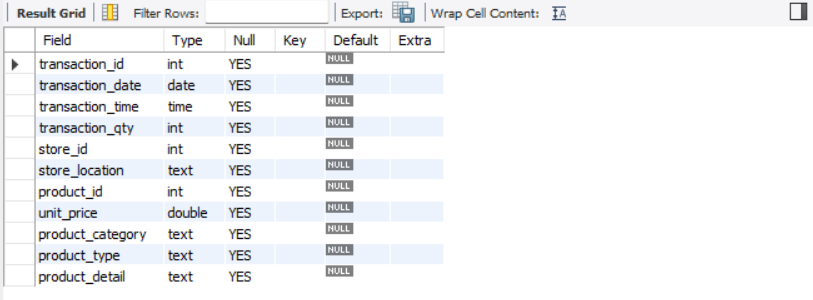
# Coffee Shop Sales Analysis - SQL Queries

Cleaning and Modifying the data

create database coffee\_data;  
  
show databases;  
  
use Coffee\_data;  
  
select count(transaction\_date) from coffee\_shop;  
  
alter table coffee\_shop   
change column ï»¿transaction\_id transaction\_id int;  
  
update coffee\_shop   
set transaction\_date = str\_to\_date(transaction\_date, '%d-%m-%y');  
  
SET SQL\_SAFE\_UPDATES = 0;  
  
UPDATE coffee\_shop   
SET transaction\_date = STR\_TO\_DATE(transaction\_date, '%d-%m-%Y')  
WHERE transaction\_id IS NOT NULL;  
  
alter table coffee\_shop  
modify column transaction\_date DATE;  
  
alter table coffee\_shop  
modify column transaction\_time time;

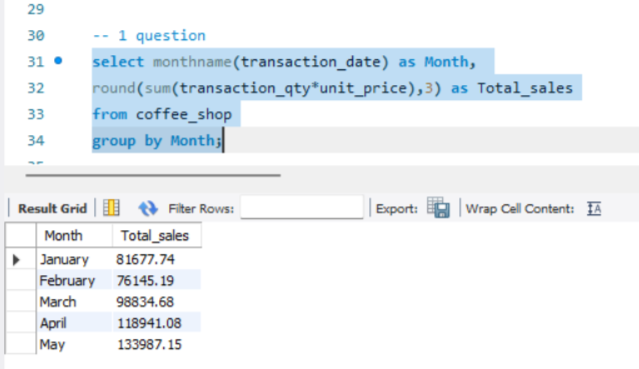
describe coffee\_shop;



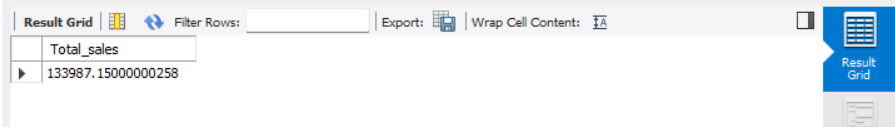
Total Sales Analysis

1. Calculate total Sales for Each Month

select monthname(transaction\_date) as Month,   
round(sum(transaction\_qty\*unit\_price),3) as Total\_sales  
from coffee\_shop  
group by Month;

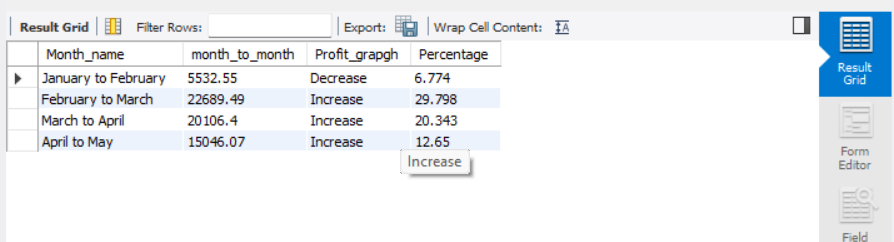
  
2. Calculate total Sales for selected month

select sum(transaction\_qty\*unit\_price) as Total\_sales  
from coffee\_shop  
where month(transaction\_date)=5;



3) Determine Month on Month Decrease or Increase in Sales

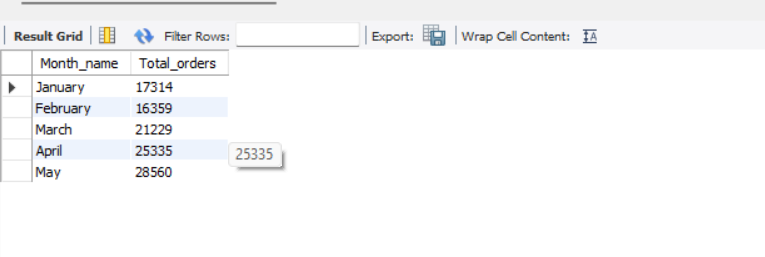
with a as (select monthname(transaction\_date) as Month,  
month(transaction\_date) as month\_num,   
round(sum(transaction\_qty\*unit\_price),3) as Total\_sales  
from coffee\_shop  
group by Month, month\_num)  
  
select concat(month, " to ", lead(month) OVER (order by month\_num )) as Month\_name,   
abs(round(total\_sales-lead(total\_sales) OVER (order by month\_num), 3)) as month\_to\_month,  
case   
 when total\_sales-lead(total\_sales) OVER (order by month\_num)>0 then "Decrease"  
 else  
 "Increase"  
end as Profit\_grapgh,  
abs(round(((total\_sales-lead(total\_sales) OVER (order by month\_num))/  
total\_sales)\*100, 3)) as Percentage  
from a   
limit 4;



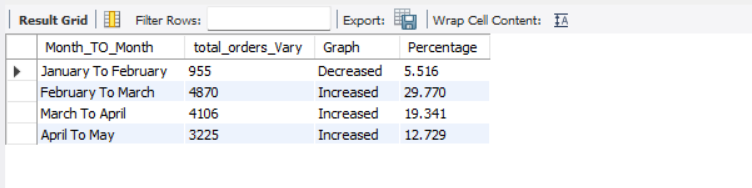
-- Total Orders Analysis

1) Calculate the total orders of each month

select monthname(transaction\_date) as Month\_name,   
count(transaction\_id) as Total\_orders   
from coffee\_shop  
group by Month\_name;

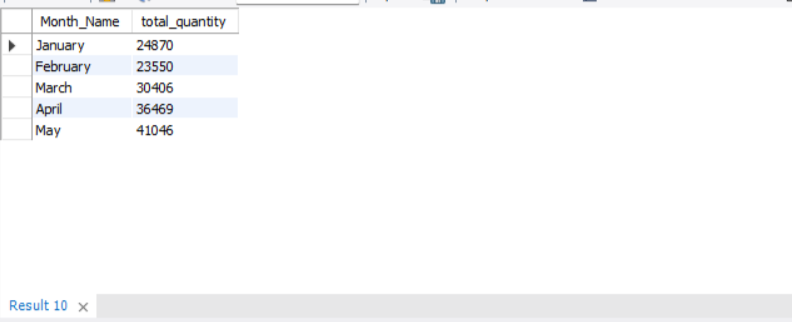
  
  
2) Month to month decrease or increase of total order

with a as (select monthname(transaction\_date) as Month\_name,   
month(transaction\_date) as Month\_num,  
count(transaction\_id) as Total\_orders   
from coffee\_shop  
group by Month\_name, Month\_num)  
  
select   
concat(Month\_name, " To ", lead(month\_name) over (order by month\_num))   
as Month\_TO\_Month,  
abs(total\_orders-lead(total\_orders) Over (order by month\_num)) as total\_orders\_Vary,  
  
Case  
When (total\_orders-lead(total\_orders) Over (order by month\_num)) < 0   
then "Increased"  
else  
"Decreased"  
end as Graph,  
abs(round(((total\_orders-lead(total\_orders) Over (order by month\_num))/total\_orders)\*100,3))  
as Percentage  
from a  
limit 4;

  
  
-- Total Quantity Analysis

1) Total quantity sold per each month

select monthname(transaction\_date) as Month\_Name,  
sum(transaction\_qty) as total\_quantity  
from Coffee\_shop  
group by Month\_name;

  
  
2) Month on Month increase and decrease in total quantity

With Tab as (select monthname(transaction\_date) as Month\_Name,  
month(transaction\_date) as Month\_num,  
sum(transaction\_qty) as total\_qt  
from Coffee\_shop  
group by Month\_name, month\_num)  
  
select   
concat(Month\_name, " To ", lead(month\_name) over (order by month\_num))   
as Month\_TO\_Month,  
abs(total\_qt-lead(total\_qt) Over (order by month\_num)) as total\_orders\_Vary,  
  
Case  
When (total\_qt-lead(total\_qt) Over (order by month\_num)) < 0   
then "Increased"  
else  
"Decreased"  
end as Graph,  
abs(round(((total\_qt-lead(total\_qt) Over (order by month\_num))/total\_qt)\*100,3))  
as Percentage  
from tab  
limit 4;

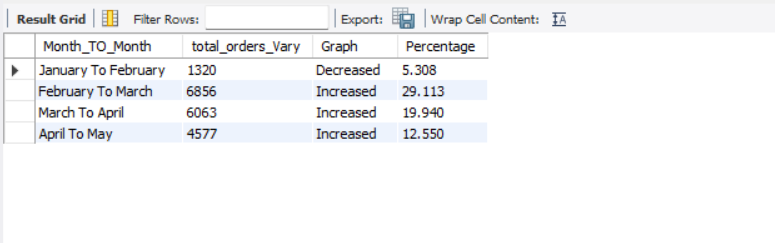
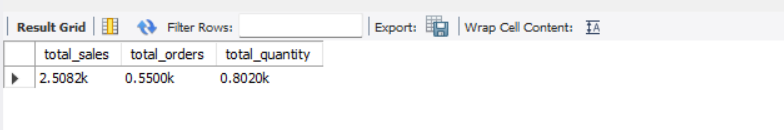
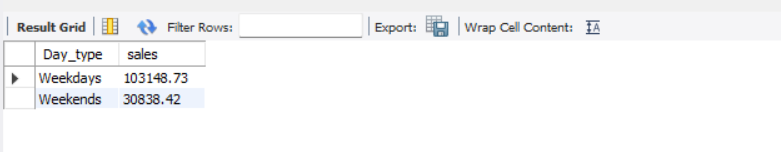


Chart Requirements   
**1) Detailed metrics for specific day**

select concat((round(sum(unit\_price\* transaction\_qty),3))/1000,"k") as total\_sales,  
concat(count(transaction\_id)/1000,"k") as total\_orders,  
concat(sum(transaction\_qty)/1000,"k") as total\_quantity  
from coffee\_shop  
where transaction\_date='2023-01-01';

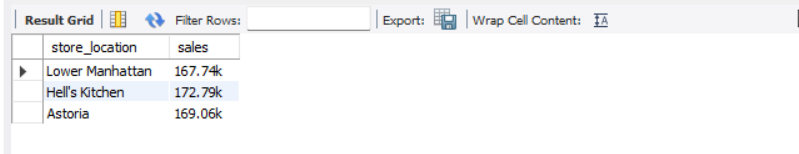
  
**2) Sales data with respect weekdays and weekend**

select   
case  
 when dayname(transaction\_date) in ("Sunday", "Saturday") then "Weekends"  
 else   
 "Weekdays"  
 end as Day\_type,  
 round(sum(transaction\_qty\*unit\_price),3) as sales  
 from coffee\_shop  
 where month(transaction\_date)=5  
 group by Day\_type;



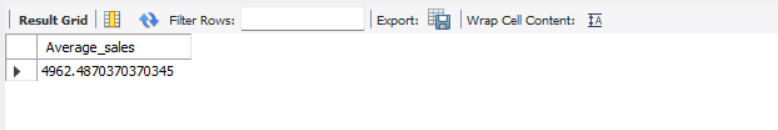
**3) Sales analysis by store locations**

select store\_location,   
concat(round((sum(transaction\_qty\*unit\_price)/1000),2),"k") as sales  
from coffee\_shop  
group by store\_location;

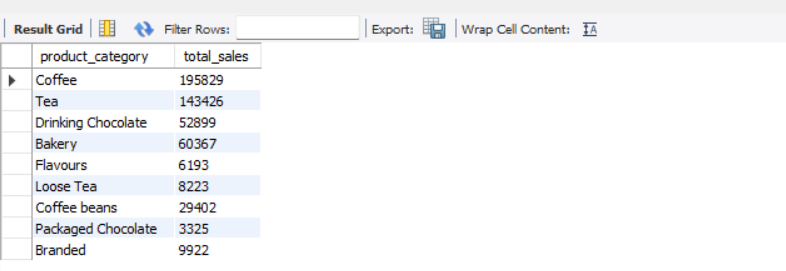


**4) Average Daily sales of selected month**

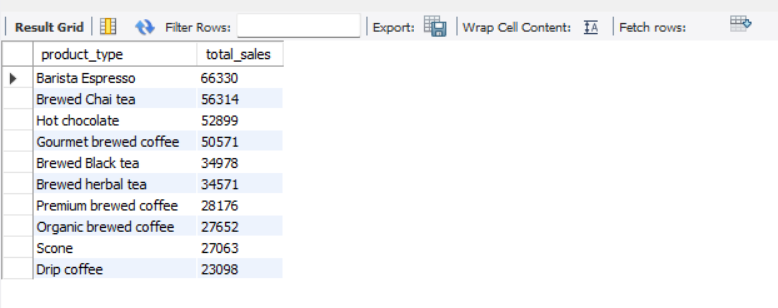
select avg(total\_sales) as Average\_sales  
from   
(select sum(transaction\_qty\*unit\_price) as total\_sales  
from Coffee\_shop  
Where month(transaction\_date)=5  
group by transaction\_date) as Internal ;

  
  
**5) Sales Analysis by Product Category**

select product\_category, round(sum(transaction\_qty\*unit\_price)) as total\_sales  
from coffee\_shop  
group by product\_category;

  
  
**6) Top 10 products by sales volume**

select product\_type, round(sum(transaction\_qty\*unit\_price)) as total\_sales  
from coffee\_shop  
group by product\_type  
order by total\_sales desc   
limit 10;

  
  
**7) Sales analysis by days and hour**  
select   
Hour(transaction\_time) as Hour,  
round(sum(transaction\_qty\*unit\_price),2) as Sales,  
sum(transaction\_qty) as Total\_quantity,  
count(transaction\_id) as Total\_Orders  
from coffee\_shop  
where day(transaction\_date)=3 and  
month(transaction\_date)=3  
group by Hour;

  
  
select   
dayname(transaction\_date) as day,  
hour(transaction\_time) as Hour,  
round(sum(transaction\_qty\*unit\_price),2) as Sales,  
sum(transaction\_qty) as Total\_quantity,  
count(transaction\_id) as Total\_Orders  
from coffee\_shop  
where day(transaction\_date)=3 and  
month(transaction\_date)=3  
group by day, hour;

