

MY SQL QUERIES

COFFEE SHOP SALES PROJECT

Table cleaning

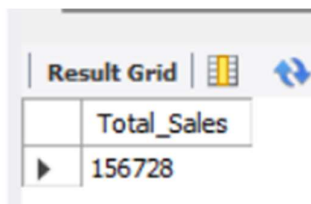
```
1 Create Database coffee_shop_sales;

2
3 -- Using Table data import wizard load the csv file into the database.
4
5 SELECT * FROM coffee_shop;
6
7 describe coffee_shop;
8
9 /* This data is not in the perfect form it has improper data type in some
10 field and few unknown characters are also available in field name
11 so we will clean the date in perfect format. here for this case we
12 will disable safe update mode as we are not
13 using primary key to update or alter the table*/
14
15 SET SQL_SAFE_UPDATES = 0;
16
17 UPDATE coffee_shop
18 SET transaction_date = STR_TO_DATE(transaction_date, '%d-%m-%Y');
19
20
21 -- transaction_date is in text format now we will convert it into proper date format.
22
23 alter table coffee_shop
24 modify column transaction_date date;
25
26
27 UPDATE coffee_shop
28 SET transaction_time = STR_TO_DATE(transaction_time, '%H:%i:%s');
29
30
31 -- it is good practice to convert the table in proper format and then alter it.
32 -- ther is a possibility that the table is in perfecct format then also we overdo it to be sure.
33
34 alter table coffee_shop
35 modify column transaction_time time;
36
37 describe coffee_shop;
38
39 -- now we will change first column name 'i»transaction_id' to correct name 'transaction_id'.
40
41 alter table coffee_shop
42 change column i»transaction_id transaction_id int;
43
44
45 -- Now the table is cleaned and ready to analyse.
```

1. Calculate the total sales for each respective month.

-- 1 • Calculate the total sales for each respective month.

```
select Round(sum(unit_price * transaction_qty)) as Total_Sales
from coffee_shop
where month(transaction_date) = 5; -- for may month we can change the month here, cross check in powerbi also.
```

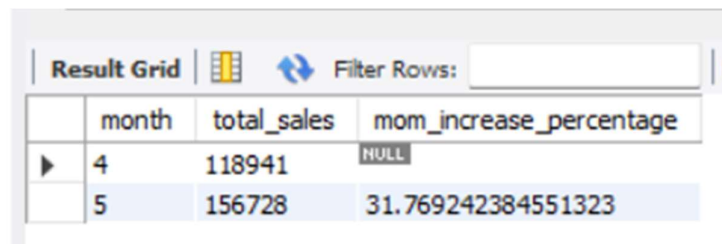


Total_Sales
156728

2. Determine the month-on-month increase or decrease on sales.

-- 2 • Determine the month-on-month increase or decrease on sales.

```
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)) - 1
    ) * 100 AS mom_increase_percentage
FROM coffee_shop
WHERE MONTH(transaction_date) IN (4, 5) -- for months of April and May
GROUP BY MONTH(transaction_date)
ORDER BY MONTH(transaction_date);
```



month	total_sales	mom_increase_percentage
4	118941	NULL
5	156728	31.769242384551323

3. Calculate the difference in sales between the selected month and the previous month.

-- 3 • Calculate the difference in sales between the selected month and the previous month.

```
With cte as ( SELECT
    MONTH(transaction_date) AS month,
    ROUND(SUM(unit_price * transaction_qty)) AS total_sales
FROM coffee_shop
WHERE MONTH(transaction_date) IN (4, 5) -- for months of April and May
GROUP BY MONTH(transaction_date)
ORDER BY MONTH(transaction_date))

SELECT
    (SELECT total_sales FROM cte WHERE month = 5) -
    (SELECT total_sales FROM cte WHERE month = 4) AS difference;
```

Result Grid	
	difference
▶	37787

4. Calculate the total number of orders for each respective month.

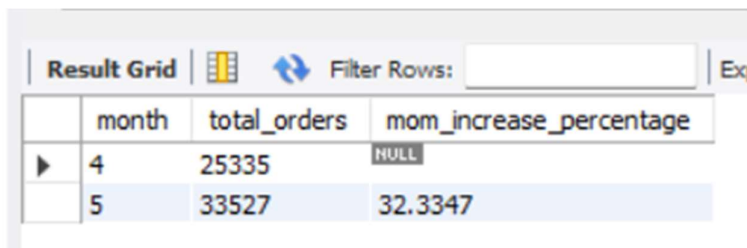
-- 1 • Calculate the total number of orders for each respective month.

```
select count(transaction_id) as Total_Order
from coffee_shop
where month(transaction_date) = 3;
-- for march month we can change the month here, cross check in powerbi also.
```

Result Grid	
	Total_Order
▶	21229

5. Determine the month-on-month increase or decrease in the number of orders.

```
-- 2 * Determine the month-on-month increase or decrease in the number of orders.
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)) - 1) * 100 AS mom_increase_percentage
FROM coffee_shop
WHERE MONTH(transaction_date) IN (4, 5) -- for April and May
GROUP BY MONTH(transaction_date)
ORDER BY MONTH(transaction_date);
```



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of the SQL query for months 4 and 5. The columns are 'month', 'total_orders', and 'mom_increase_percentage'. For month 4, the total orders are 25335 and the increase is NULL. For month 5, the total orders are 33527 and the increase is 32.3347.

	month	total_orders	mom_increase_percentage
▶	4	25335	NULL
	5	33527	32.3347

6. Calculate the difference in the number of orders between the selected month and the previous month.

```
-- 3 * Calculate the difference in the number of orders between the selected month and the previous month.
-- In result from above query we can find order difference using calculator i.e 33527 - 25335 = 8192 for april and may
```

We can directly use calculator and crosscheck the answer or if want to write the query then it is similar to 3rd Query.

7. Calculate the total Quantity sold for each respective month.

-- 1 • Calculate the total Quantity sold for each respective month.

```
SELECT SUM(transaction_qty) as Total_Quantity_Sold
FROM coffee_shop
WHERE MONTH(transaction_date) = 5; -- for month of (CM-May)
```

Result Grid		Filter
	Total_Quantity_Sold	
▶	48233	

8. Determine the month-on-month increase or decrease in the total quantity sold.

-- 2 • Determine the month-on-month increase or decrease in the total quantity sold.

```
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)) - 1) * 100 AS mom_increase_percentage
FROM coffee_shop
WHERE MONTH(transaction_date) IN (4, 5) -- for April and May
GROUP BY MONTH(transaction_date)
ORDER BY MONTH(transaction_date);
```




Result Grid		Filter Rows:	Export:
	month	total_quantity_sold	mom_increase_percentage
▶	4	36469	NULL
	5	48233	32.2575

9. Calculate the difference in the total quantity sold between the selected month and the previous month.

```
-- 3 • Calculate the difference in the total quantity sold between the selected month and the previous month.  
-- In result from above query we can find total quantity difference using calculator i.e 48233 - 36469 = 11764 for april and may.
```

10. Implement tooltips to display detailed metrics (Sales, order, Quantity) when hovering over a specific day.

```
-- 1• Implement tooltips to display detailed metrics (Sales, order, Quantity) when hovering over a specific day.  
SELECT  
    Concat(round(SUM(unit_price * transaction_qty)/1000, 1), 'K') AS total_sales,  
    Concat(Round(SUM(transaction_qty)/1000,1),'k') AS total_quantity_sold,  
    Concat(Round(COUNT(transaction_id)/1000,1), 'k') AS total_orders  
FROM coffee_shop  
WHERE transaction_date = '2023-05-18'; -- For 18 May 2023
```

Result Grid			Filter Rows:	
	total_sales	total_quantity_sold	total_orders	
	5.6K	1.7k	1.2k	

11. Sales Analysis by weekdays and weekends.

```
-- Note: Here sunday = 1 and hence saturday will be 7
```

```
-- 2• Sales Analysis by weekdays and weekends.
```

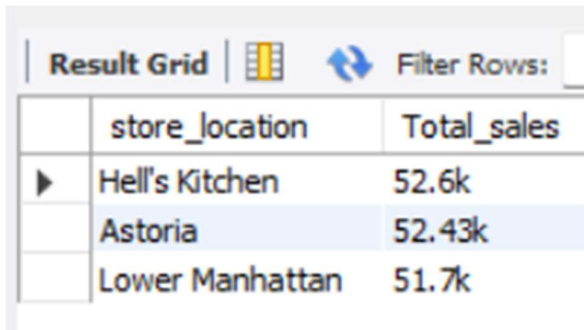
```
select  
case when dayofweek(transaction_date) in (1,7) then 'weekend'  
else 'weekdays'  
end as day_type,  
concat(round(sum(unit_price * transaction_qty)/1000,1),'k') as Total_sales  
from coffee_shop  
where month(transaction_date) = 5 -- for may month  
group by day_type;
```

Result Grid	Filter Rows:
day_type	Total_sales
weekdays	116.6k
weekend	40.1k

12. Sales Analysis by store location.

-- 3* sales analysis by store location

```
select
    store_location,
    concat(round(sum(unit_price * transaction_qty)/1000,2),'k') as Total_sales
from coffee_shop
where month(transaction_date) = 5 -- for may month
group by store_location
order by Total_sales
desc;
```



The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid contains three rows of data. The first row is 'Hell's Kitchen' with 'Total_sales' of '52.6k'. The second row is 'Astoria' with 'Total_sales' of '52.43k'. The third row is 'Lower Manhattan' with 'Total_sales' of '51.7k'.

	store_location	Total_sales
▶	Hell's Kitchen	52.6k
	Astoria	52.43k
	Lower Manhattan	51.7k

13. Daily Sales Analysis with Average Line

-- 4* Daily Sales Analysis with Average Line

```
with cte as (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_month_sales
FROM coffee_shop
WHERE MONTH(transaction_date) = 5 -- Filter for May
GROUP BY day_of_month) -- here we want for each and every day so we have group by day of month and not month.

SELECT day_of_month,
CASE
    WHEN total_sales > avg_month_sales THEN 'Above Average'
    WHEN total_sales < avg_month_sales THEN 'Below Average'
    ELSE 'Average'
END AS sales_status,
total_sales, avg_month_sales from cte;
```


Result Grid			
Filter Rows:		Export:	Wrap Cell Content:
day_of_month	sales_status	total_sales	avg_month_sales
1	Below Average	4731.449999999999	5055.7341935483855
2	Below Average	4625.499999999997	5055.7341935483855
3	Below Average	4714.599999999994	5055.7341935483855
4	Below Average	4589.699999999995	5055.7341935483855
5	Below Average	4700.999999999997	5055.7341935483855
6	Below Average	4205.149999999998	5055.7341935483855
7	Below Average	4542.699999999998	5055.7341935483855
8	Above Average	5604.209999999995	5055.7341935483855
9	Above Average	5100.969999999997	5055.7341935483855
10	Above Average	5256.329999999999	5055.7341935483855
11	Below Average	4850.059999999996	5055.7341935483855
12	Below Average	4681.1299999999965	5055.7341935483855
13	Above Average	5511.529999999999	5055.7341935483855
14	Below Average	5052.649999999999	5055.7341935483855
15	Above Average	5384.9800000000005	5055.7341935483855
16	Above Average	5542.129999999997	5055.7341935483855
17	Above Average	5418.000000000001	5055.7341935483855
18	Above Average	5583.470000000000	5055.7341935483855

14.Sales Analysis by product Category

-- 5• Sales analysis by product category.

Select

```
product_category,
round(sum(unit_price * transaction_qty),2) as total_sales
from coffee_shop
where month(transaction_date) = 5
group by product_category
order by total_sales
Desc;
```

Result Grid		
Filter Rows:		
	product_category	total_sales
►	Coffee	60362.85
	Tea	44539.85
	Bakery	18565.52
	Drinking Chocolate	16319.75
	Coffee beans	8768.95
	Branded	2889
	Loose Tea	2395.15
	Flavours	1905.6
	Packaged Chocolate	981.09

15. Top 10 Product by sales

-- 6• Top 10 product by sales

Select

```
product_type,  
round(sum(unit_price * transaction_qty),2) as total_sales  
from coffee_shop  
where month(transaction_date) = 5  
group by product_type  
order by total_sales  
Desc  
limit 10;
```

Result Grid	Filter Rows:
product_type	total_sales
Barista Espresso	20423.75
Brewed Chai tea	17427.35
Hot chocolate	16319.75
Gourmet brewed coffee	15559.2
Brewed herbal tea	10930
Brewed Black tea	10778
Premium brewed coffee	8739.2
Organic brewed coffee	8350.2
Scone	8305.28
Drip coffee	7290.5

16. Sales Analysis by Days and Hours

-- 7• Sales Analysis by Days and Hours

Select

```
sum(unit_price * transaction_qty) as Total_Sales,  
Sum(transaction_qty) as Total_qty_Sold,  
Count(*) as Total_orders  
From coffee_shop  
where month(transaction_date) = 5 -- For May  
And dayofweek(transaction_date) = 2 -- Monday  
and Hour(Transaction_time) = 8; -- Hour no 8
```

Result Grid	Filter Rows:	Export
Total_Sales	Total_qty_Sold	Total_orders
2697.0299999999999	819	572

17.Total Hour wise sale

-- 8.1 Total sales hour wise

Select

```
hour(transaction_time),  
sum(unit_price * transaction_qty) as total_sales  
from coffee_shop  
where month(transaction_date) = 5  
Group by Hour(Transaction_time)  
order by Hour(Transaction_time) ;
```

Result Grid			Filter Rows:
	hour(transaction_time)	total_sales	
▶	6	4912.930000000001	
	7	14350.680000000037	
	8	18822.310000000003	
	9	19145.270000000022	
	10	19639.130000000001	
	11	10312.160000000014	
	12	8869.790000000008	
	13	9379.210000000008	
	14	9057.660000000007	
	15	9525.150000000002	
	16	9154.310000000012	
	17	8966.850000000013	
	18	7679.909999999997	
	19	6256.469999999997	
	20	655.9300000000002	

18. Weekday's individual sale from Monday to Sunday.

-- 8.2 TO GET SALES FROM MONDAY TO SUNDAY FOR MONTH OF MAY

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

WHERE

MONTH(transaction_date) = 5 -- Filter for May (month number 5)

GROUP BY Day_of_Week;

Result Grid			Filter Rows:
	Day_of_Week	Total_Sales	
▶	Monday	25221	
	Tuesday	25347	
	Wednesday	25465	
	Thursday	20254	
	Friday	20341	
	Saturday	20795	
	Sunday	19305	