**COFEE SALES- SQL QUERIES**

**KPI REQUIREMENTS**

1. Total Sales Analysis
2. Total Order Analysis
3. Total Quantity Sold Analysis
4. **Total Sale Analysis**
5. Calculate total sales for each respective month.
6. Determine the month-on-month increase or decrease in sales.
7. Calculate the difference in sales between the selected month and the previous month.
8. **Total Order Analysis**
9. Calculate the total number of orders for each respective month.
10. Determine the month-on-month increase or decrease in the number of orders.
11. Calculate the difference in the number of orders between the selected month and previous month.
12. **Total Quantity Sold Analysis:**
13. Calculate the total quantity sold for each respective month.
14. Determine the month-on-month increase or decrease in the total quantity sold.
15. Calculate the difference in the total quantity sold between the selected month and the previous month.

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

* **Total Sales Analysis**

|  |  |
| --- | --- |
| **-1-**  **-KPI- Total Sales /Month** | Calculate total sales for each respective month. |
| SQL QUERY | SELECT ROUND(SUM(transaction\_qty \* unit\_price)) AS Total\_Sales\_March  FROM coffee\_shop\_sales  WHERE MONTHNAME(transaction\_date) = "March"; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-1-**  **-KPI-**  **Sales**  **Month\_on\_Month Increase** | Determine the month-on-month increase or decrease in sales. |
| SQL QUERY | SELECT  MONTH(transaction\_date) AS month,  ROUND(SUM(unit\_price \* transaction\_qty)) AS total\_sales,  (SUM(unit\_price \* transaction\_qty) - LAG(SUM(unit\_price \* transaction\_qty), 1)  OVER (ORDER BY MONTH(transaction\_date))) / LAG(SUM(unit\_price \* transaction\_qty), 1)  OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS mom\_increase\_percentage  FROM  coffee\_shop\_sales  WHERE  MONTH(transaction\_date) IN (4, 5) -- for months of April and May  GROUP BY  MONTH(transaction\_date)  ORDER BY  MONTH(transaction\_date); |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-2-**  **-KPI- Total Order/ Month** | Calculate the total number of orders for each respective month. |
| SQL QUERY | SELECT COUNT(\*)  AS Total\_Orders\_March  FROM coffee\_shop\_sales  WHERE MONTHNAME(transaction\_date) = "March"; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-2-**  **-KPI- Order**  **Month\_On\_Month** | Determine the month-on-month increase or decrease in the number of orders. |
| SQL QUERY | SELECT  MONTH(transaction\_date) AS month,  ROUND(COUNT(transaction\_id)) AS total\_orders,  (COUNT(transaction\_id) - LAG(COUNT(transaction\_id), 1)  OVER (ORDER BY MONTH(transaction\_date))) / LAG(COUNT(transaction\_id), 1)  OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS mom\_increase\_percentage  FROM  coffee\_shop\_sales  WHERE  MONTH(transaction\_date) IN (4, 5) -- for April and May  GROUP BY  MONTH(transaction\_date)  ORDER BY  MONTH(transaction\_date); |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-3-**  **-KPI- Total Quantity/**  **Month** | Calculate the total quantity sold for each respective month. |
| SQL QUERY | SELECT SUM(transaction\_qty)  AS Total\_Quantity\_March  FROM coffee\_shop\_sales  WHERE MONTHNAME(transaction\_date) = "March"; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-2-**  **-KPI- Quantity**  **Month\_On\_Month** | Determine the month-on-month increase or decrease in the total quantity sold. |
| SQL QUERY | SELECT  MONTH(transaction\_date) AS month,  ROUND(SUM(transaction\_qty)) AS total\_quantity\_sold,  (SUM(transaction\_qty) - LAG(SUM(transaction\_qty), 1)  OVER (ORDER BY MONTH(transaction\_date))) / LAG(SUM(transaction\_qty), 1)  OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS mom\_increase\_percentage  FROM  coffee\_shop\_sales  WHERE  MONTH(transaction\_date) IN (4, 5) -- for April and May  GROUP BY  MONTH(transaction\_date)  ORDER BY  MONTH(transaction\_date); |
| OUTPUT |  |

**COFEE SALES- SQL QUERIES**

**CHART REQUIREMENTS**

1. Calendar Heat Map
2. Sales Analysis by Weekdays and Weekends
3. Sales Analysis by Store location.
4. **Calendar Heat Map:**
5. Implement a calendar heat map that dynamically adjust based on the selected month from a slicer.
6. Each day on the calendar will be color-coded to represent sales volume, with darker shades indicating higher sales.
7. Implement tooltips to display detailed metrics (Sales, Orders, Quantity) when hovering over a specific day.
8. **Sales Analysis by Weekdays and Weekends:**
9. Segment sales data into weekdays and weekends to analyze performance variations.
10. Provide insights into whether sales patterns differ significantly between weekdays and weekends.
11. **Sales Analysis by Store location:**
12. Visualize sales data by different store location.
13. Include month-over-month (MoM) difference metrics based on the selected month in the slicer.
14. Highlight MoM sales increase or decrease for each store location to identify trends.
15. **Daily Sales Analysis with Average Line:**
16. Display the daily sales for the selected month with the line chart.
17. Incorporate an average on the chart to represent the average daily sales.
18. Highlighting bars exceeding or falling below the average sales to identify exceptional sales days.
19. **Sales Analysis by Product Categories**
20. Analyze sales performance across different products categories.
21. Provide insights into which product categories contribute most to overall sales.
22. **Top 10 Product by Sales**
23. Identify and display the top 10 product based on sales volume
24. Allow users to quickly visualize the best performing products in terms of sales.
25. **Sales Analysis by days and hours**
26. Utilize a heat map to visualize sales patterns by days and hours.
27. Implement tooltips to display detailed metrics (Sales, Orders, Quantity), When hovering over a specific day-hour.

|  |  |
| --- | --- |
| **-CHART-**  **-1-C-**  **Tooltip**  **Sales, Orders, Quantity**  **/Day** | C. Implement tooltips to display detailed metrics (Sales, Orders, Quantity) when hovering over a specific day. |
| SQL QUERY | SELECT  CONCAT(ROUND(SUM(transaction\_qty \* unit\_price)/1000,1),'K') AS Total\_sales,  CONCAT(ROUND(SUM(transaction\_qty)/1000,1),'k') AS Total\_orders,  CONCAT(ROUND(COUNT(transaction\_id)/1000,1),'k') AS Total\_quantity  FROM coffee\_shop\_sales  WHERE transaction\_date = '2023-05-18'; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-CHART-**  **-2-A-**  **Doughnut Chart**  **Weekdays-Weekends** | Segment sales data into weekdays and weekends to analyze performance variations. |
| SQL QUERY | SELECT  CASE WHEN DAYOFWEEK(transaction\_date) IN (1,7) THEN 'Weekend'  ELSE 'Weekdays'  END AS Day\_type,  CONCAT(ROUND(SUM(transaction\_qty \* unit\_price)/1000,1),'K') AS Total\_sales  FROM coffee\_shop\_sales  WHERE MONTH(transaction\_date) = 5 -- May Month  GROUP BY Day\_type; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-CHART-**  **Total Sales by Store location/ Month** | Visualize sales data by different store location. |
| SQL QUERY | SELECT  store\_location,  CONCAT(ROUND(SUM(transaction\_qty \* unit\_price)/1000,1),'k')AS Total\_sales  FROM coffee\_shop\_sales  WHERE MONTH(transaction\_date) = 5  GROUP BY store\_location  ORDER BY Total\_sales DESC; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-CHART-**  **Daily Sales Analysis with Average Line** | Incorporate an average on the chart to represent the **average daily sales** |
| SQL QUERY | SELECT  CONCAT(ROUND(AVG(total\_sales)/1000,1),'K')AS avg\_sales  FROM  (  SELECT SUM(unit\_price \* transaction\_qty) AS total\_sales  FROM coffee\_shop\_sales  WHERE MONTH(transaction\_date) = 5  GROUP BY transaction\_date  ) AS Internal\_query; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-CHART-**  **Daily Sales Analysis** | Display the daily sales for the selected month with the line chart. |
| SQL QUERY | SELECT  DAY(transaction\_date) AS day\_of\_month,  SUM(unit\_price \* transaction\_qty) AS Daily\_sales  FROM coffee\_shop\_sales  WHERE MONTH(transaction\_date) =5  GROUP BY day\_of\_month; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-CHART-**  **Daily Sales Analysis** | Display the daily sales for the selected month with the line chart. |
| SQL QUERY | SELECT  DAY(transaction\_date) AS day\_of\_month,  SUM(unit\_price \* transaction\_qty) AS Daily\_sales  FROM coffee\_shop\_sales  WHERE MONTH(transaction\_date) =5  GROUP BY day\_of\_month;  SELECT  day\_of\_month,  CASE  WHEN total\_sales > avg\_sales THEN 'Above Average'  WHEN total\_sales < avg\_sales THEN 'Below Average'  ELSE 'Equal to Average'  END AS sales\_status,  total\_sales  FROM (  SELECT  DAY(transaction\_date) AS day\_of\_month,  SUM(unit\_price \* transaction\_qty) AS total\_sales,  AVG(SUM(unit\_price \* transaction\_qty)) OVER () AS avg\_sales  FROM  coffee\_shop\_sales  WHERE  MONTH(transaction\_date) = 5 -- Filter for May  GROUP BY  DAY(transaction\_date)  ) AS sales\_data  ORDER BY  day\_of\_month; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-CHART-**  **Sales Analysis by Product Categories/ Month** | Analyze sales performance across different products categories. |
| SQL QUERY | SELECT  product\_category,  CONCAT(ROUND(SUM(unit\_price \* transaction\_qty)/1000,1),'k') AS Total\_sales  FROM coffee\_shop\_sales  WHERE MONTH(transaction\_date) = 5  GROUP BY product\_category; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-CHART-**  **Top 10 Product by Sales** | Identify and display the top 10 product based on sales volume |
| SQL QUERY | SELECT  product\_type,  SUM(unit\_price \* transaction\_qty)AS Total\_sales  FROM coffee\_shop\_sales  WHERE MONTH(transaction\_date) = 5  GROUP BY product\_type  ORDER BY Total\_sales DESC  LIMIT 10; |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-CHART-**  **Sales Analysis by hours** | Implement tooltips to display detailed metrics (Sales, Orders, Quantity), When hovering over a specific hour. |
| SQL QUERY | SELECT  SUM(unit\_price \* transaction\_qty) AS Total\_sales,  SUM(transaction\_qty) AS Total\_quantity,  COUNT(\*) AS Total\_orders  FROM coffee\_shop\_sales  WHERE MONTH(transaction\_date) = 5  AND DAYOFWEEK(transaction\_date) = 2 -- Monday  AND HOUR(transaction\_time) = 8; -- Hour No. 8 |
| OUTPUT |  |

|  |  |
| --- | --- |
| **-CHART-**  **Sales Analysis by days** | Implement tooltips to display detailed metrics (Sales, Orders, Quantity), When hovering over a specific day |
| SQL QUERY | SELECT  CASE  WHEN DAYOFWEEK(transaction\_date) = 2 THEN 'Monday'  WHEN DAYOFWEEK(transaction\_date) = 3 THEN 'Tuesday'  WHEN DAYOFWEEK(transaction\_date) = 4 THEN 'Wednesday'  WHEN DAYOFWEEK(transaction\_date) = 5 THEN 'Thursday'  WHEN DAYOFWEEK(transaction\_date) = 6 THEN 'Friday'  WHEN DAYOFWEEK(transaction\_date) = 7 THEN 'Saturday'  ELSE 'Sunday'  END AS Day\_of\_Week,  ROUND(SUM(unit\_price \* transaction\_qty)) AS Total\_Sales  FROM  coffee\_shop\_sales  WHERE  MONTH(transaction\_date) = 5 -- Filter for May (month number 5)  GROUP BY  CASE  WHEN DAYOFWEEK(transaction\_date) = 2 THEN 'Monday'  WHEN DAYOFWEEK(transaction\_date) = 3 THEN 'Tuesday'  WHEN DAYOFWEEK(transaction\_date) = 4 THEN 'Wednesday'  WHEN DAYOFWEEK(transaction\_date) = 5 THEN 'Thursday'  WHEN DAYOFWEEK(transaction\_date) = 6 THEN 'Friday'  WHEN DAYOFWEEK(transaction\_date) = 7 THEN 'Saturday'  ELSE 'Sunday'  END; |
| OUTPUT |  |