ONLINE RETAIL CASE STUDY

Key Business Questions

- What is the revenue generated by each customer, and how are the customers ranked based on their revenue, using a 5-level ranking system?

```
--SEGMENTING THE CUSTOMERS BASED ON THEIR TOTAL REVENUE
SELECT CUSTOMER_ID, SUM(QUANTITY*PRICE) REVENUE,
                    NTILE(5) OVER (ORDER BY SUM(QUANTITY*PRICE) DESC) CUST_RANK
FROM TABLERETAIL
GROUP BY CUSTOMER_ID;
```

- What are the top products in terms of revenue, and what is the percentage rank of each product's revenue compared to the total revenue, considering all products?

```
--CALCULATING THE TOP 10% OF REVENUE-GENERATING PRODUCTS
WITH TOP_REVENUE_PRODUCT AS (SELECT STOCKCODE,
                                    ROUND(PERCENT_RANK() OVER (ORDER BY REVENUE
                                                               DESC)*100, 2) RANK_PERC
                             FROM (
                                   SELECT STOCKCODE, SUM(QUANTITY*PRICE) REVENUE
                                   FROM TABLERETAIL
                                   GROUP BY STOCKCODE
                                   ORDER BY REVENUE DESC
                                   )
                            )

SELECT * FROM TOP_REVENUE_PRODUCT
WHERE RANK_PERC <= 10;
```

- What is the average number of transactions needed for customers to reach a cumulative purchase of £50, based on the quantity and price of items purchased on each transaction?
  The Pareto principle, also known as the 80/20 rule, suggests that a minority of causes can account for most consequences. Applied to sales, it implies that a small fraction of customers, typically 20-30%, are responsible for most sales. Is this principle applicable to sales?

```
--CALCULATING CUMULATIVE REVENUE FOR THE CUSTOMERS ALONG WITH THE CUMULATIVE DISTRIBUTION
OF CUSTOMERS (THE PARETO PRINCIPLE)

SELECT CUSTOMER_ID, SUM(QUANTITY*PRICE) REVENUE,
       ROUND(100*SUM(SUM(QUANTITY*PRICE)) OVER (ORDER BY SUM(QUANTITY*PRICE)
             DESC)/SUM(SUM(QUANTITY*PRICE)) OVER (), 2) CUM_REV_PERCENTAGE,

       ROUND(100*CUME_DIST() OVER (ORDER BY SUM(QUANTITY*PRICE) DESC), 2)
                                                                CUM_CUST_PERCENTAGE
FROM TABLERETAIL
GROUP BY CUSTOMER_ID
ORDER BY REVENUE DESC;
```

- What is the number of days since the last purchase made by each customer, based on their most recent invoice date?

```
--CALCULATES THE NUMBER OF DAYS SINCE EACH CUSTOMER'S LAST PURCHASE
SELECT DISTINCT CUSTOMER_ID, TRUNC(MAX(INVOICE_DATE) OVER () - LAST_PURCHASE)
                                                              DAYS_SINCE_LAST_PURCHASE
FROM (
      SELECT CUSTOMER_ID, INVOICE_DATE,
             LAST_VALUE(INVOICE_DATE) OVER (PARTITION BY CUSTOMER_ID ORDER BY
                        INVOICE_DATE RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
                        FOLLOWING) LAST_PURCHASE
      FROM TABLERETAIL
      )
ORDER BY DAYS_SINCE_LAST_PURCHASE DESC;
```

- What is the monthly revenue and percentage change in revenue compared to the previous month, for each month in the data set?

```
--CALCULATING THE MONTH-OVER-MONTH PERCENTAGE CHANGE IN REVENUE
WITH MONTH_REVENUE_DIFF AS (SELECT EXTRACT(MONTH FROM INVOICE_DATE) "MONTH",
                                   SUM(QUANTITY*PRICE) MONTH_REV,
                                   LAG(SUM(QUANTITY*PRICE)) OVER (ORDER BY EXTRACT(MONTH
                                       FROM INVOICE_DATE)) PREV_MON_REV
                            FROM TABLERETAIL
                            GROUP BY EXTRACT(MONTH FROM INVOICE_DATE)
                            )


SELECT MONTH, MONTH_REV, PREV_MON_REV,
              ROUND ((MONTH_REV - PREV_MON_REV) / PREV_MON_REV * 100, 2) CHANGE
FROM MONTH_REVENUE_DIFF;
```

- What is the average revenue per day of the week, based on the quantity and price of the items sold on each day of the week?

```
--CALCULATE THE AVERAGE REVENUE PER DAY OF THE WEEK
SELECT TO_CHAR(INVOICE_DATE, 'DAY') DAY_OF_WEEK, ROUND(AVG(QUANTITY * PRICE), 2)
                                                                  AVG_REV_PER_DAY
FROM TABLERETAIL
GROUP BY TO_CHAR(INVOICE_DATE, 'DAY');
```

- On average, how many days/transactions does it take a customer to reach a spending threshold of £50?

```
--CUSTOMERS IDs WITH CUMULATIVE TRANSACTIONS OVER £50
WITH CUST_OVER_50 AS (SELECT CUSTOMER_ID
                      FROM TABLERETAIL
                      GROUP BY CUSTOMER_ID
                      HAVING SUM(QUANTITY*PRICE) >= 50
                      ),


--CALCULATING NUMBER OF TRANSACTIONS NEEDED TO REACH £50 FOR EACH CUSTOMER
CUST_TRANSACTIONS_COUNT AS (SELECT CUSTOMER_ID,
                                   COUNT(DISTINCT INVOICE_DATE)+1 TRANSACTIONS_COUNT
                            FROM (
                                  SELECT TR.CUSTOMER_ID, TR.INVOICE_DATE,
                                         SUM(TR.QUANTITY*TR.PRICE) OVER (PARTITION BY
                                             TR.CUSTOMER_ID ORDER BY TR.INVOICE_DATE)
                                                                                CUM_SUM
                                  FROM TABLERETAIL TR
                                  JOIN CUST_OVER_50 C50
                                  ON TR.CUSTOMER_ID = C50.CUSTOMER_ID
                                  )
                            WHERE CUM_SUM <= 250
                            GROUP BY CUSTOMER_ID
                            ORDER BY CUSTOMER_ID
                            )


--WE CAN CONCLUDE THAT IT TAKES A CUSTOMER 2 TRANSACTIONS ON AVERAGE TO REACH £50
SELECT ROUND(AVG(TRANSACTIONS_COUNT), 2) AVG_TRANSACTIONS_50
FROM CUST_TRANSACTIONS_COUNT;
```