**COFFEE SHOP SALES**

*SQL QUERIES*

1. **Data Preview (Top 10)**

SELECT

\*

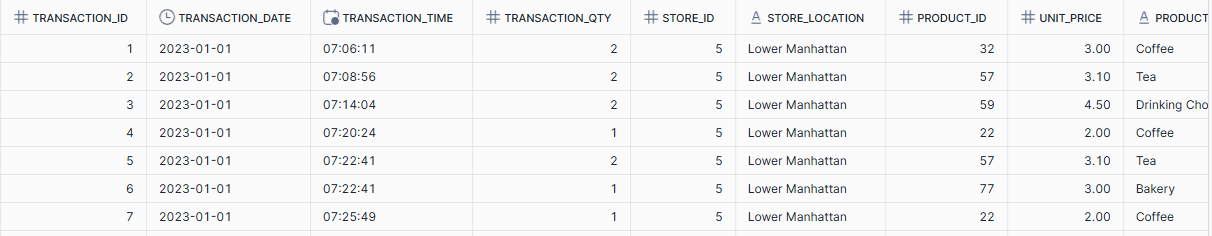
FROM

"MYDB"."PUBLIC"."COFFEESHOPSALES"

LIMIT

10;

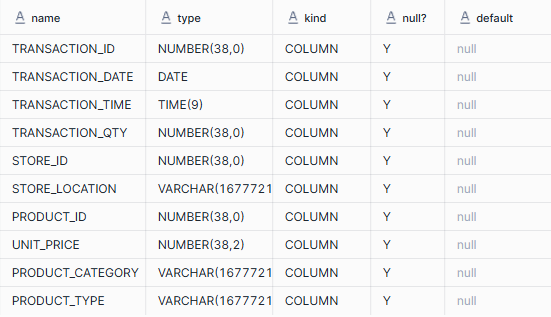
**Output:**



1. **Display data types**

-- Display table data types --

DESCRIBE TABLE coffeeshopsales;



1. **Total Sales**

-- Calculating total sales --

SELECT

SUM(transaction\_qty\*unit\_price) AS TOTAL\_SALES

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 5; -- (May)

Output:



1. **Month on Month Percentage Increase in Sales**

SELECT

MONTH(transaction\_date) AS MONTH,

ROUND(SUM(transaction\_qty\*unit\_price)) AS TOTAL\_SALES,

ROUND((SUM(transaction\_qty\*unit\_price) - LAG(SUM(transaction\_qty\*unit\_price),1) OVER (ORDER BY MONTH(transaction\_date)))/

LAG(SUM(transaction\_qty\*unit\_price), 1) OVER (ORDER BY MONTH(transaction\_date))\*100,2) AS MOM\_INCREASE\_PERCENTAGE

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) IN (4, 5) – (April, May)

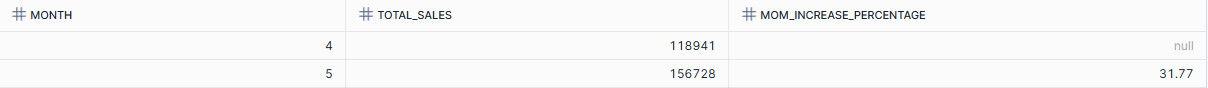
GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);

**Output:**



1. **Total Orders**

-- Calculating total orders --

SELECT

COUNT(transaction\_id) AS TOTAL\_ORDERS

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 5; (May)

**Output:**



1. **Month on Month Percentage Increase in Orders**

-- Calculating month on month increase in percentage of orders --

SELECT

MONTH(transaction\_date) AS MONTH,

COUNT(transaction\_id) AS TOTAL\_ORDERS,

ROUND((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))\*100,2) AS MOM\_INCREASE\_PERCENTAGE

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) IN (4, 5) – (April & May)

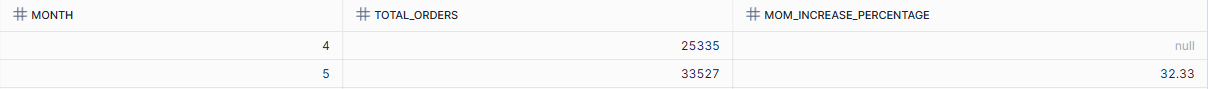
GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);

**Output:**



1. **Total Quantity**

-- Calculating total quantity --

SELECT

SUM(transaction\_qty) AS TOTAL\_QUANTITY\_SOLD

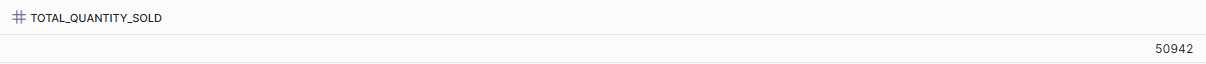
FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 6; -- (June)

**Output:**



1. **Month on Month Percentage Increase in Quantity**

-- Calculating month on month increase in percentage of quantity --

SELECT

MONTH(transaction\_date) AS MONTH,

SUM(transaction\_qty) AS TOTAL\_QUANTITY\_SOLD,

ROUND((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))\*100,2) AS MOM\_INCREASE\_PERCENTAGE

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) IN (4, 5) – (April & May)

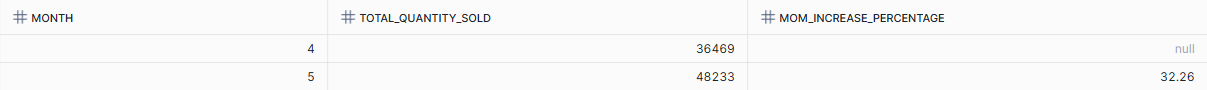
GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);

**Output:**



1. **Calculating Sales on Weekdays/Weekends in a Particular Month**

-- Calculating sales on weekdays and weekends

SELECT

CASE WHEN DAYOFWEEK(transaction\_date) IN (1, 7) THEN 'Weekends'

ELSE 'Weekdays'

END AS day\_type,

SUM(unit\_price\*transaction\_qty) AS Total\_Sales

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 5

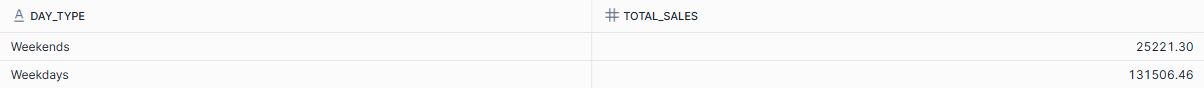
GROUP BY

CASE WHEN DAYOFWEEK(transaction\_date) IN (1, 7) THEN 'Weekends'

ELSE 'Weekdays'

END

**Output:**



1. **Calculating sales as per each store location**

-- Calculating sales per store location --

SELECT

store\_location,

SUM(transaction\_qty\*unit\_price) AS Total\_Sales

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 5

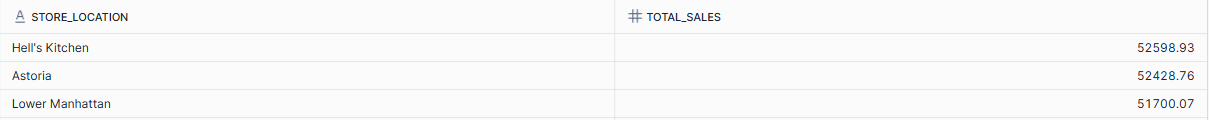
GROUP BY

store\_location

ORDER BY

Total\_Sales DESC;

**Output:**



1. **Calculating average sales for each day in a month**

-- Calculating average sales for every day in a month --

SELECT

ROUND(AVG(Total\_Sales), 2) AS Average\_Sales

FROM(

SELECT

ROUND(SUM(transaction\_qty\*unit\_price), 2) AS Total\_Sales

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 5 -- May

GROUP BY

transaction\_date

) AS inner\_query;

**Output:**



1. **Calculating daily sales in a month as ‘Above Average’ & ‘Below Average’**

-- Categorizing daily sales in a month as 'Above average' & 'Below average'

SELECT

Day\_Month,

CASE

WHEN Total\_Sales > Average\_Sales THEN 'Above Average'

WHEN Total\_Sales < Average\_Sales THEN 'Below Average'

ELSE 'Equal to Average'

END AS Sales\_Status,

Total\_Sales

FROM(

SELECT

DAY(transaction\_date) AS Day\_Month,

SUM(transaction\_qty\*unit\_price) AS Total\_Sales,

AVG(SUM(transaction\_qty\*unit\_price)) OVER() AS Average\_Sales

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 5

GROUP BY

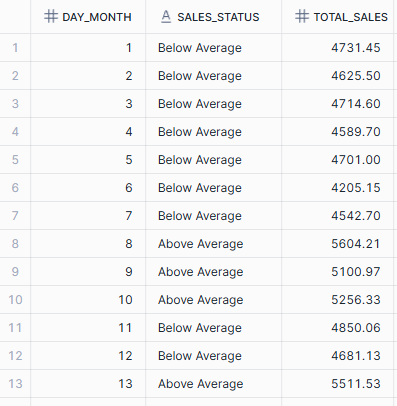
DAY(transaction\_date)

)

ORDER BY

Day\_Month;

**Output:**





1. **Sales by product category**

-- Sales by product category --

SELECT

product\_category,

SUM(transaction\_qty\*unit\_price) AS Total\_Sales

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 5

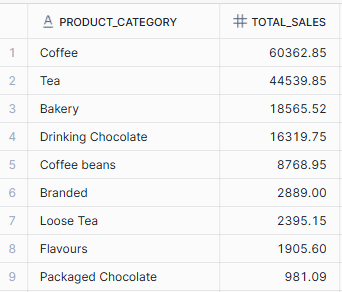
GROUP BY

product\_category

ORDER BY

Total\_Sales DESC;

**Output:**



1. **Sales by product type (top 10)**

-- Sales by product type --

SELECT

product\_type,

SUM(transaction\_qty\*unit\_price) AS Total\_Sales

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 5

GROUP BY

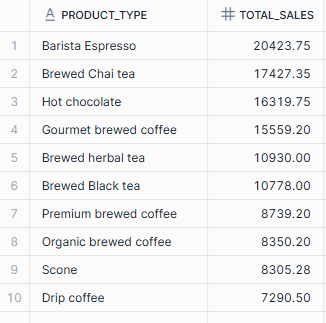
product\_type

ORDER BY

Total\_Sales DESC

LIMIT 10;

**Output:**



1. **Sales Analysis by days & hours**

-- Sales Analysis by days & hours

SELECT

SUM(unit\_price\*transaction\_qty) AS Total\_Sales,

SUM(transaction\_qty) AS Total\_Quantity\_Sold,

COUNT(\*) AS Total\_Orders

FROM

coffeeshopsales

WHERE

MONTH(transaction\_date) = 5

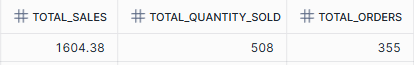
AND

DAYOFWEEK(transaction\_date) = 1

AND

HOUR(transaction\_time) = 14

**Output:**



1. **Sales by days of the week**

-- Sales by days of week --

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

coffeeshopsales

WHERE

MONTH(transaction\_date) = 4

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:**

