COFFEE SHOP SALES ANALYSIS -SQL QUERRIES WITH RESULTS

1. DATA TYPES OF DIFFERENT COLUMNS

```
14
15
16      /* DATA TYPES OF DIFFERENT COLUMNS */
17 •    DESCRIBE coffee_shop_sales;
18
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
transaction_id	int	YES		NULL	
transaction_date	date	YES		NULL	
transaction_time	time	YES		NULL	
transaction_qty	int	YES		NULL	
store_id	int	YES		NULL	
store_location	text	YES		NULL	
product_id	int	YES		NULL	
unit_price	double	YES		NULL	
product_category	text	YES		NULL	
product_type	text	YES		NULL	
product_detail	text	YES		NULL	

2.TOTAL SALES

```
22
23      /* TOTAL SALES */
24 •    SELECT ROUND(SUM(unit_price * transaction_qty)) as Total_Sales
25      FROM coffee_shop_sales
26      WHERE MONTH(transaction_date) = 5 -- for month of (CM-May)
27
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Total_Sales
156728

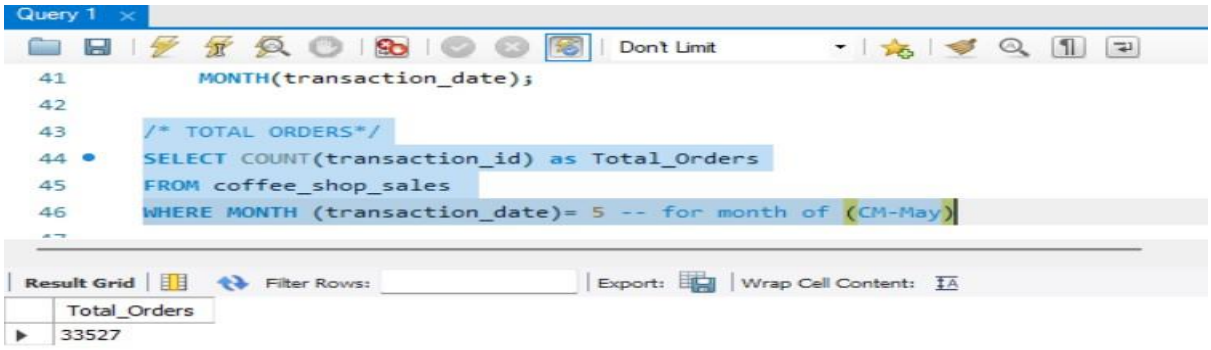
3.TOTAL SALES – MOM DIFFERENCE & MOM GROWTH (APRIL & MAY)

```
27  /* TOTAL SALES KPI - MOM DIFFERENCE AND MOM GROWTH */
28  SELECT
29      MONTH(transaction_date) AS month,
30      ROUND(SUM(unit_price * transaction_qty)) AS total_sales,
31      (SUM(unit_price * transaction_qty) - LAG(SUM(unit_price * transaction_qty), 1)
32      OVER (ORDER BY MONTH(transaction_date))) / LAG(SUM(unit_price * transaction_qty), 1)
33      OVER (ORDER BY MONTH(transaction_date)) * 100 AS mom_increase_percentage
34  FROM
35      coffee_shop_sales
36  WHERE
37      MONTH(transaction_date) IN (4, 5) -- for months of April and May
38  GROUP BY
39      MONTH(transaction_date)
40  ORDER BY
41      MONTH(transaction_date);
42
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

month	total_sales	mom_increase_percentage
4	118941	NULL
5	156728	31.769242384551315

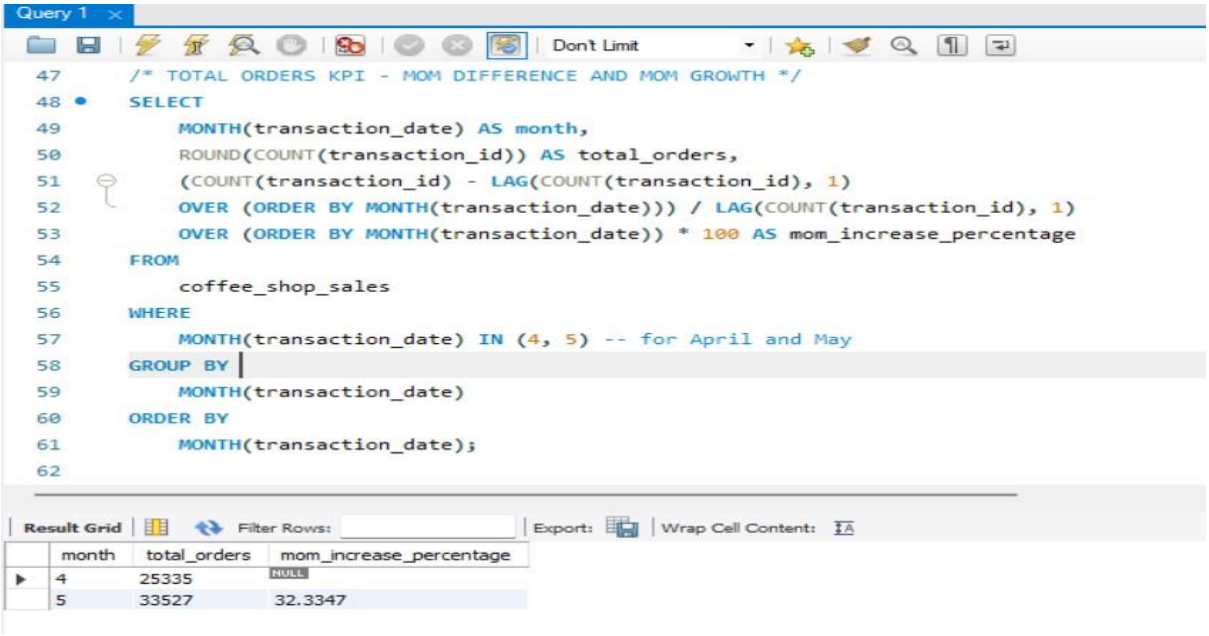
4. TOTAL ORDERS



```
41          MONTH(transaction_date);
42
43      /* TOTAL ORDERS*/
44      SELECT COUNT(transaction_id) as Total_Orders
45      FROM coffee_shop_sales
46      WHERE MONTH (transaction_date)= 5 -- for month of (CM-May)
```

Total_Orders
33527

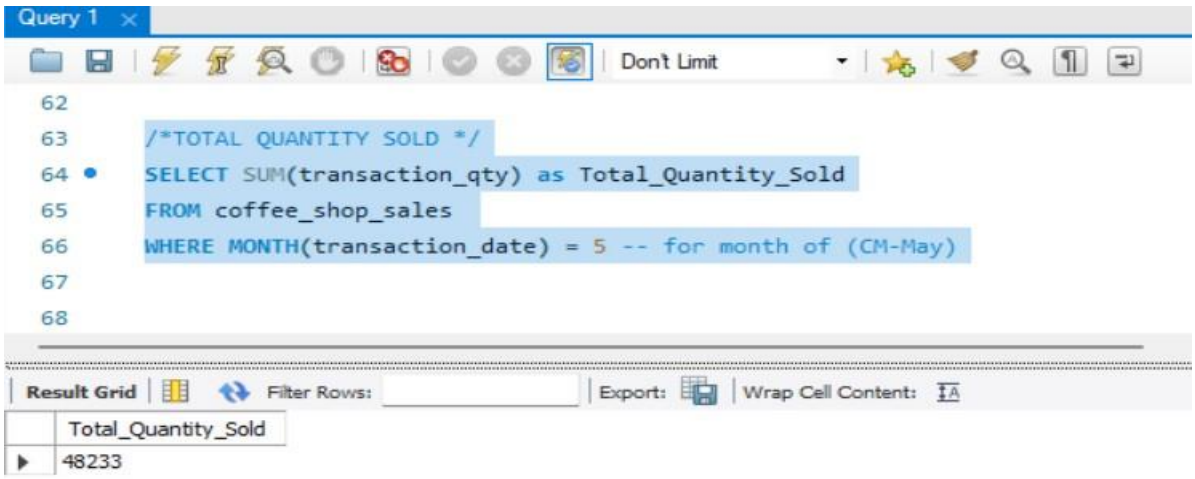
5. TOTAL ORDERS – MOM DIFFERENCE & MOM GROWTH (APRIL & MAY)



```
47      /* TOTAL ORDERS KPI - MOM DIFFERENCE AND MOM GROWTH */
48      SELECT
49          MONTH(transaction_date) AS month,
50          ROUND(COUNT(transaction_id)) AS total_orders,
51          (COUNT(transaction_id) - LAG(COUNT(transaction_id), 1)
52          OVER (ORDER BY MONTH(transaction_date))) / LAG(COUNT(transaction_id), 1)
53          OVER (ORDER BY MONTH(transaction_date)) * 100 AS mom_increase_percentage
54      FROM
55          coffee_shop_sales
56      WHERE
57          MONTH(transaction_date) IN (4, 5) -- for April and May
58      GROUP BY
59          MONTH(transaction_date)
60      ORDER BY
61          MONTH(transaction_date);
62
```

month	total_orders	mom_increase_percentage
4	25335	NULL
5	33527	32.3347

6. TOTAL QUANTITY SOLD



```
62
63      /*TOTAL QUANTITY SOLD */
64      SELECT SUM(transaction_qty) as Total_Quantity_Sold
65      FROM coffee_shop_sales
66      WHERE MONTH(transaction_date) = 5 -- for month of (CM-May)
67
68
```

Total_Quantity_Sold
48233

7. TOTAL QUANTITY SOLD – MOM DIFFERENCE & MOM GROWTH (APRIL & MAY)

```sql
/* TOTAL QUANTITY SOLD KPI - MOM DIFFERENCE AND MOM GROWTH */
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);
```

month	total_quantity_sold	mom_increase_percentage
4	36469	NULL
5	48233	32.2575

8.CALENDER TABLE- DAILY SALES, QUANTITY & TOTAL ORDERS

```sql
/*  CALENDAR TABLE - DAILY SALES, QUANTITY and TOTAL ORDERS */
SELECT
 CONCAT(ROUND(SUM(unit_price * transaction_qty) / 1000, 1),'K') AS total_sales,
 CONCAT(ROUND(COUNT(transaction_id) / 1000, 1),'K') AS total_orders,
 CONCAT(ROUND(SUM(transaction_qty) / 1000, 1),'K') AS total_quantity_sold
FROM
    coffee_shop_sales
WHERE
    transaction_date = '2023-05-18'; -- For 18 May 2023
```

total_sales	total_orders	total_quantity_sold
5.6K	1.2K	1.7K

9. SALES TREND OVER PERIOD

```sql
/* SALES TREND OVER PERIOD */
SELECT AVG(total_sales) AS average_sales
FROM (
    SELECT
        SUM(unit_price * transaction_qty) AS total_sales
    FROM
        coffee_shop_sales
    WHERE
        MONTH(transaction_date) = 5  -- Filter for May
    GROUP BY
        transaction_date
) AS internal_query;
```

average_sales
5055.7341935483855

10. DAILY SALES FOR MAY

Query 1

Don't Limit

```
103    /* DAILY SALES FOR MONTH SELECTED */
104    SELECT
105        DAY(transaction_date) AS day_of_month,
106        ROUND(SUM(unit_price * transaction_qty),1) AS total_sales
107    FROM
108        coffee_shop_sales
109    WHERE
110        MONTH(transaction_date) = 5  -- Filter for May
111    GROUP BY
112        DAY(transaction_date)
113    ORDER BY
114        DAY(transaction_date);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

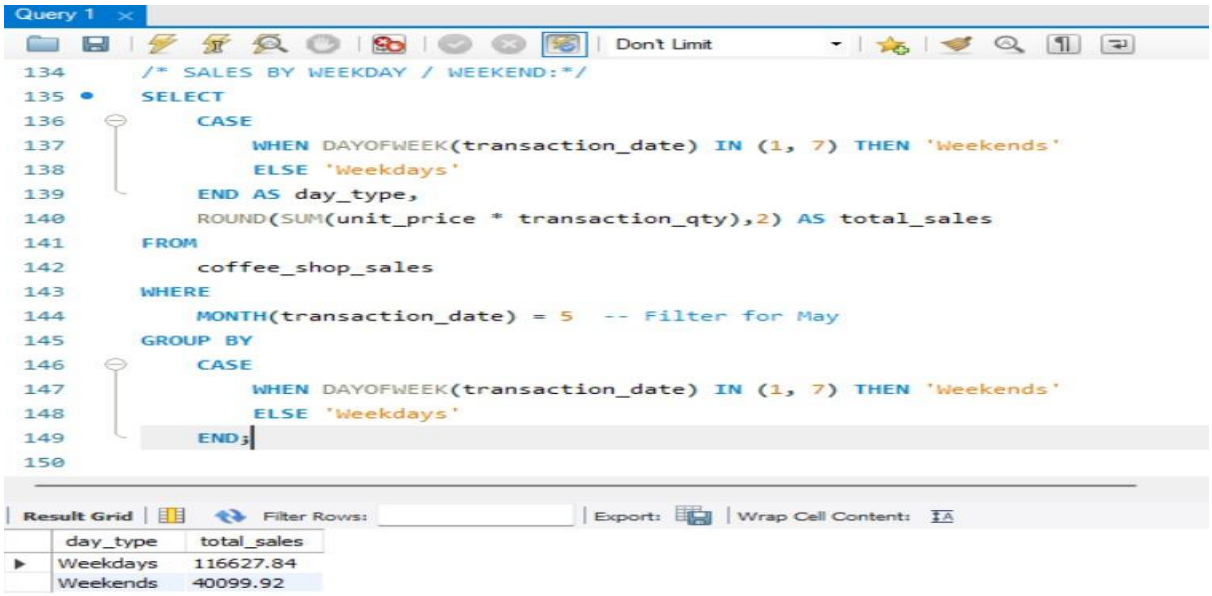
day_of_month	total_sales
1	4731.4
2	4625.5
3	4714.6
4	4589.7
5	4701
6	4205.1
7	4542.7
8	5604.2
9	5101
10	5256.3
11	4850.1
12	4681.1
13	5511.5
14	5052.6
15	5385
16	5542.1
17	5418

11. COMPARING DAILY SALES WITH AVERAGE SALES

```sql
/* COMPARING DAILY SALES WITH AVERAGE SALES - IF GREATER THAN "ABOVE AVERAGE" and LESSER THAN "BELOW AVERAGE" */
SELECT
    day_of_month,
    CASE  WHEN total_sales > avg_sales THEN 'Above Average'
        WHEN total_sales < avg_sales THEN 'Below Average'
        ELSE '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;
```

day_of_month	sales_status	total_sales
1	Below Average	4731.449999999999
2	Below Average	4625.499999999997
3	Below Average	4714.599999999994
4	Below Average	4589.699999999995
5	Below Average	4700.999999999997
6	Below Average	4205.149999999998
7	Below Average	4542.699999999998
8	Above Average	5604.209999999995
9	Above Average	5100.969999999997
10	Above Average	5256.329999999999
11	Below Average	4850.059999999996
12	Below Average	4681.1299999999965
13	Above Average	5511.529999999999
14	Below Average	5052.649999999999
15	Above Average	5384.9800000000005
16	Above Average	5542.129999999997
17	Above Average	5418.000000000001
18	Above Average	5583.470000000001
19	Above Average	5657.880000000005
20	Above Average	5519.280000000003
21	Above Average	5370.810000000003
22	Above Average	5541.16
23	Above Average	5242.910000000001
24	Above Average	5391.45
25	Above Average	5230.8499999999985
26	Above Average	5300.949999999998
27	Above Average	5559.1500000000015
28	Below Average	4338.649999999998
29	Below Average	3959.499999999998
30	Below Average	4835.479999999997
31	Below Average	4684.129999999993

12. SALES BY WEEKDAY / WEEKEND

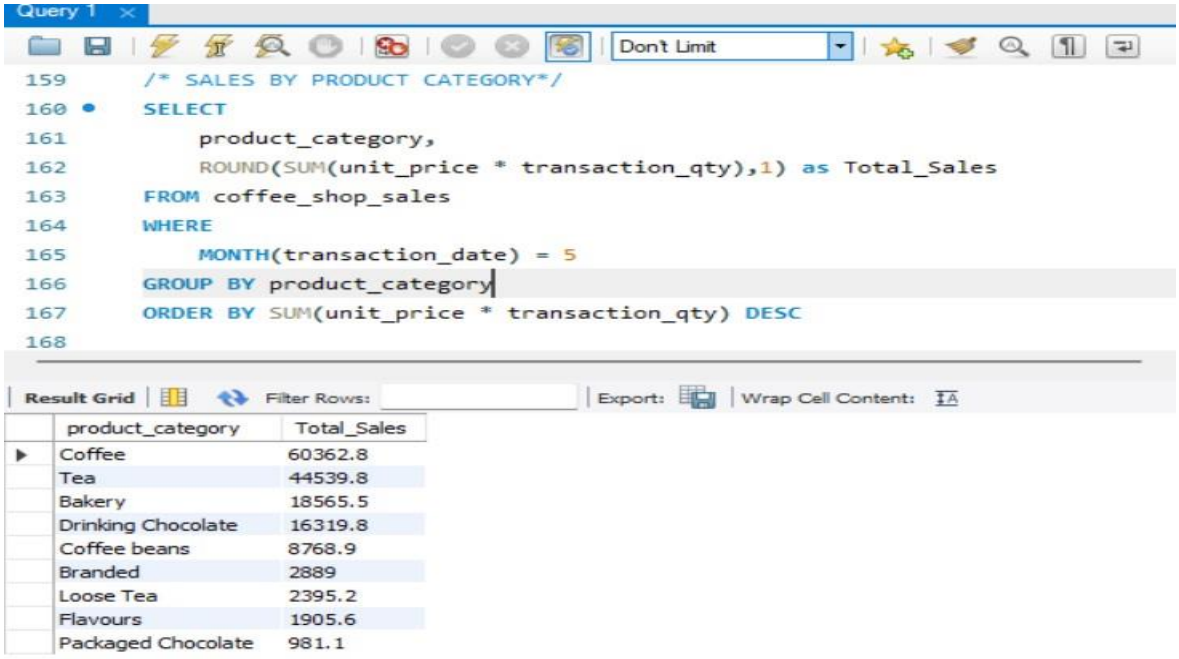


```
/* SALES BY WEEKDAY / WEEKEND:*/
SELECT
    CASE
        WHEN DAYOFWEEK(transaction_date) IN (1, 7) THEN 'Weekends'
        ELSE 'Weekdays'
    END AS day_type,
    ROUND(SUM(unit_price * transaction_qty),2) AS total_sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5   -- Filter for May
GROUP BY
    CASE
        WHEN DAYOFWEEK(transaction_date) IN (1, 7) THEN 'Weekends'
        ELSE 'Weekdays'
    END;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

day_type	total_sales
Weekdays	116627.84
Weekends	40099.92

13. SALES BY PRODUCT CATEGORY

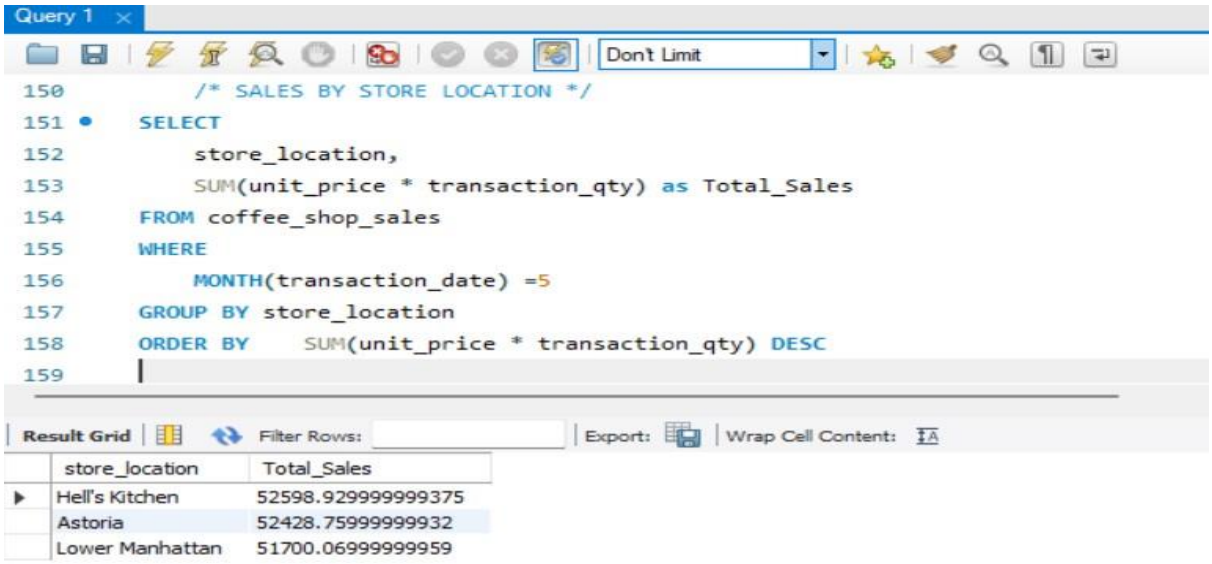


```
/* SALES BY PRODUCT CATEGORY*/
SELECT
    product_category,
    ROUND(SUM(unit_price * transaction_qty),1) as Total_Sales
FROM coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5
GROUP BY product_category
ORDER BY SUM(unit_price * transaction_qty) DESC
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_category	Total_Sales
Coffee	60362.8
Tea	44539.8
Bakery	18565.5
Drinking Chocolate	16319.8
Coffee beans	8768.9
Branded	2889
Loose Tea	2395.2
Flavours	1905.6
Packaged Chocolate	981.1

14. SALES BY STORE LOCATION

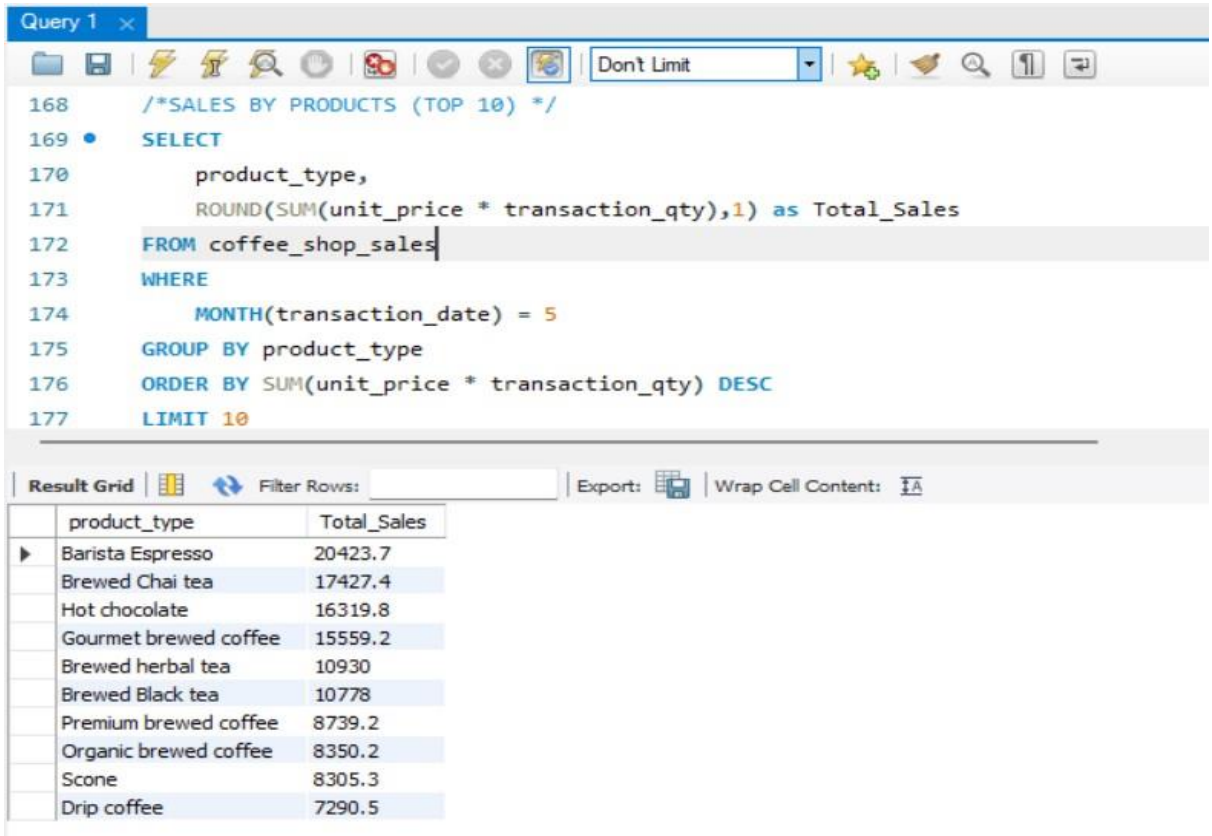


```
150         /* SALES BY STORE LOCATION */
151 •   SELECT
152         store_location,
153         SUM(unit_price * transaction_qty) as Total_Sales
154     FROM coffee_shop_sales
155     WHERE
156         MONTH(transaction_date) =5
157     GROUP BY store_location
158     ORDER BY    SUM(unit_price * transaction_qty) DESC
159
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

store_location	Total_Sales
Hell's Kitchen	52598.929999999375
Astoria	52428.75999999932
Lower Manhattan	51700.06999999959

15. SALES BY PRODUCTS

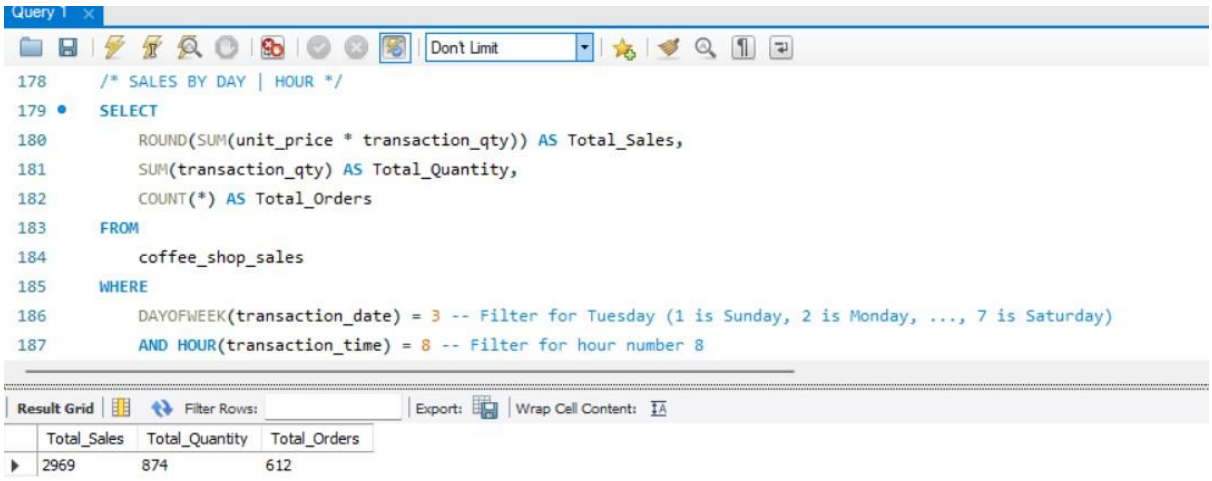


```
168     /*SALES BY PRODUCTS (TOP 10) */
169 •   SELECT
170         product_type,
171         ROUND(SUM(unit_price * transaction_qty),1) as Total_Sales
172     FROM coffee_shop_sales
173     WHERE
174         MONTH(transaction_date) = 5
175     GROUP BY product_type
176     ORDER BY SUM(unit_price * transaction_qty) DESC
177     LIMIT 10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_type	Total_Sales
Barista Espresso	20423.7
Brewed Chai tea	17427.4
Hot chocolate	16319.8
Gourmet brewed coffee	15559.2
Brewed herbal tea	10930
Brewed Black tea	10778
Premium brewed coffee	8739.2
Organic brewed coffee	8350.2
Scone	8305.3
Drip coffee	7290.5

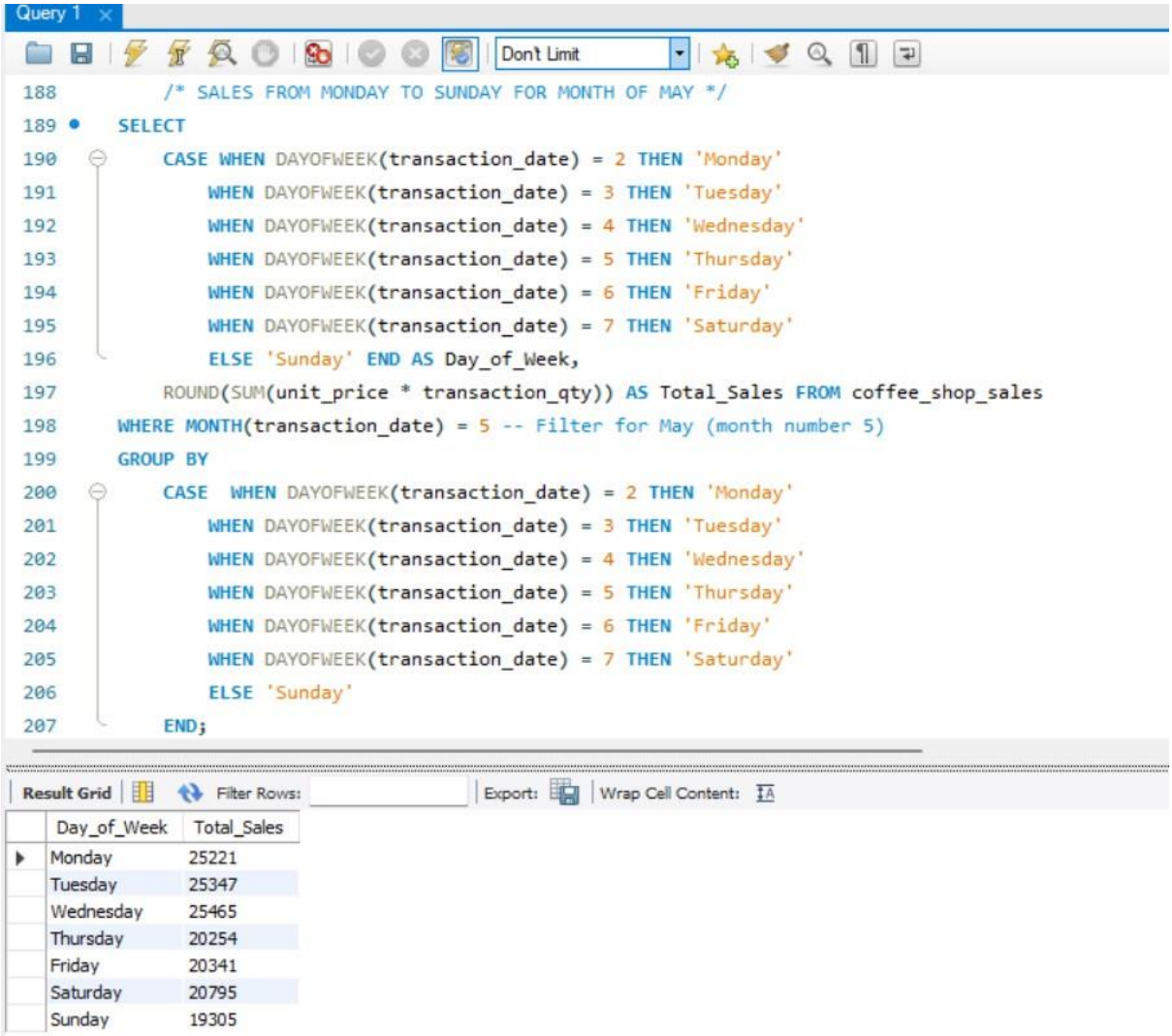
16. SALES BY DAY / HOUR



```
/* SALES BY DAY | HOUR */
SELECT
    ROUND(SUM(unit_price * transaction_qty)) AS Total_Sales,
    SUM(transaction_qty) AS Total_Quantity,
    COUNT(*) AS Total_Orders
FROM
    coffee_shop_sales
WHERE
    DAYOFWEEK(transaction_date) = 3 -- Filter for Tuesday (1 is Sunday, 2 is Monday, ..., 7 is Saturday)
    AND HOUR(transaction_time) = 8 -- Filter for hour number 8
```

Total_Sales	Total_Quantity	Total_Orders
2969	874	612

17. SALES FROM MONDAY TO SUNDAY FOR MONTH OF MAY



```
/* 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_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;
```

Day_of_Week	Total_Sales
Monday	25221
Tuesday	25347
Wednesday	25465
Thursday	20254
Friday	20341
Saturday	20795
Sunday	19305

18. SALES FOR ALL HOURES FOR MONTH OF MAY

Query 1

Don't Limit

```
208    /* SALES FOR ALL HOURS FOR MONTH OF MAY */
209    SELECT
210        HOUR(transaction_time) AS Hour_of_Day,
211        ROUND(SUM(unit_price * transaction_qty)) AS Total_Sales
212    FROM
213        coffee_shop_sales
214    WHERE
215        MONTH(transaction_date) = 5 -- Filter for May (month number 5)
216    GROUP BY
217        HOUR(transaction_time)
218    ORDER BY
219        HOUR(transaction_time);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Hour_of_Day	Total_Sales
6	4913
7	14351
8	18822
9	19145
10	19639
11	10312
12	8870
13	9379
14	9058
15	9525
16	9154
17	8967
18	7680
19	6256
20	656