

naikdarshan221@gmail.com

MALMARI QTR-01 SALES DATA ANALYSIS USING MYSQL

Walmart > <

Key Metrics

```
SELECT
          Total_purchases,
          Total_Quantity,
          avg_ordervalue,
 8
 9
          Revenue,
          profit,
10
          CASE
11
12
             WHEN revenue = 0 THEN 0
13
             ELSE ROUND(((profit / revenue) * 100), 2)
          END AS Profit_percentage
14
15
      FROM
16
          (SELECT
             COUNT(invoice_id) AS Total_Purchases,
17
                 SUM(quantity) AS Total_Quantity,
18
                 ROUND(AVG(total), 2) AS avg_ordervalue,
19
                 ROUND(SUM(total), 2) AS Revenue,
20
                 ROUND(SUM(total) - SUM(Cogs), 2) AS Profit
21
22
          FROM
             sales) AS sub_overall;
23
```

Result Grid			E	xport: 📳 🛮 V	Vrap Cell Conf	tent: IA
1	Total_purchases	Total_Quantity	avg_ordervalue	Revenue	profit	Profit_percentage
▶ 9	95	5472	322.50	320886.39	15280.30	4.76



Walmart : Top Customer Type

```
Limit to 10000 rows
SELECT
    customer_type,
    ROUND(revenue, 2) AS revenue,
    ROUND(((revenue / (SELECT
                    SUM(total)
                FROM
                    sales)) * 100),
            2) AS percentage_share
FROM
    (SELECT
        customer_type, SUM(total) AS Revenue
    FROM
        sales
    GROUP BY customer_type
    ORDER BY revenue DESC) AS ct
GROUP BY customer_type
LIMIT 1;
```

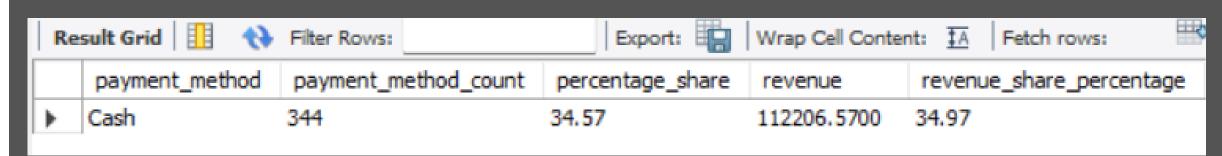
Re	sult Grid	Filter Rows:	
	customer_type	revenue	percentage_share
•	Member	163625.10	50.99



Most Preferred Payment

```
Limit to 10000 rows 🔻 🤘
SELECT
    payment_method,
    payment method count,
    ROUND(((payment_method_count / (SELECT
                    COUNT(invoice id)
                FROM
                    sales)) * 100),
            AS percentage_share,
    revenue,
    ROUND(((revenue / (SELECT
                    SUM(total)
                FROM
                    sales)) * 100),
            2) AS revenue share percentage
FROM
    (SELECT
        COUNT(invoice_id) AS payment_method_count,
            payment AS payment method,
            SUM(total) AS revenue
    FROM
        sales
    GROUP BY payment method) AS payment
ORDER BY payment method count DESC
LIMIT 1;
```

<u>Type</u>





Top Selling Product Line

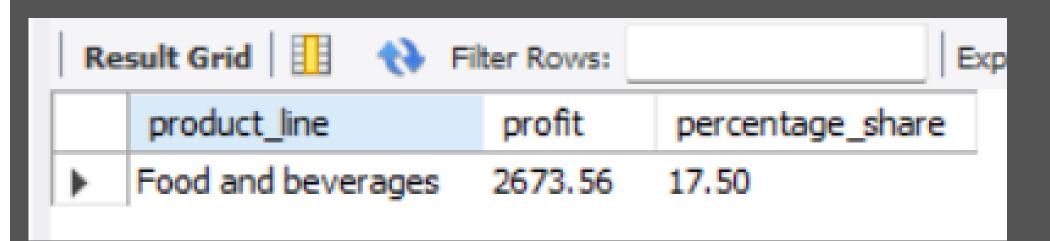
```
Limit to 10000 rows
SELECT
    product_line,
    Total_revenue,
    ROUND((Total_revenue / (SELECT
                     SUM(total)
                FROM
                     sales) * 100),
            2) AS percentage_share
FROM
    (SELECT
        product_line, SUM(total) A5 Total_Revenue
    FROM
        sales
    GROUP BY product_line
    ORDER BY Total_Revenue DESC
    LIMIT 1) AS pl;
```

Re	sult Grid 🔢 🙌 F	Filter Rows:	Export:	
	product_line	Total_revenue	percentage_share	
>	Food and beverages	56144.8440	17.50	



Most Profitable Product

```
Limit to 10000 rows ▼ | 🏂 | 🥩
        SELECT
 89 •
            product_line,
            ROUND(profit, 2) AS profit,
            ROUND((SUM(profit) / (SELECT
 92
                            SUM(total) - SUM(cogs)
 93
 94
                        FROM
                            sales) * 100),
 95
                    2) AS percentage_share
 96
        FROM
 97
            (SELECT
 98
                product_line, total_revenue - total_cogs AS profit
 99
            FROM
100
101
                (SELECT
                product_line,
102
                    SUM(total) AS total Revenue,
103
                    SUM(cogs) AS total_cogs
104
105
            FROM
                sales
106
            GROUP BY product_line) AS pl
107
            GROUP BY product_line
108
            ORDER BY profit DESC) AS p2
109
        GROUP BY product_line
110
111
        LIMIT 1;
```





Product Line With Quantity Sold

```
SELECT
    product_line, SUM(quantity) AS total_quantity
FROM
    sales
GROUP BY product_line
ORDER BY total_quantity DESC;
```

	product_line	total_quantity
•	Electronic accessories	961
	Food and beverages	952
	Home and lifestyle	911
	Sports and travel	902
	Fashion accessories	902
	Health and beauty	844





Month Wise Revenue Distribution

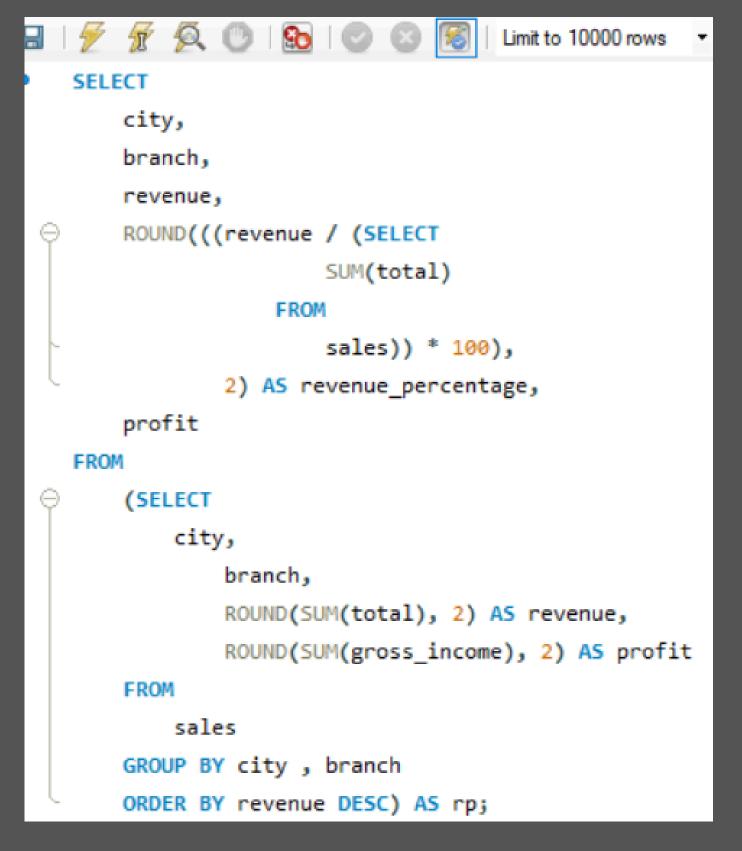
```
SELECT
    month,
    month_name,
    ROUND(Total_revenue, 2) AS Total_revenue,
    ROUND(((Total_revenue / (SELECT
                    SUM(total)
                FROM
                    sales)) * 100),
            2) AS Percentage_share
FROM
    (SELECT
        MONTH(date) AS month,
            MONTHNAME(date) AS month_name,
            SUM(total) AS Total_revenue
    FROM
        sales
    GROUP BY month , month_name
    ORDER BY month) A5 revenue;
```

	month	month_name	Total_revenue	Percentage_share
•	1	January	116291.87	36.24
	2	February	95727.38	29.83
	3	March	108867.15	33.93



Branch Wise Revenue and

Profit Distribution



Re	sult Grid 🛚 🔢	₹ Filt	er Rows:	Export	Wrap
	city	branch	revenue	revenue_percentage	profit
•	Naypyitaw	С	110490.78	34.43	5261.47
	Yangon	A	105861.01	32.99	5041.00
	Mandalay	В	104534.61	32.58	4977.84





<u>Distribution of Payment</u> <u>Method In Total Revenue</u>

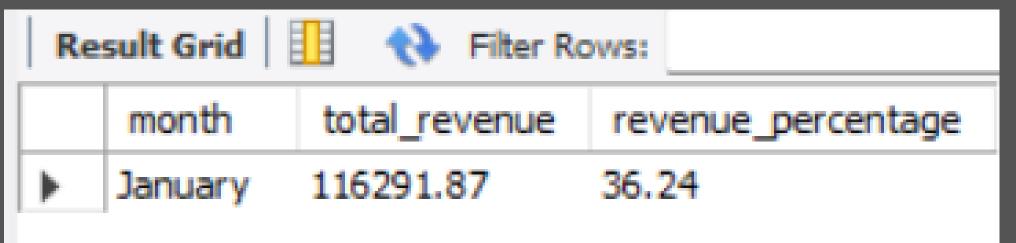
```
SELECT
    payment,
    count_of_transactions,
    ROUND(((total_revenue / (SELECT
                    SUM(total)
                FROM
                    sales)) * 100),
            2) AS revenue_percentage
FROM
    (SELECT
        payment,
            COUNT(invoice_id) AS count_of_Transactions,
            SUM(total) AS total_revenue
    FROM
        sales
    GROUP BY payment
    ORDER BY total_revenue DESC) AS pt;
```

	payment	count_of_transactions	revenue_percentage
•	Cash	344	34.97
	Ewallet	342	33.76
	Credit card	309	31.27



Peak Sales Month

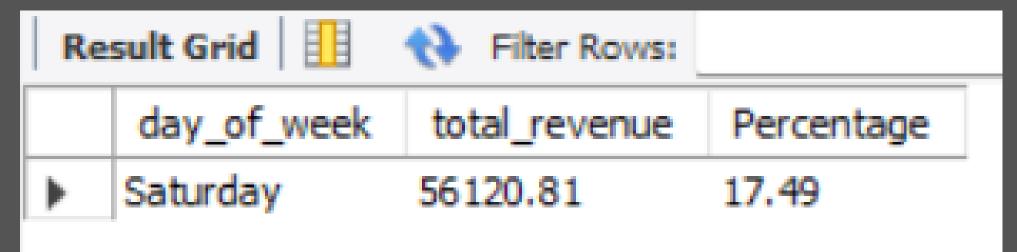
```
🙊 🕛 | 🔂 | 🔘 🔞
                                Limit to 10000 rows
SELECT
    month,
    ROUND(total_revenue, 2) AS total_revenue,
    ROUND(((total_revenue / (SELECT
                   SUM(total)
               FROM
                    sales)) * 100),
            2) AS revenue_percentage
FROM
    (SELECT
       MONTHNAME(date) AS month,
            ROUND(SUM(total), 2) AS total_revenue
    FROM
       sales
    GROUP BY month
    ORDER BY total revenue DESC
    LIMIT 1) pst;
```





Peak Sales Day

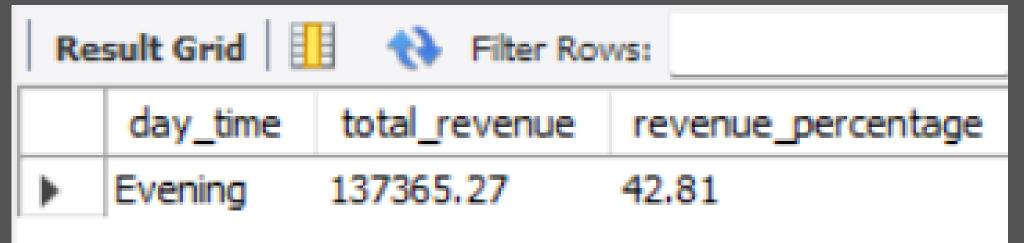
```
Limit to 10000 rows
133 •
        SELECT
            day_of_week,
134
            ROUND(total_revenue, 2) AS total_revenue,
135
            ROUND(((total_revenue / (SELECT
136
                             SUM(total)
137
138
                         FROM
                             sales)) * 100),
139
140
                     2) AS Percentage
141
        FROM
             (SELECT
142
                 day of week, SUM(total) AS total revenue
143
144
            FROM
                 sales
145
            GROUP BY day_of_week
146
            ORDER BY total_revenue DESC
147
             LIMIT 1) AS ps;
148
```





Peak Sales Day Time

```
Limit to 10000 rows ▼ | 🏡
L51 •
        SELECT
            day_time,
L52
            ROUND(total_revenue, 2) AS total_revenue,
L53
            ROUND(((total_revenue / (SELECT
L54
                             SUM(total)
L55
L56
                         FROM
                             sales)) * 100),
L57
158
                     2) AS revenue_percentage
L59
        FROM
L60
            (SELECT
                 day_time, ROUND(SUM(total), 2) AS total_revenue
L61
L62
            FROM
                 sales
L63
            GROUP BY day time
L64
            ORDER BY total_revenue DESC
L65
            LIMIT 1) pst;
L66
```





Peak Sales Time Shift on a

Daily Basis

```
select day_of_week, day_time, total_sales, revenue_percentage, ranking

from(
    select day_of_week, day_time, total_sales, round(((total_sales/(select sum(total) from sales))*100),2)
    as revenue_percentage, rank() over( partition by day_of_week order by total_sales desc) as ranking
    from(
        select day_of_week, day_time, sum(total) as total_sales
        from sales
        group by day_of_week, day_time
        order by day_of_week, day_time desc) as total_sales
) as total_sales_2
where ranking=1
order by total_sales desc;
```

	day_of_week	day_time	total_sales	revenue_percentage	ranking
•	Saturday	Evening	27924.5295	8.70	1
	Tuesday	Evening	21997.9200	6.86	1
	Thursday	Evening	19115.6805	5.96	1
	Sunday	Evening	18408.0435	5.74	1
	Wednesday	Afternoon	18041.1735	5.62	1
	Friday	Afternoon	17845.8840	5.56	1
	Monday	Evening	15657.6000	4.88	1



Top 3 Peak Sales Day Branch wise

```
select branch, day_of_week,revenue, percentage, ranking
from(
    select branch, day_of_week, revenue, percentage, rank() over(partition by branch order by percentage desc) as ranking
    from
    (
        select branch, day_of_week, sum(total) as revenue, round(((sum(quantity)/(select sum(quantity) from sales))*100),2) as percentage
        from sales
        group by branch, day_of_week
        )as br
)as br1
    where ranking in (1,2,3);
```

	branch	day_of_week	revenue	percentage	ranking
•	Α	Saturday	16765.4130	5.54	1
	Α	Sunday	17006.8185	5.19	2
	A	Friday	14543.8650	5.01	3
	В	Saturday	21284.4240	6.12	1
	В	Tuesday	18859.2390	5.68	2
	В	Wednesday	12119.2365	5.12	3
	С	Sunday	17035.7460	5.65	1
	С	Tuesday	17667.7935	5.59	2
	С	Saturday	18070.9725	5.14	3

Walmart >

```
SELECT
    CASE
        WHEN
            branch_avg > (SELECT
                    AVG(quantity)
                FROM
                    sales)
        THEN
            branch
        ELSE NULL
    END AS branch,
    branch_avg,
    (SELECT
            AVG(quantity)
        FROM
            sales) AS avg quantity
FROM
    (SELECT
        branch, AVG(quantity) branch_avg
    FROM
        sales
    GROUP BY branch) AS p1
WHERE
   branch_avg > (SELECT
            AVG(quantity)
        FROM
            sales)
ORDER BY branch_avg DESC;
```

Branch Whose Average Quantity Sold Is More Than Average Quantity Sold

naikdarshan221@gmail.com







Naming Product Line as Good and Bad based on

Average Sales

```
SELECT
    product_line,
    CASE
        WHEN
            avg_sales > (SELECT
                    ROUND(AVG(total), 2)
                FROM
                    sales)
        THEN
            'Good'
        ELSE 'Bad'
    END AS product_category,
    avg_sales,
    (SELECT
            ROUND(AVG(total), 2)
        FROM
            sales) AS avg_total_sales
FROM
    (SELECT
        product_line, ROUND(AVG(total), 2) AS avg_sales
    FROM
        sales
    GROUP BY product_line) AS p1;
```

	product_line	product_category	avg_sales	avg_total_sales
•	Food and beverages	Good	322.67	322.50
	Health and beauty	Good	323.54	322.50
	Sports and travel	Good	330.90	322.50
	Fashion accessories	Bad	305.09	322.50
	Home and lifestyle	Good	336.64	322.50
	Electronic accessories	Bad	318.24	322.50





Gender Wise Product Preferences

```
select gender, product_line, product_count
from(
    select gender, product_line, product_count, rank() over(partition by gender order by product_count desc) as ranking
from(
    select gender, product_line, count(product_line) as product_count
    from sales
    group by gender, product_line
    order by gender, product_count desc) as p1)as p2
    where ranking in (1,2, 3);
```

	gender	product_line	product_count
•	Female	Fashion accessories	96
	Female	Food and beverages	90
	Female	Sports and travel	86
	Male	Health and beauty	88
	Male	Electronic accessories	86
	Male	Food and beverages	84





Product Line Wise Average Rating

```
353 • SELECT
354     product_line, ROUND(AVG(rating), 2) AS avg_rating
355     FROM
356     sales
357     GROUP BY product_line
358     ORDER BY avg_rating DESC;
```

	product_line	avg_rating
•	Food and beverages	7.11
	Fashion accessories	7.03
	Health and beauty	6.98
	Electronic accessories	6.91
	Sports and travel	6.86
	Home and lifestyle	6.84





Time Of The Day Most Ratings Received

```
521 • SELECT

522 day_time, COUNT(rating) AS count_of_rating

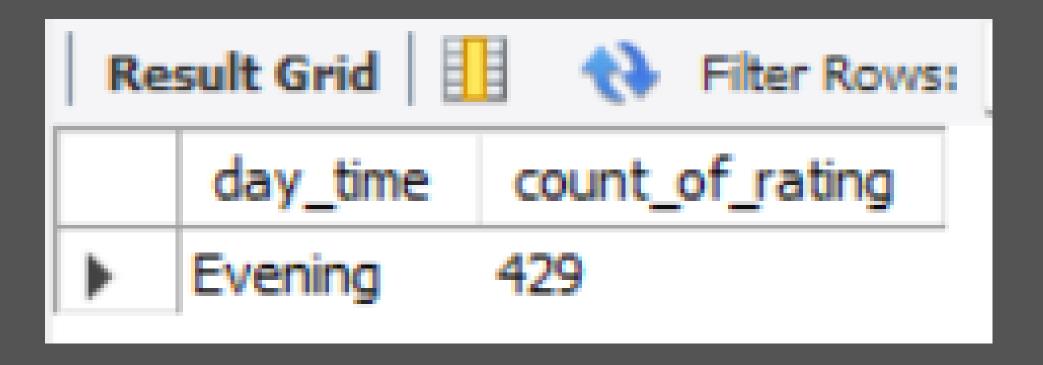
523 FROM

524 sales

525 GROUP BY day_time

526 ORDER BY count_of_rating DESC

527 LIMIT 1;
```







naikdarshan221@gmail.com

```
SELECT

day_of_week, AVG(rating) AS avg_rating

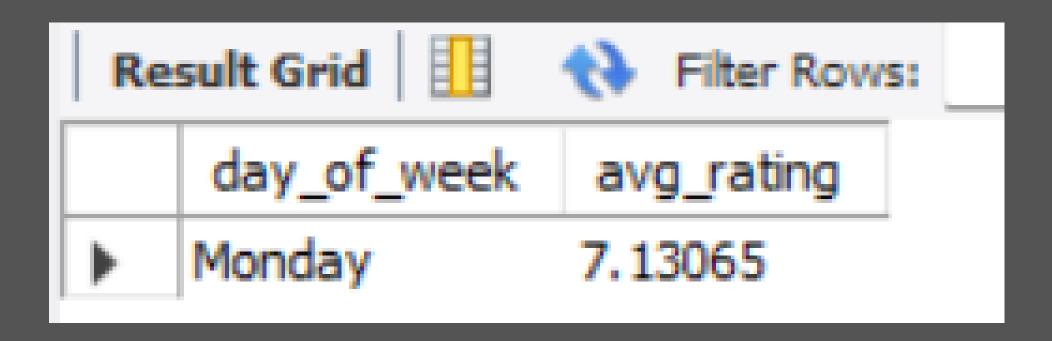
FROM

sales

GROUP BY day_of_week

ORDER BY avg_rating DESC

LIMIT 1;
```







<u>Day That Has Highest</u> <u>Average Rating Branch</u> <u>Wise</u>

```
select branch, day_of_week, round(avg_rating,2) as avg_rating
from(
select branch, day_of_week, avg(rating) as avg_rating, rank() over(partition by branch order by avg(rating) desc) as ranking
from sales
group by branch, day_of_week
order by branch) as b
where ranking=1;
```

	branch	day_of_week	avg_rating
•	A	Friday	7.31
	В	Monday	7.27
	С	Saturday	7.23



Thank You Express your Views On Analysis In Comment Section