--Pareto Chart data--

--combines order\_items, products and translation tables to enable needed calculations--

WITH

  prod\_revenue AS (

  SELECT

    string\_field\_1 AS product\_category\_name,

    ROUND(SUM(price + freight\_value),0) AS revenue,

  FROM

    `tc-da-1.olist\_db.olist\_order\_items\_dataset` items

  JOIN

    `tc-da-1.olist\_db.olist\_products\_dataset` products

  ON

    items.product\_id = products.product\_id

  JOIN

    `tc-da-1.olist\_db.product\_category\_name\_translation` translations

  ON

    products.product\_category\_name = translations.string\_field\_0

  GROUP BY

    product\_category\_name

  ORDER BY

    revenue DESC ),

--Calculates running total revenue and total revenue--

  total\_revenue AS (

  SELECT

    product\_category\_name,

    revenue,

    SUM(revenue) OVER(ORDER BY revenue DESC) AS running\_total,

    SUM(revenue) OVER() AS total\_rev

  FROM

    prod\_revenue )

--Calculates % of total revenue--

SELECT

  \*,

  revenue / total\_rev AS perc\_of\_revenue

FROM

  total\_revenue

ORDER BY

  revenue DESC;

--Calculates general statistics from sellers table--

SELECT

  COUNT(DISTINCT seller\_id) AS num\_sellers,

  COUNT(DISTINCT seller\_city) AS num\_seller\_cities,

  COUNT(DISTINCT seller\_state) AS num\_seller\_states

FROM

  `tc-da-1.olist\_db.olist\_sellers\_dataset`

--Joins customer,orders,reviews and items tables--

WITH t1 AS (

  SELECT customers.\*,

      orders.order\_id,

      orders.order\_status,

      orders.order\_purchase\_timestamp,

      orders.order\_delivered\_carrier\_date,

      orders.order\_estimated\_delivery\_date,

      reviews.review\_id,

      reviews.review\_score,

      reviews.review\_creation\_date,

      reviews.review\_answer\_timestamp,

      items.order\_id AS check\_order\_id,

      items.order\_item\_id,

      items.product\_id,

      items.seller\_id,

      items.shipping\_limit\_date,

      items.price,

      items.freight\_value

  FROM `tc-da-1.olist\_db.olist\_customesr\_dataset` customers

  JOIN `tc-da-1.olist\_db.olist\_orders\_dataset` orders

  ON customers.customer\_id = orders.customer\_id

  LEFT JOIN `tc-da-1.olist\_db.olist\_order\_reviews\_dataset` reviews

  ON orders.order\_id = reviews.order\_id

  LEFT JOIN `tc-da-1.olist\_db.olist\_order\_items\_dataset` items

  ON orders.order\_id = items.order\_id

)

--Calculates overall statistics--

SELECT

  COUNT(DISTINCT t1.customer\_unique\_id) AS total\_customers,

  COUNT(DISTINCT t1.customer\_city) AS num\_customer\_cities,

  COUNT(DISTINCT t1.customer\_state) AS num\_customer\_states,

  COUNT(DISTINCT t1.order\_id) AS num\_orders,

  AVG(t1.review\_score) AS avg\_review\_score,

  SUM(price+freight\_value) AS total\_revenue

FROM t1;

--Calculates number of reviews by review score--

SELECT review\_score,

      COUNT(review\_score) AS num\_reviews,

FROM `tc-da-1.olist\_db.olist\_order\_reviews\_dataset`

GROUP BY review\_score

ORDER BY review\_score

--Calculates revenue and number of orders on day of week basis--

SELECT

    EXTRACT(DAYOFWEEK FROM order\_purchase\_timestamp) AS day\_of\_week,

    COUNT(DISTINCT orders.order\_id) AS num\_orders,

    SUM(price+freight\_value) AS revenue

FROM `tc-da-1.olist\_db.olist\_orders\_dataset` orders

LEFT JOIN `tc-da-1.olist\_db.olist\_order\_items\_dataset` items

ON orders.order\_id = items.order\_id

GROUP BY day\_of\_week

ORDER BY 1;

--Calculates revenue and number of orders on day of month basis--

SELECT

    EXTRACT(DAY FROM order\_purchase\_timestamp) AS purchase\_day,

    COUNT(DISTINCT orders.order\_id) AS num\_orders,

    SUM(price+freight\_value) AS revenue

FROM `tc-da-1.olist\_db.olist\_orders\_dataset` orders

LEFT JOIN `tc-da-1.olist\_db.olist\_order\_items\_dataset` items

ON orders.order\_id = items.order\_id

GROUP BY purchase\_day

ORDER BY 1;

--Calculates revenue and number of orders on hour of day basis--

SELECT

    EXTRACT(HOUR FROM order\_purchase\_timestamp) AS purchase\_hour,

    COUNT(DISTINCT orders.order\_id) AS num\_orders,

    SUM(price+freight\_value) AS revenue

FROM `tc-da-1.olist\_db.olist\_orders\_dataset` orders

LEFT JOIN `tc-da-1.olist\_db.olist\_order\_items\_dataset` items

ON orders.order\_id = items.order\_id

GROUP BY purchase\_hour

ORDER BY 1;

--Calculates revenue and number of orders on year and month basis--

SELECT

    EXTRACT(YEAR FROM order\_purchase\_timestamp) AS purchase\_year,

    EXTRACT(MONTH FROM order\_purchase\_timestamp) AS purchase\_month,

    COUNT(DISTINCT orders.order\_id) AS num\_orders,

    SUM(price+freight\_value) AS revenue

FROM `tc-da-1.olist\_db.olist\_orders\_dataset` orders

LEFT JOIN `tc-da-1.olist\_db.olist\_order\_items\_dataset` items

ON orders.order\_id = items.order\_id

GROUP BY purchase\_year, purchase\_month

ORDER BY 1, 2;

--Calculates days to deliver and difference between estimated and actual delivery times for top product catecories--

WITH t1 AS(

SELECT translations.string\_field\_1 AS product\_category,

    orders.order\_id,

    orders.customer\_id,

    orders.order\_purchase\_timestamp,

    orders.order\_delivered\_carrier\_date,

    orders.order\_delivered\_customer\_date,

    orders.order\_estimated\_delivery\_date,

    items.seller\_id,

    items.price,

    items.freight\_value,

    reviews.review\_score

FROM `tc-da-1.olist\_db.olist\_orders\_dataset` orders

LEFT JOIN `tc-da-1.olist\_db.olist\_order\_items\_dataset` items

ON orders.order\_id = items.order\_id

LEFT JOIN `tc-da-1.olist\_db.olist\_products\_dataset` products

ON items.product\_id = products.product\_id

LEFT JOIN `tc-da-1.olist\_db.olist\_sellers\_dataset` sellers

ON items.seller\_id = sellers.seller\_id

LEFT JOIN `tc-da-1.olist\_db.olist\_order\_reviews\_dataset` reviews

ON orders.order\_id = reviews.order\_id

LEFT JOIN `tc-da-1.olist\_db.product\_category\_name\_translation` translations

ON products.product\_category\_name = translations.string\_field\_0

)

SELECT

review\_score,

AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) AS days\_to\_deliver,

AVG(DATE\_DIFF(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, DAY)) AS days\_estimated\_vs\_actual

FROM t1

WHERE review\_score IS NOT NULL

AND product\_category IN ('health\_beauty', 'watches\_gifts', 'bed\_bath\_table', 'sports\_leisure', 'computers\_accessories', 'furniture\_decor', 'housewares', 'cool\_stuff', 'auto', 'garden\_tools', 'toys', 'baby', 'perfumery', 'telephony', 'office\_furniture', 'stationary')

GROUP BY review\_score

ORDER BY review\_score;

--delivery times by review score--

SELECT

 reviews.review\_score,

 AVG(TIMESTAMP\_DIFF(order\_approved\_at, order\_purchase\_timestamp, HOUR)) AS hours\_to\_approve,

 AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) AS days\_to\_deliver,

 AVG(DATE\_DIFF(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, DAY)) AS days\_estimated\_vs\_actual

FROM `tc-da-1.olist\_db.olist\_orders\_dataset` orders

LEFT JOIN `tc-da-1.olist\_db.olist\_order\_reviews\_dataset` reviews

ON orders.order\_id = reviews.order\_id

WHERE

 order\_status = 'delivered'

group by 1

order by 1 DESC;

--delivery times for top 15 product categories--

SELECT

    translations.string\_field\_1 AS product\_category,

    AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) AS days\_to\_deliver,

    AVG(DATE\_DIFF(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, DAY)) AS days\_estimated\_vs\_actual

FROM `tc-da-1.olist\_db.olist\_orders\_dataset` orders

LEFT JOIN `tc-da-1.olist\_db.olist\_order\_items\_dataset` items

ON orders.order\_id = items.order\_id

LEFT JOIN `tc-da-1.olist\_db.olist\_products\_dataset` products

ON items.product\_id = products.product\_id

LEFT JOIN `tc-da-1.olist\_db.product\_category\_name\_translation` translations

ON products.product\_category\_name = translations.string\_field\_0

WHERE order\_status = 'delivered'

    AND translations.string\_field\_1 IS NOT NULL

    AND translations.string\_field\_1 IN ('health\_beauty', 'watches\_gifts', 'bed\_bath\_table', 'sports\_leisure', 'computers\_accessories', 'furniture\_decor', 'housewares', 'cool\_stuff', 'auto',              'garden\_tools', 'toys', 'baby', 'perfumery', 'telephony', 'office\_furniture', 'stationary')

GROUP BY product\_category;

--Customer city and state data--

SELECT DISTINCT(customer\_unique\_id),

      customer\_city,

      customer\_state

FROM `tc-da-1.olist\_db.olist\_customesr\_dataset`

GROUP BY customer\_unique\_id, customer\_city, customer\_state;

--Seller city and state data--

SELECT seller\_id,

    seller\_city,

    seller\_state

FROM `tc-da-1.olist\_db.olist\_sellers\_dataset`

GROUP BY seller\_id, seller\_city, seller\_state;

--Calculates average delivery times by state--

SELECT

    customers.customer\_state,

    AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) AS days\_to\_deliver,

FROM `tc-da-1.olist\_db.olist\_orders\_dataset` orders

JOIN`tc-da-1.olist\_db.olist\_customesr\_dataset` customers

ON orders.customer\_id = customers.customer\_id

GROUP BY customer\_state

ORDER BY days\_to\_deliver DESC;

--Calculates average delivery times, avg price and avg freight\_value by state--

SELECT

    CONCAT(customers.customer\_state, ',', ' ', 'Brazil') AS customer\_state,

    ROUND(AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)),0) AS days\_to\_deliver,

    AVG(price) as avg\_price,

    AVG(freight\_value) AS avg\_freight\_value,

FROM `tc-da-1.olist\_db.olist\_orders\_dataset` orders

JOIN`tc-da-1.olist\_db.olist\_customesr\_dataset` customers

ON orders.customer\_id = customers.customer\_id

JOIN `tc-da-1.olist\_db.olist\_order\_items\_dataset` items

ON orders.order\_id = items.order\_id

GROUP BY customer\_state

ORDER BY days\_to\_deliver DESC;