ALL SQL QUERIES

CREATE TABLE olist\_customers\_dataset (

customer\_id TEXT,

customer\_unique\_id TEXT,

customer\_zip\_code\_prefix TEXT,

customer\_city TEXT,

customer\_state TEXT

);

select \* from olist\_customers\_dataset

-------------------------------------

create table geolocation\_dataset(geolocation\_zip\_code\_prefix int,

geolocation\_lat numeric (10,2),

geolocation\_lng numeric (10,2),

geolocation\_city varchar (50),

geolocation\_state varchar

);

select \* from geolocation\_dataset;

----------------------------------

CREATE TABLE olist\_order\_items\_dataset (

order\_id TEXT,

order\_item\_id INT,

product\_id TEXT,

seller\_id TEXT,

shipping\_limit\_date DATE,

price NUMERIC(10,2),

freight\_value NUMERIC(10,2)

);

select \* from olist\_order\_items\_dataset

---------------------------------------

CREATE TABLE olist\_order\_payments\_dataset (

order\_id TEXT,

payment\_sequential INT,

payment\_type TEXT,

payment\_installments INT,

payment\_value NUMERIC(10,2)

);

select \* from olist\_order\_payments\_dataset

----------------------------------------

CREATE TABLE olist\_order\_reviews\_dataset (

review\_id TEXT,

order\_id TEXT,

review\_score INT,

review\_comment\_title TEXT,

review\_comment\_message TEXT,

review\_creation\_date TIMESTAMP,

review\_answer\_timestamp TIMESTAMP

);

select \* from olist\_order\_reviews\_dataset

------------------------------------------

CREATE TABLE olist\_orders\_dataset (

order\_id TEXT,

customer\_id TEXT,

order\_status TEXT,

order\_purchase\_timestamp TIMESTAMP,

order\_approved\_at TIMESTAMP,

order\_delivered\_carrier\_date TIMESTAMP,

order\_delivered\_customer\_date TIMESTAMP,

order\_estimated\_delivery\_date TIMESTAMP

);

select \* from olist\_orders\_dataset

------------------------------------------

CREATE TABLE olist\_products\_dataset (

product\_id TEXT,

product\_category\_name TEXT,

product\_name\_length INT,

product\_description\_length INT,

product\_photos\_qty INT,

product\_weight\_g NUMERIC(10,2),

product\_length\_cm NUMERIC(10,2),

product\_height\_cm NUMERIC(10,2),

product\_width\_cm NUMERIC(10,2)

);

select \* from olist\_products\_dataset

------------------------------------------

CREATE TABLE olist\_sellers\_dataset (

seller\_id TEXT,

seller\_zip\_code\_prefix TEXT,

seller\_city TEXT,

seller\_state TEXT

);

select \* from olist\_sellers\_dataset

-----------------------------------------

CREATE TABLE olist\_product\_category\_name\_translation (

product\_category\_name TEXT,

product\_category\_name\_english TEXT

);

select \* from olist\_product\_category\_name\_translation

------------------------------------------

------------------------------------------

select \* from olist\_customers\_dataset

select \* from geolocation\_dataset;

SELECT \* FROM olist\_order\_items\_dataset

select \* from olist\_order\_payments\_dataset;

select \* from olist\_order\_reviews\_dataset

select \* from olist\_orders\_dataset

select \* from olist\_sellers\_dataset

select \* from olist\_products\_dataset

select \* from olist\_product\_category\_name\_translation

-------------------------------------------

--QUERY NO 1

---What was the total sales (total payment\_value) for each year?

SELECT

EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS year,

SUM(p.payment\_value) AS total\_sales

FROM olist\_orders\_dataset o

JOIN olist\_order\_payments\_dataset p

ON o.order\_id = p.order\_id

GROUP BY year

ORDER BY total\_sales;

---------------------------------------------

--QUERY NO 2

---For each product category, what were the total sales and number of orders?

select p. product\_category\_name, sum(o.price) as total\_sales, count(o.order\_item\_id) as total\_orders

from olist\_products\_dataset p

join olist\_order\_items\_dataset o

on p.product\_id = o. product\_id

group by product\_category\_name

order by total\_sales desc;

--------------------------------------------

--QUERY NO 3 \*

---Top 10 cities (customer\_city) that made the highest purchases.

select c.customer\_state, c.customer\_city, sum(p.payment\_value) as total\_purchase

from olist\_customers\_dataset c

join olist\_orders\_dataset o ---ha asach join kela, same column aahe mhanun

on o.customer\_id = c.customer\_id

join olist\_order\_payments\_dataset p ----mg ha join kela.

on o.order\_id = p.order\_id

group by c. customer\_city, c. customer\_state

order by total\_purchase desc limit 10;

---------------------------------------------

--QUERY NO 4

--Top 10 sellers (seller\_id) with the highest total revenue

select \* from olist\_orders\_dataset

select s.seller\_state, s.seller\_city, s.seller\_id, sum(p.price) as total\_sales

from olist\_sellers\_dataset s

join olist\_order\_items\_dataset p

on s.seller\_id = p.seller\_id

group by s.seller\_state, s.seller\_city, s.seller\_id

order by total\_sales desc limit 10

---------------------------------------------

SELECT

s.seller\_state,

s.seller\_city,

s.seller\_id,

SUM(p.price) AS total\_sales

FROM olist\_sellers\_dataset s

JOIN olist\_order\_items\_dataset p

ON s.seller\_id = p.seller\_id

GROUP BY s.seller\_state, s.seller\_city, s.seller\_id

ORDER BY total\_sales DESC

LIMIT 10;

----------------------------------------------

--QUERY NO 4

---Which payment\_type recorded the highest total payment? (Credit Card, Boleto, etc.)

select payment\_type, sum(payment\_value) as total\_payment

from olist\_order\_payments\_dataset

group by payment\_type

order by payment\_type

----------------------------------------------

delivery

----------------------------------------------

---Calculate the difference between the 'actual delivery date' (order\_delivered\_customer\_date) and

---the 'estimated delivery date' (order\_estimated\_delivery\_date) for each order.

---Find the percentage of orders delivered on time vs delayed orders.

---QUERY NO 1

SELECT

order\_id,

order\_delivered\_customer\_date,

order\_estimated\_delivery\_date,

CASE

WHEN order\_delivered\_customer\_date <= order\_estimated\_delivery\_date THEN 'On Time'

WHEN order\_delivered\_customer\_date > order\_estimated\_delivery\_date THEN 'Delayed'

ELSE 'Not Delivered'

END AS delivery\_status,

(order\_estimated\_delivery\_date - order\_delivered\_customer\_date) AS delivery\_diff\_days

FROM

olist\_orders\_dataset

order by delivery\_diff\_days asc;

----------------------------------

select \* from olist\_orders\_dataset

---------------------------------

---QUERY NO 2 = PERCENTAGE

WITH order\_status AS (

SELECT

order\_id,

CASE

WHEN order\_delivered\_customer\_date <= order\_estimated\_delivery\_date THEN 'On Time'

WHEN order\_delivered\_customer\_date > order\_estimated\_delivery\_date THEN 'Delayed'

ELSE 'Not Delivered'

END AS delivery\_status

FROM olist\_orders\_dataset

WHERE order\_delivered\_customer\_date IS NOT NULL

AND order\_estimated\_delivery\_date IS NOT NULL

)

SELECT

delivery\_status,

ROUND(100.0 \* COUNT(\*) / (SELECT COUNT(\*) FROM order\_status), 2) AS percentage

FROM order\_status

GROUP BY delivery\_status

ORDER BY delivery\_status;

-------------------------------

---QUERY NO 3

---What is the average shipping time (from purchase to delivered ?

SELECT

AVG(order\_delivered\_customer\_date - order\_purchase\_timestamp) AS avg\_interval

FROM olist\_orders\_dataset

WHERE order\_delivered\_customer\_date IS NOT NULL;

--------------------------------

---QUERY NO 4

---Which states have the longest delivery time?

select c.customer\_state, avg(o.order\_delivered\_customer\_date- order\_purchase\_timestamp) AS delivery\_diff\_days

from olist\_customers\_dataset c

join olist\_orders\_dataset o

on c.customer\_id = o.customer\_id

group by customer\_state

order by delivery\_diff\_days desc limit 10

---------------------------------

---Customer & Review Analysis

---------------------------------

---QUERY NO 1

--Average review\_score per product category.

SELECT

pc.product\_category\_name\_english AS product\_category,

ROUND(AVG(r.review\_score), 2) AS avg\_review\_score

FROM olist\_order\_items\_dataset i

JOIN olist\_products\_dataset p

ON i.product\_id = p.product\_id

JOIN olist\_product\_category\_name\_translation pc

ON p.product\_category\_name = pc.product\_category\_name

JOIN olist\_order\_reviews\_dataset r

ON i.order\_id = r.order\_id

GROUP BY pc.product\_category\_name\_english

ORDER BY avg\_review\_score DESC;

----------------------------------

---QUERY NO 2

---Find the Top 10 products with the lowest ratings.

SELECT

pc.product\_category\_name\_english AS product\_category,

ROUND(AVG(r.review\_score), 2) AS avg\_review\_score

FROM olist\_order\_items\_dataset i

JOIN olist\_products\_dataset p

ON i.product\_id = p.product\_id

JOIN olist\_product\_category\_name\_translation pc

ON p.product\_category\_name = pc.product\_category\_name

JOIN olist\_order\_reviews\_dataset r

ON i.order\_id = r.order\_id

GROUP BY pc.product\_category\_name\_english

ORDER BY avg\_review\_score asc limit 10;

------------------------------------

---QUERY NO 4

---What is the unique number of customers in each city?

SELECT

customer\_city,

COUNT(DISTINCT customer\_unique\_id) AS total\_count\_of\_unique\_customer

FROM olist\_customers\_dataset

GROUP BY customer\_city

ORDER BY total\_count\_of\_unique\_customer DESC;

--------------------------------------

---QUERY NO 5

---How many repeat customers (i.e., customers who made more than one purchase) are there?

SELECT

COUNT(\*) AS repeat\_customers\_count

FROM (

SELECT

customer\_unique\_id

FROM olist\_customers\_dataset c

JOIN olist\_orders\_dataset o

ON c.customer\_id = o.customer\_id

GROUP BY customer\_unique\_id

HAVING COUNT(o.order\_id) > 1

) AS repeat\_customers;

--------------------------------------

---QUERY NO 6

---Find the average price and freight value for each seller.

SELECT

s.seller\_id,

ROUND(AVG(i.price), 2) AS avg\_price,

ROUND(AVG(i.freight\_value), 2) AS avg\_freight\_value

FROM olist\_sellers\_dataset s

JOIN olist\_order\_items\_dataset i

ON s.seller\_id = i.seller\_id

GROUP BY s.seller\_id

ORDER BY avg\_price DESC;

---------------------------------------