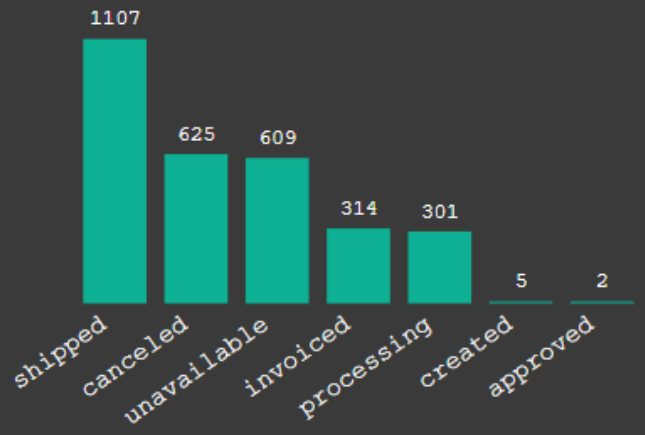
How do they pay ?



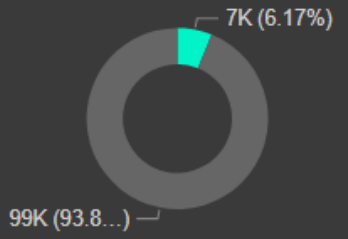
How much are they spending ?



Order Status



Share of Late orders



Category	Time_in_Days
Min_Approval_Time	0.00
Max_Approval_Time	188.00
Avg_Approval_Time	0.52
Max_Carrier_Time	205.00
Avg_Carrier_Time	9.30

There are products that are being sold at different costs.

The cost of products ranges from $0.85 to $6735, with an average of **$120.65**

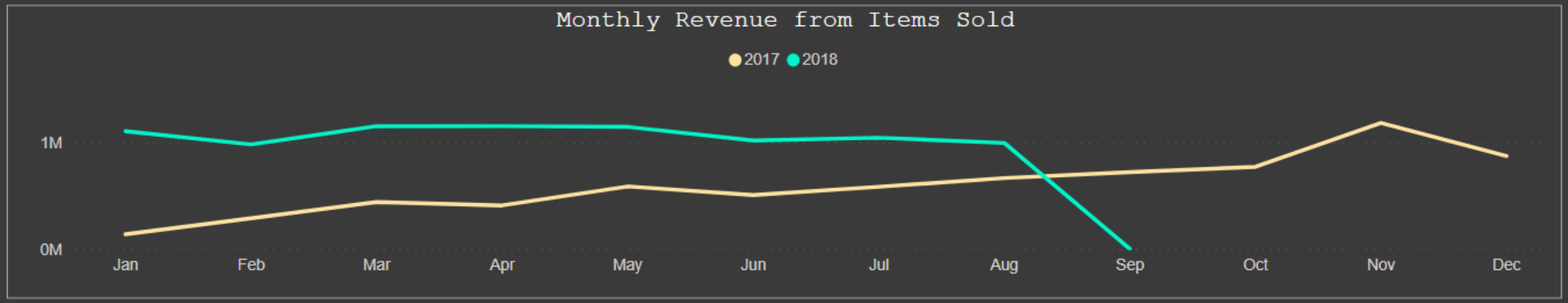
Most of the products in the order_items table cost more than $100

For every order, multiple payments can be made. This is captured by the payment_sequential column.

Most of the customers make between **1 and 3** payments per order.

Revenue

Order & Revenue Analysis



Product categories by Revenue

Product_Category_Name	2017_Revenue	2018_Revenue	%GT YoY_Increase
health_beauty	$616,877.59	$1,031,035.52	27.82%
bed_bath_table	$796,732.17	$912,234.2	7.76%
computers_accessories	$694,732.44	$874,414.67	12.07%
watches_gifts	$564,547.92	$853,699.32	19.42%
sports_leisure	$635,624.54	$743,247.93	7.23%
furniture_decor	$672,548.19	$741,109.33	4.61%
housewares	$387,633.94	$696,194.86	20.73%
auto	$389,695.61	$451,849.58	4.17%
office_furniture	$291,498.79	$353,280.52	4.15%
garden_tools	$470,233.5	$351,819.11	-7.95%

From the product categories by revenue analysis, we can conclude the following:

For the year of 2017, the category, 'bed_bath_table' generated the highest amount of revenue. But for 2018, the category 'health_beauty' is the golden category and is followed by 'bed_bath_table'.

However, we can also see that the category 'garden_tools' has seen a **decrease** in revenue from items sold in 2018 compared to 2017. It also appears to be the only category among 2018's top 10 that has decreased in sales.

The categories 'health_beauty' , 'housewares' and 'watches_gifts' seem to be showing promising YoY increase in revenue.

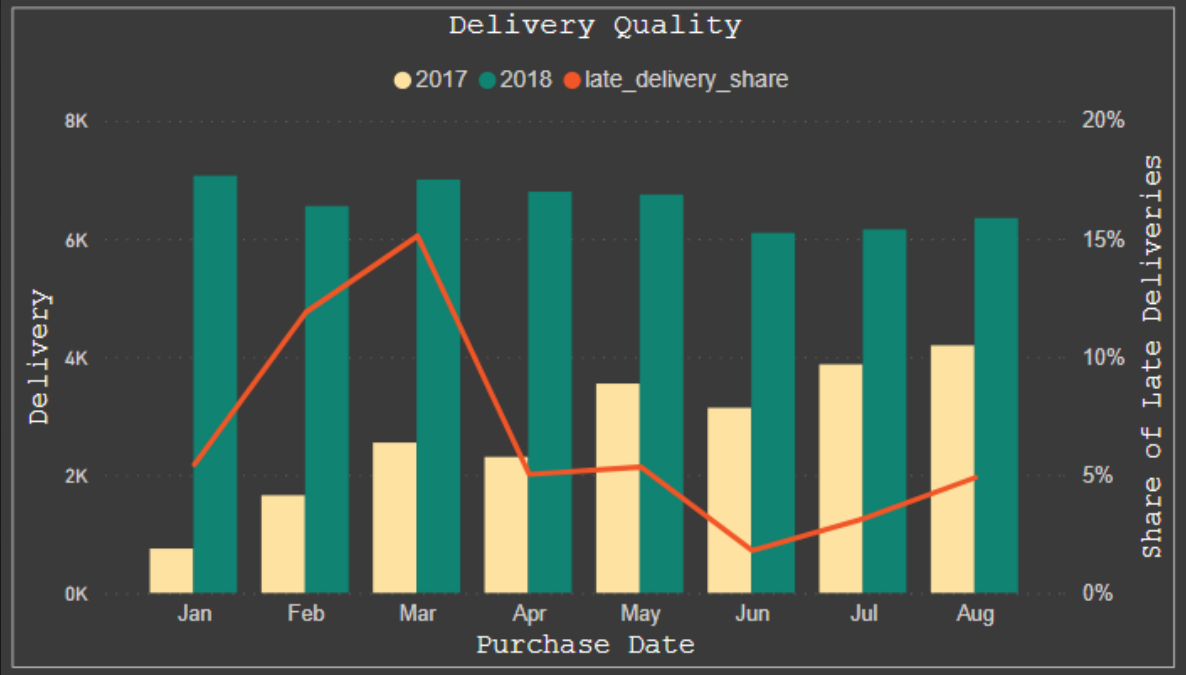
Delivery Quality

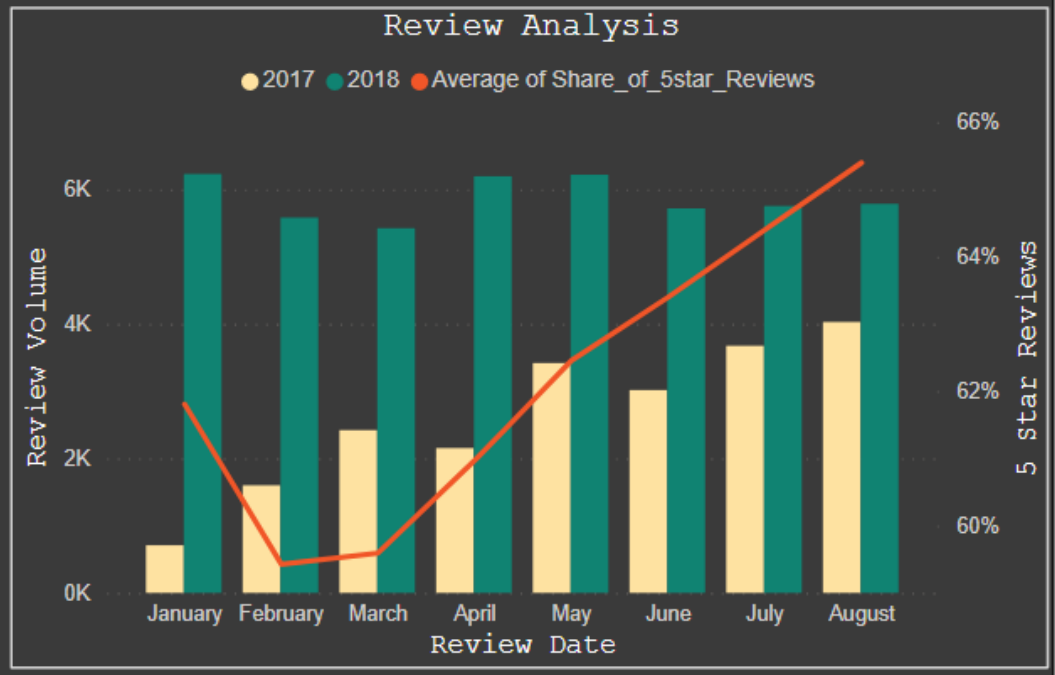
Delivery & Product Quality Analysis

93K
Reviews Volume

56K
5Star Reviews

60.57%
Shareof 5star reviews





The share of late deliveries are calculated **based on** orders with '**delivered**' status.

This analysis shows that from late 2017 to Q1 of 2018, there was a general increase in the share of late deliveries.

The total orders volume is seen to have an increasing trend over the years with the share of late deliveries showing slight increases but generally stabilizing by the third quarter.

For the above analysis, those reviews for which the order_status is '**canceled**' are **not** considered.

From the results, it can be seen that there is an overall increasing trend in the number of reviews being created and an average of 60% of reviews have 5 stars.

Key Findings & Recommendations

- Most of Maven Amazonas' customers use credit card for their purchases. So, money back with credit card could be a way to ensure customer retention.
- 7% orders are late. Identify & Incentivise delivery teams to encourage better performance.
- Running a discount on products from the garden tools category could lead to increase in sales & revenue. May be combo offers with products from houseware category is a direction to consider.
- Prepare ahead for in time deliveries especially for Q3 of 2018 to meet the order volume load.

Database Issues

- There were records with reviews for which the review creation date was earlier than the order delivery date
- The attribute shipping limit date in the order items table does not add much business value
- There are records in order payments with payment instalment number 0 for single payments
- There are some orders that have a delivered to carrier date that is before order approved date. Looks like the month & day number have been mixed up for these entries.
- There are reviews for orders with all possible values of order status including ‘canceled’
- Tables lack constraints & relationships. This has been corrected by implementing the necessary changes in the database.

Queries to answer key business questions

```sql
SELECT
    YEAR(order_purchase_date) AS Year,
    MONTHNAME(order_purchase_date) AS MonthName,
    COUNT(DISTINCT order_payments.order_id) AS Count_of_Items_Delivered,
    SUM(payment_value) AS Monthly_Revenue
FROM
    order_payments
        LEFT JOIN
    orders ON order_payments.order_id = orders.order_id
        AND order_status <>'canceled'
GROUP BY 1 , 2
ORDER BY Year , MONTH(order_purchase_date);
```

```sql
SELECT
    Productscomplete.productcategory AS Product_Category_Name,
    SUM(order_payments.payment_value) AS Revenue_in_2018,
    YEAR(orders.order_purchase_date) AS Year
FROM
    order_payments
        LEFT JOIN
    orders ON orders.order_id = order_payments.order_id
        LEFT JOIN
    order_items ON order_payments.order_id = order_items.order_id
        LEFT JOIN
    Productscomplete ON order_items.product_id =
productscomplete.product_id
WHERE
    YEAR(orders.order_purchase_date) = '2018'
AND orders.order_status <> 'canceled'
GROUP BY productscomplete.productcategory
ORDER BY Revenue_in_2018 DESC
Limit 10;
```

```sql
SELECT
    productscomplete.productcategory AS Product_Category_Name,
    SUM(CASE
        WHEN YEAR(orders.order_purchase_date) = '2018' THEN order_payments.payment_value
        ELSE NULL
    END) AS `2018_Revenue`,
    SUM(CASE
        WHEN YEAR(orders.order_purchase_date) = '2017' THEN order_payments.payment_value
        ELSE NULL
    END) AS `2017_Revenue`,
    SUM(CASE
        WHEN YEAR(orders.order_purchase_date) = '2018' THEN order_payments.payment_value
        ELSE NULL
    END) - SUM(CASE
        WHEN YEAR(orders.order_purchase_date) = '2017' THEN order_payments.payment_value
        ELSE NULL
    END) AS `YoY_Increase`
FROM
    order_payments
        LEFT JOIN
    orders ON orders.order_id = order_payments.order_id
        LEFT JOIN
    order_items ON order_payments.order_id = order_items.order_id
        LEFT JOIN
    productscomplete ON order_items.product_id = productscomplete.product_id
WHERE
    Orders.order_status <> 'canceled'
        AND productcategory IN ('health_beauty' , 'bed_bath_table',
        'computers_accessories',
        'watches_gifts',
        'furniture_decor',
        'sports_leisure',
        'housewares',
        'auto',
        'office_furniture',
        'garden_tools')
GROUP BY productscomplete.productcategory
ORDER BY 2 DESC;
```

```sql
CREATE TEMPORARY TABLE orders_with_latedelivery_info
SELECT *,
    (CASE
        WHEN order_delivered_customer_date >
order_estimated_delivery_date THEN 1
        ELSE 0
    END) AS late_delivery
FROM  orders;
```

```sql
SELECT
    YEAR(order_purchase_date) AS Year,
    MONTHNAME(order_purchase_date) AS Month,
    COUNT(DISTINCT order_id) AS Total_Orders,
    count(case when order_status='delivered' then 1 else
null end) as Total_Delievered_Orders,
    COUNT(CASE
        WHEN late_delivery = 1 THEN late_delivery
        ELSE NULL
    END) AS Total_Late_Orders,
    CONCAT(ROUND(COUNT(CASE
                        WHEN late_delivery = 1 THEN
late_delivery
                        ELSE NULL
                    END) / count(case when
order_status='delivered' then 1 else null end) * 100,
                    2),
            '%') AS Late_Orders_Percentage
FROM
    orders_with_latedelivery_info
-- where order_status='delivered'
GROUP BY YEAR(order_purchase_date) ,
MONTH(order_purchase_date) , MONTHNAME(order_purchase_date)
ORDER BY YEAR(order_purchase_date) ,
MONTH(order_purchase_date) ,
MONTHNAME(order_purchase_date);
```

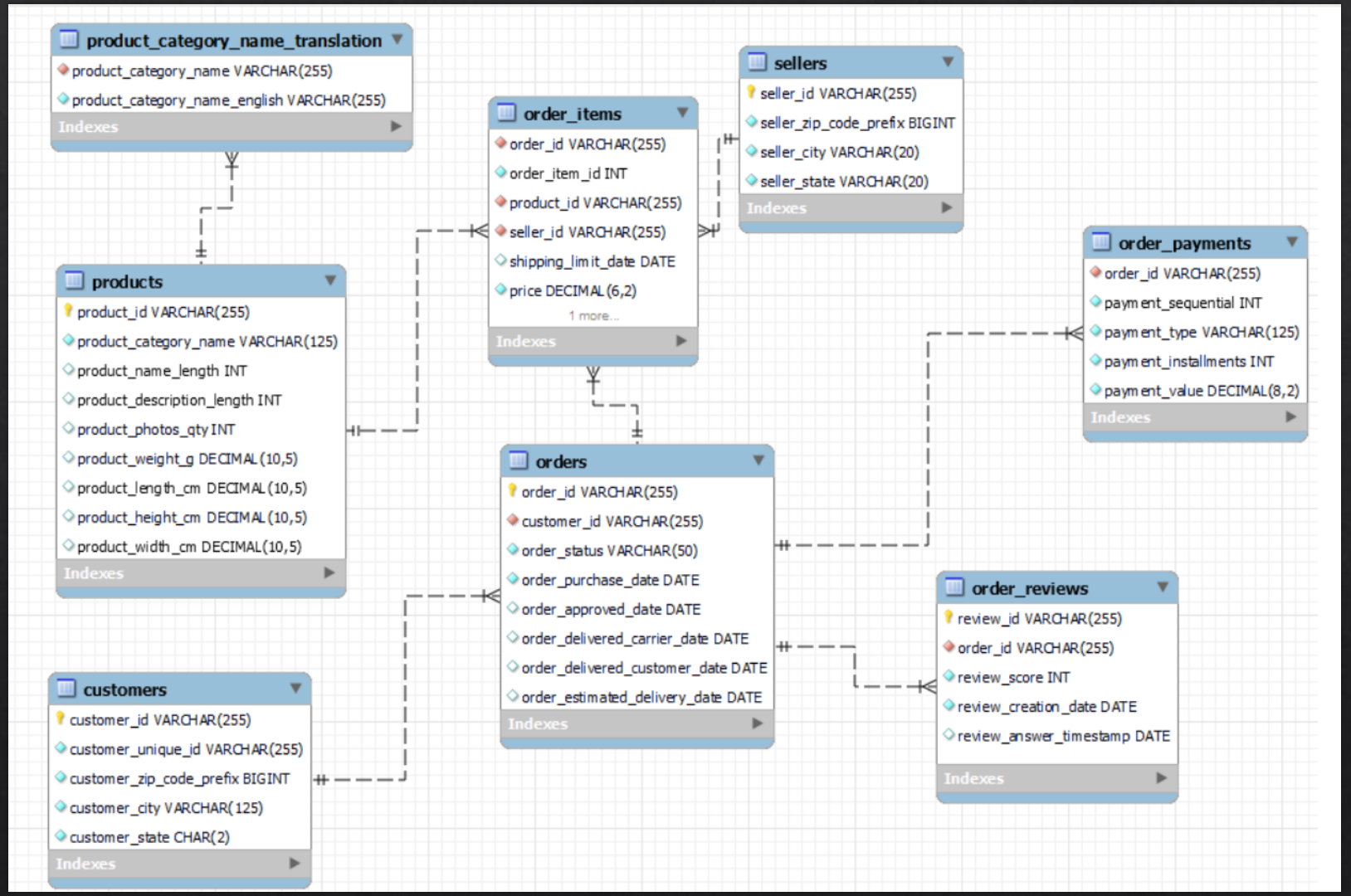
```sql
select order_status, count(distinct review_id) as count_of_reviews
from (select order_reviews.order_id, order_reviews.review_id,
order_reviews.review_score,
orders.order_status
from order_reviews left join orders
on order_reviews.order_id = orders. order_id) ordersQA
group by order_status;
```

```sql
SELECT
YEAR(order_purchase_date) AS yr,
MONTH(order_purchase_date) AS mo,
    MONTHNAME(order_purchase_date) AS mo_name,
    COUNT(DISTINCT orders.order_id) AS order_volume,
    COUNT(DISTINCT review_id) AS review_volume,
    COUNT(DISTINCT CASE WHEN review_score = 5 THEN review_id else null end) AS
review_5star_volume

FROM orders
LEFT JOIN order_reviews
ON orders.order_id = order_reviews.order_id
WHERE
order_reviews.review_creation_date > orders.order_delivered_customer_date
    AND orders.order_status <> 'canceled'

GROUP BY 1,2,3
```

Final EER Diagram




Thank you