

BUSINESS CASE

Prepared by: Kontham Tarun

Date: Aug 4, 2025

Executive Summary

This report provides an in-depth analysis of Target's retail operations in Brazil, based on a dataset of 100,000 orders placed between 2016 and 2018. The document explores key business dimensions such as order lifecycle, pricing trends, payment and freight efficiency, customer locations, product categories, and customer feedback. Through a series of SQL queries and data exploration, the analysis offers actionable insights into Target's operational performance, customer behavior, and overall market engagement. This report serves as a valuable resource for understanding business drivers, identifying improvement areas, and supporting data-driven decision-making in retail strategy.

Strategic Context

Problem

Target's retail operations in Brazil faced challenges in optimizing order fulfillment, managing shipping delays, inconsistent customer satisfaction, and varied payment behaviors. With limited visibility into how these factors interacted, inefficiencies likely affected customer experience, operational costs, and overall profitability.

Opportunity

With access to a rich dataset of 100,000 orders from 2016–2018, there is a clear opportunity to extract actionable insights across multiple dimensions: order processing speed, pricing strategies, customer demographics, shipping performance, and product feedback. Leveraging this data enables informed decision-making that can enhance operational efficiency, improve customer engagement, and drive revenue growth.

Solution

This analysis uses SQL-driven queries and exploratory data analysis to uncover trends and performance gaps in Target's Brazil operations. It identifies key patterns in payment methods, freight performance, customer location impact, product preferences, and reviews. These insights can inform strategic interventions such as:

- Streamlining logistics and order handling
- Personalizing marketing based on regional behavior
- Adjusting pricing and promotion strategies
- Enhancing customer support for high-impact touchpoints

By implementing these data-backed recommendations, Target can improve customer satisfaction, reduce costs, and strengthen its competitive position in emerging markets like Brazil.

1.) Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

- Data type of all columns in the "customers" table.

```
select column_name,data_type from `evident-syntax-452811-b5.Business_case_1.INFORMATION_SCHEMA.COLUMNS` WHERE TABLE_NAME="customers"
```

Query results

Job information		Results	Chart	JSON	Execution detail
Row	column_name	data_type			
1	customer_id	STRING			
2	customer_unique_id	STRING			
3	customer_zip_code_prefix	INT64			
4	customer_city	STRING			
5	customer_state	STRING			

- Get the time range between which the orders were placed

```
select min(order_purchase_timestamp) as start_date,max(order_purchase_timestamp) as end_date  
from `Business_case_1.orders`;
```

Query results

Job information		Results	Chart	JSON	Execution details	Execution graph
Row	start_date	end_date				
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC				

- Insights - All the orders that were placed are between 2016 september and 2018 october.
- Recommendation's - NIL

- Count the Cities & States of customers who ordered during the given period.

```
select count(distinct(customer_city)) as total_cities , count(distinct(customer_state)) as total_states from
`Business_case_1.customers`;
```

Query results

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	total_cities	total_states			
1	4119	27			

2.) In depth - exploration

- Is there a growing trend in the no. of orders placed over the past years?

```
select Extract(year from order_purchase_timestamp) as order_year , count(*) as total_orders from
`Business_case_1.orders`  

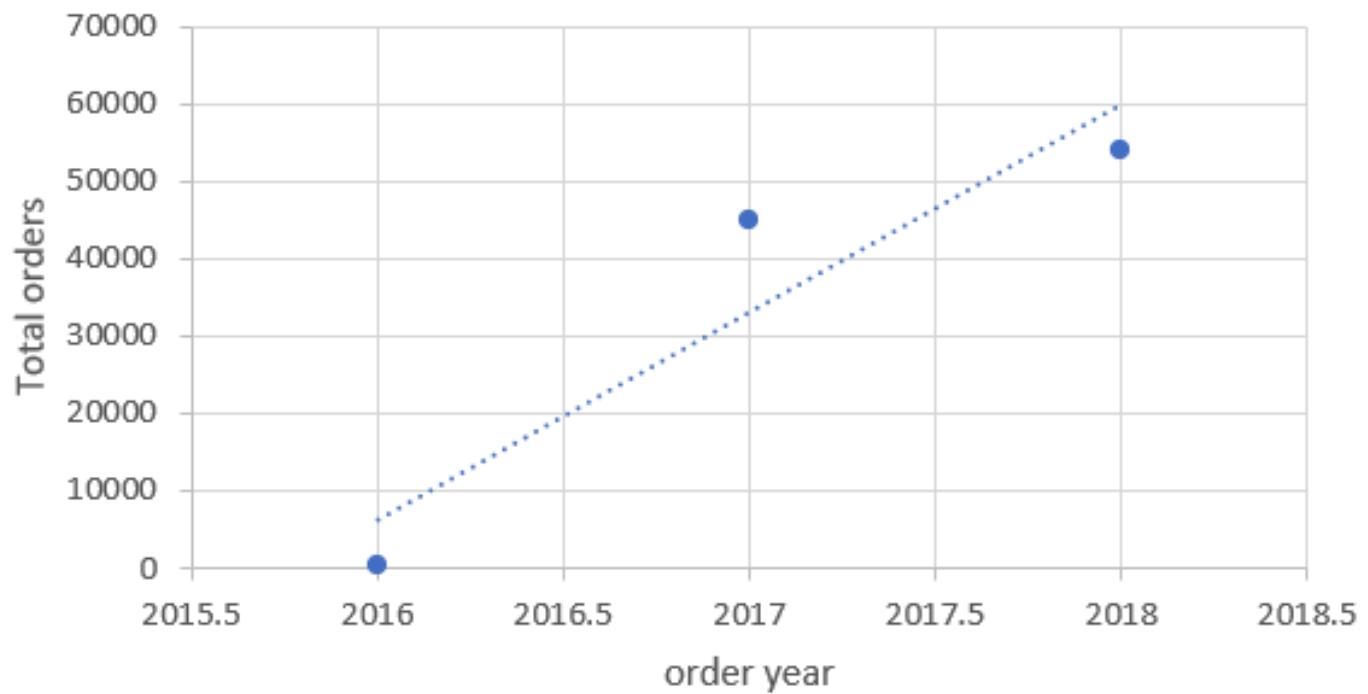
group by order_year  

order by total_orders;
```

Query results

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	order_year	total_orders			
1	2016	329			
2	2017	45101			
3	2018	54011			

trend line



Insights -

- Yes, there is a growing trend as we can see the no. of orders placed over the past three years from 2016 to 2018 have been increased
- **Recommendations - NIL**

- Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

```
select FORMAT_DATE('%B',order_purchase_timestamp) as month, count(*) as total_orders from `Business_case_1.orders`  
group by month  
order by total_orders desc;
```

Row	month	total_orders
1	August	10843
2	May	10573
3	July	10318
4	March	9893
5	June	9412
6	April	9343
7	February	8508
8	January	8069
9	November	7544
10	December	5674
11	October	4959
12	September	4305

Insights -

- Highest no. of orders placed in month of August , May, July.
- Lowest no. of orders where placed in month of september and october.

Recommendations -

- Analyze which products still perform well in low months and promote those more heavily.
- Introduce seasonally relevant products or bundles to align with customer needs during low sale periods.
- Launch time-limited discounts or combo offers during slow months to incentivize purchases.

- During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
 - 0-6 hrs : Dawn
 - 7-12 hrs : Mornings
 - 13-18 hrs : Afternoon
 - 19-23 hrs : Night

with final as(

```
(select 1, customer_id, order_purchase_timestamp, extract(hour from order_purchase_timestamp) as hr,
case when extract(hour from order_purchase_timestamp) between 0 and 6 then "Dawn"
      when extract(hour from order_purchase_timestamp) between 7 and 12 then "Mornings"
      when extract(hour from order_purchase_timestamp) between 13 and 18 then "Evening"
      when extract(hour from order_purchase_timestamp) between 19 and 23 then "Night"
      else null end as time_of_the_day from `Business_case_1.orders` ))
select time_of_the_day, count(time_of_the_day) as total_orders from final
group by time_of_the_day
order by 1;
```

Query results

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	// time_of_the_day ▾	// total_orders ▾ //			
1	Dawn	5242			
2	Evening	38135			
3	Mornings	27733			
4	Night	28331			

Insights -

- Brazilian's place most numbers of order in the evening

Recommendations -

- Run key marketing campaigns, offers, and push notifications during evening hours to capitalize on peak user activity.
- Align customer support and delivery planning with peak ordering times—more staff in evenings and less during dawn.
- Investigate if low dawn activity is due to limited service availability, poor UX, or natural customer behavior. Explore tactics like scheduled orders or dawn-exclusive deals to increase engagement.

3.) In depth - exploration Evolution of e-commerce in the Brazil region

- Get the month on month no. of orders placed in each state.

with final as

```
(select o.customer_id ,c.customer_state ,o.order_purchase_timestamp,  
     extract(month from o.order_purchase_timestamp) as month,  
     extract(year from o.order_purchase_timestamp) as year,  
     row_number() over(order by order_purchase_timestamp) as row_num  
  from `Business_case_1.orders` o inner join `Business_case_1.customers` c  
    on c.customer_id = o.customer_id))
```

```
select customer_state , year , month,  
       count(row_num) as total_orders from final group by 1,2,3 order by 1;
```

Query results

Save results ▾ Open in ▾

Job information		Results	Chart	JSON	Execution details	Execution graph
Row	customer_state	year	month	total_orders		
1	AC	2017	1	2		
2	AC	2017	2	3		
3	AC	2017	3	2		
4	AC	2017	4	5		
5	AC	2017	5	8		
6	AC	2017	6	4		
7	AC	2017	7	5		
8	AC	2017	8	4		
9	AC	2017	9	5		
10	AC	2017	10	-		

Results per page: 50 ▾ 1 – 50 of 565 |< < > >|

Insights -

- Top 10 performing states dominate the order volume and these states have larger customer bases or strong logistics
- overall monthly there is visible fluctuation in no. of orders per year

Recommendations -

- Invest in region-specific campaigns and loyalty programs for these top-performing states to retain and expand user base.
- Improve Awareness in Underperforming States.
- Forecast for Seasonal Demand

- How are the customers distributed across all the states?

```
select customer_state,
       count(distinct(customer_id)) as total_customers
  from `Business_case_1.customers`
 group by customer_state
 order by 1;
```

Query results Save results ▾

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	customer_state		total_customers		
1	AC		81		
2	AL		413		
3	AM		148		
4	AP		68		
5	BA		3379		
6	CE		1336		
7	DF		2139		
8	ES		2032		
9	GO		2019		
10	MA		747		
11	MG		11637		
12	MS		715		
13	MT		907		
14	PA		975		
15	PB		535		
16	PE		1653		
17	DL		405		

Results per page: 50 ▾ 1 – 27 of 27

Insights -

- With **41,746 customers**, SP accounts for a significant portion of the customer base far ahead of any other state.
- SP,RJ,MG these three states are the **core market hubs**, combining to contribute over **50%** of the total customers.
- States like PR, SC, RS, BA, and DF have **2,000–5,000 customers** each.

Recommendations -

- Increase premium services in high performing states
- For states with low customers audit logistic coverage and price related issues.
- Perform campaigns to know the demands in states with low customers

4.) Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight and others.

- Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

with final as

```
select extract(year from o.order_purchase_timestamp) as year,  
extract(month from o.order_purchase_timestamp) as month,  
p.payment_value  
from `Business_case_1.orders` o inner join `Business_case_1.payments` p  
on o.order_id = p.order_id  
where extract(year from o.order_purchase_timestamp) between 2017 and 2018 and extract(month from  
o.order_purchase_timestamp) between 1 and 8),  
  
old_ttl as(  
select year ,sum(payment_value) as total_2017 from final where year = 2017 group by year),  
  
new_ttl as(  
select year ,sum(payment_value) as total_2018 from final where year = 2018 group by year)  
  
select total_2017,total_2018,round(((total_2018 - total_2017 )/o.total_2017)*100,2) as percen_increase from old_ttl o join new_ttl n on  
o.year!=n.year;
```

Query results

Job information		Results	Chart	JSON	Execution details
Row	total_2017	total_2018	percen_increase		
1	3669022.120000...	8694733.840000...	136.98		

Insights -

- With **41,746 customers**, SP accounts for a significant portion of the customer base far ahead of any other state.
- SP,RJ,MG these three states are the **core market hubs**, combining to contribute over **50%** of the total customers.
- States like PR, SC, RS, BA, and DF have **2,000–5,000 customers** each.

Recommendations -

- Increase premium services in high performing states
- For states with low customers audit logistic coverage and price related issues.

- Perform campaigns to know the demands in states with low customers

- Calculate the Total & Average value of order price for each state.

```
select c.customer_state,round(sum(p.payment_value),2) as total_payment,round(avg(p.payment_value),2) as avrg_payment
from `Business_case_1.customers` c left join `Business_case_1.orders` o on c.customer_id=o.customer_id
left join `Business_case_1.payments` p on p.order_id=o.order_id
group by 1
order by 1
```

Query results

Job information		Results	Chart	JSON	Execution details
Row	customer_state	total_payment		avrg_payment	
1	AC	19680.62		234.29	
2	AL	96962.06		227.08	
3	AM	27966.93		181.6	
4	AP	16262.8		232.33	
5	BA	616645.82		170.82	
6	CE	279464.03		199.9	
7	DF	355141.08		161.13	
8	ES	325967.55		154.71	
9	GO	350092.31		165.76	
10	MA	152523.02		198.86	
11	MG	1872257.26		154.71	
12	MS	137534.84		186.87	
13	MT	187029.29		195.23	
14	PA	218295.85		215.92	
15	PB	141545.72		248.33	
16	PE	324850.44		187.99	

Insights -

- PB,RO,AC show strong average order value

- RC,AC,TO,RO have low total value

Recommendations -

- Strengthen logistics & retention in SP , RJ and MG
- Improve premium campaigns in PB,RO AN ac to maximise per user value
- Launch low cost pilots in RC,AC,TO and RO

- Calculate the Total & Average value of order FREIGHT for each state.

```
select c.customer_state ,
       round(sum(freight_value),2) as total_freight_price ,
       round(avg(freight_value),2) as avrg_freight_price
  from `Business_case_1.customers` c left join `Business_case_1.orders` o
    on c.customer_id = o.customer_id
   left join `Business_case_1.orders_items` t
    on t.order_id = o.order_id
   group by 1
  order by 2,3;
```

Row	customer_state	total_freight_price	avrg_freight_price
1	RR	2235.190000000...	42.98442307692...
2	AP	2788.500000000...	34.00609756097...
3	AC	3686.749999999...	40.07336956521...
4	AM	5478.89	33.20539393939...
5	RO	11417.38	41.06971223021...
6	TO	11732.67999999...	37.24660317460...
7	SE	14111.46999999...	36.65316883116...
8	AL	15914.58999999...	35.84367117117...
9	RN	18860.10000000...	35.65236294896...
10	MS	19144.03000000...	23.37488400488...
11	PI	21218.20000000...	39.14797047970...
12	PB	25719.73000000...	42.72380398671...
13	MT	29715.43000000...	28.16628436018...
14	MA	31523.77000000...	38.25700242718...

Insights -

- RJ, MG, and PR possess the highest freight value.
- PR, MG, and DF exhibit the lowest average freight value.
- RR, PB, and RO show the highest average freight value, likely due to their remoteness.
- Mid-tier states with manageable freight values include GO, ES, and PE.

Recommendations -

- Leverage efficiency to run free shipping in low freight areas
- Adjust pricing or shipping policies based on average freight per region

5.) Analysis based on sales, freight and delivery time

- Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order.

SELECT

```
order_purchase_timestamp,  
order_estimated_delivery_date,  
order_delivered_customer_date,  
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS time_to_deliver,  
DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY) AS diff_estimated_delivery  
FROM `Business_case_1.orders` WHERE order_delivered_customer_date IS NOT NULL;
```

Query results

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	order_purchase_timestamp	order_estimated_delivery_date	order_delivered_customer_date	time_to_deliver	diff_estimated_d...
1	2016-10-03 21:01:41 UTC	2016-11-25 00:00:00 UTC	2016-11-08 10:58:34 UTC	35	-16
2	2016-10-09 15:39:56 UTC	2016-12-08 00:00:00 UTC	2016-11-09 14:53:50 UTC	30	-28
3	2018-02-19 19:48:52 UTC	2018-03-09 00:00:00 UTC	2018-03-21 22:03:51 UTC	30	12
4	2016-09-15 12:16:38 UTC	2016-10-04 00:00:00 UTC	2016-11-09 07:47:38 UTC	54	36
5	2016-10-04 15:44:44 UTC	2016-11-24 00:00:00 UTC	2016-11-17 08:38:57 UTC	43	-6
6	2016-10-04 12:53:17 UTC	2016-11-24 00:00:00 UTC	2016-11-09 13:37:38 UTC	36	-14
7	2016-10-04 16:05:29 UTC	2016-11-24 00:00:00 UTC	2016-11-03 10:08:51 UTC	29	-20
8	2016-10-04 18:52:56 UTC	2016-11-24 00:00:00 UTC	2016-11-04 17:54:00 UTC	30	-19
9	2016-10-04 09:59:03 UTC	2016-11-24 00:00:00 UTC	2016-11-18 08:51:07 UTC	44	-5
10	2016-10-04 21:54:52 UTC	2016-11-24 00:00:00 UTC	2016-12-12 20:31:54 UTC	68	18
11	2016-10-04 17:08:39 UTC	2016-11-24 00:00:00 UTC	2016-11-21 10:38:51 UTC	47	-2
12	2016-10-05 12:44:09 UTC	2016-11-25 00:00:00 UTC	2016-11-17 19:53:49 UTC	43	-7
13	2016-10-05 08:04:21 UTC	2016-11-25 00:00:00 UTC	2016-11-17 16:23:47 UTC	43	-7
14	2016-10-05 11:05:52 UTC	2016-11-25 00:00:00 UTC	2016-11-14 15:03:36 UTC	40	-10
15	2016-10-04 19:25:29 UTC	2016-11-28 00:00:00 UTC	2016-11-17 13:07:53 UTC	43	-10

Insights -

- Orders with low delivery time indicates strong logistics
- Orders with long delivery time shows some potential issues like inventory unavailability and delivery services
- Negative values in difference in estimated delivery indicates faster delivery time and values with 0 indicates on time delivery , values with positive indicates slow delivery

Recommendations -

- Investigate the root causes of late deliveries, such as issues with courier services.
- Implement a buffer delivery system in areas where delays are frequent to establish better expectations.
- Keep track of the delivery timeframes of your logistics partners.

- Find out the top 5 states with the highest & lowest average freight value.

```
WITH avrg AS (
```

```
SELECT
```

```
c.customer_state,
```

```
AVG(freight_value) AS avrg_freight_price
```

```
FROM `Business_case_1.customers` c
```

```
LEFT JOIN `Business_case_1.orders` o ON c.customer_id = o.customer_id
```

```
LEFT JOIN `Business_case_1.orders_items` t ON t.order_id = o.order_id
```

```
GROUP BY c.customer_state
```

```
),
```

```
rnk AS (
```

```
SELECT
```

```
customer_state,
```

```
avrg_freight_price,
```

```
ROW_NUMBER() OVER(ORDER BY avrg_freight_price DESC) AS highest_val,
```

```
ROW_NUMBER() OVER(ORDER BY avrg_freight_price ASC) AS lowest_val
```

```
FROM avrg
```

```
),
```

```
lowest_avg AS (
```

```
SELECT
```

```
customer_state AS lowest_state,
```

```
avrg_freight_price AS lowest_average,
```

```
ROW_NUMBER() OVER () AS rn
```

```
FROM rnk
```

```
WHERE lowest_val BETWEEN 1 AND 5
```

```
),
```

```

highest_avg AS (
    SELECT customer_state AS highest_state,
        avrg_freight_price AS highest_average,
        ROW_NUMBER() OVER () AS rn
    FROM rnk
    WHERE highest_val BETWEEN 1 AND 5
)

```

SELECT lowest_state,lowest_average,highest_state,highest_average FROM lowest_avg JOIN highest_avg USING (rn);

Query results

Job information		Results	Chart	JSON	Execution details	Execution graph
Row	lowest_state	lowest_average	highest_state	highest_average		
1	SP	15.14727539041...	RR	42.98442307692...		
2	PR	20.53165156794...	PB	42.72380398671...		
3	MG	20.63016680630...	RO	41.06971223021...		
4	RJ	20.96092393168...	AC	40.07336956521...		
5	DF	21.04135494596...	PI	39.14797047970...		

Insights -

- RR,PB and RO have the highest average freight value which is costing more for transportation than any other state across brazil

Recommendations -

- Optimize packaging use bulk packaging where possible
- Consolidate shipments
- Build-long term relationship to negotiate carriers
- Use transportation management systems (TMS) for better planning and visibility
- Use freight audit and payment software to detect overcharges and billing errors

- Find out the top 5 states with the highest & lowest average delivery time value.

```
(select c.customer_state,round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp),2)) as avg_delivery_time,
      "Highest" as Category
   from `Business_case_1.customers` c inner join `Business_case_1.orders` o on c.customer_id=o.customer_id
  where o.order_delivered_customer_date is not null and o.order_purchase_timestamp is not null
  group by c.customer_state
  order by avg_delivery_time desc limit 5)

union all

(select c.customer_state,round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp),2)) as avg_delivery_time,
      "Lowest" as Category
   from `Business_case_1.customers` c inner join `Business_case_1.orders` o on c.customer_id=o.customer_id
  where o.order_delivered_customer_date is not null and o.order_purchase_timestamp is not null
  group by c.customer_state
  order by avg_delivery_time asc limit 5)
```

Query results

Job information		Results	Chart	JSON	Execution details	Ex
Row	customer_state	avg_delivery_time		Category		
1	SP	8.3		Lowest		
2	PR	11.53		Lowest		
3	MG	11.54		Lowest		
4	DF	12.51		Lowest		
5	SC	14.48		Lowest		
6	RR	28.98		Highest		
7	AP	26.73		Highest		
8	AM	25.99		Highest		
9	AL	24.04		Highest		
10	PA	23.32		Highest		

Insights -

- Though the average is same the variation lies in the last 3 to 4 digits
- States with higher average delivery time takes more time deliver the order compare to the states with lowest average delivery time

Recommendations -

- Focus on logistics to improve the delivery time for low performing states
- Implement automated order processing to reduce handling delays.
- Optimize cut-off times for same/next-day delivery.
- Add buffer time in the estimated delivery date to avoid customer disappointment.

- Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```
select c.customer_state,round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY)),2) as delivery_speed
from `Business_case_1.customers` c inner join `Business_case_1.orders` o on c.customer_id=o.customer_id
where o.order_delivered_customer_date is not null and o.order_estimated_delivery_date is not null and
date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY)>0
group by c.customer_state
order by delivery_speed desc limit 5
```

Query results

Job information		Results	Chart	JSON
Row	customer_state	delivery_speed		
1	RR	23.75		
2	AP	21.87		
3	AC	21.54		
4	AM	20.28		
5	RO	20.03		

Insights -

- (AM) leads with deliveries almost 2 days earlier than the estimated date.
- (AC), (AP), and (PA) also show faster deliveries but with smaller margins.
- Customers in AM, AC, AP, and PA are likely having a very positive delivery experience, which could lead to higher satisfaction, loyalty, and repeat purchases.

Recommendations -

- Highlight fast delivery times in these top states through localized ads, emails, or promotion's
- Study what is working well in AM, AC, AP, and PA (warehousing, courier partners, process timings) and apply similar strategies to other slower states.

6.) Analysis based on the payments

- Find the month on month no. of orders placed using different payment types.

SELECT

```
FORMAT_DATE('%Y-%m', o.order_purchase_timestamp) AS order_month,
```

```
p.payment_type,
```

```
COUNT(DISTINCT o.order_id) AS total_orders
```

FROM

```
`Business_case_1.orders` o
```

JOIN

```
`Business_case_1.payments` p
```

ON

```
o.order_id = p.order_id
```

GROUP BY

```
order_month, p.payment_type
```

ORDER BY

```
order_month, p.payment_type;
```

Query results

Job information		Results	Chart	JSON	Execution details	Export
Row	order_month	payment_type		total_orders		
1	2016-09	credit_card		3		
2	2016-10	UPI		63		
3	2016-10	credit_card		253		
4	2016-10	debit_card		2		
5	2016-10	voucher		11		
6	2016-12	credit_card		1		
7	2017-01	UPI		197		
8	2017-01	credit_card		582		
9	2017-01	debit_card		9		
10	2017-01	voucher		33		
11	2017-02	UPI		398		
12	2017-02	credit_card		1347		
13	2017-02	debit_card		13		
14	2017-02	voucher		69		

Insights -

- Increasing Trend in Digital Payments, Orders via UPI and credit cards have shown a sharp increase from September 2016 to January 2017.
- Low Adoption of Debit Cards & Vouchers and there is a clear customer preference for credit cards and UPI, likely due to cashback, ease, or offers.

Recommendation's :

- Promote high-performing payment modes and analyse low usage modes.
- Identify repeat customers using UPI or credit card and introduce a loyalty program for continued use.
- For first-time users of UPI, offer bonus cashback on 2nd and 3rd transactions.
- Find the no. of orders placed on the basis of the payment installments that have been paid.

```
select payment_installments,
       count(distinct(order_id)) as total_orders
  from `Business_case_1.payments`
 group by 1 order by 1,2
```

Row	payment_installments	total_orders
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644
11	10	5315
12	11	23
13	12	133
14	13	16
15	14	15
16	15	74

Insights -

- Majority of Customers Prefer Fewer Installments A .very high number of orders (49,060) are paid using 1 instalment

- Higher Installment Options See Steep Drop , beyond 5 installments, the number of orders drops significantly.

Recommendations -

- Promote 1–3 Instalment Options More Actively. Focus marketing or promotional offers on these most-used options, as customers are more likely to convert.
- Review Long-Term Instalment Options, Evaluate the cost-benefit of offering instalment plans beyond 6–7 months. Consider limiting or restructuring these plans to reduce financial risk.

