MY SQL QUERIES

COFFEE SHOP SALES PROJECT

DESCRIBE shop_sales;

TO CONVERT DATE (transaction_date) COLUMN TO PROPER DATE FORMAT

```
UPDATE  shop_sales
SET transaction_date = STR_TO_DATE ( transaction_date ,  '%d-%m-%Y');
 # To get dd-mm-yyyy format
```

ALTER DATE (transaction_date) COLUMN TO DATE DATA TYPE

```
ALTER TABLE shop_sales
MODIFY COLUMN  transaction_date DATE;
# converting to proper format
```

CONVERT TIME (transaction_time) COLUMN TO PROPER DATE FORMAT

```
UPDATE  shop_sales
SET transaction_time = STR_TO_DATE ( transaction_date ,  '%r');
 # our transaction_time column is in hh:mm:ss AM/PM format used %r
```

ALTER TIME (transaction_time) COLUMN TO DATE DATA TYPE

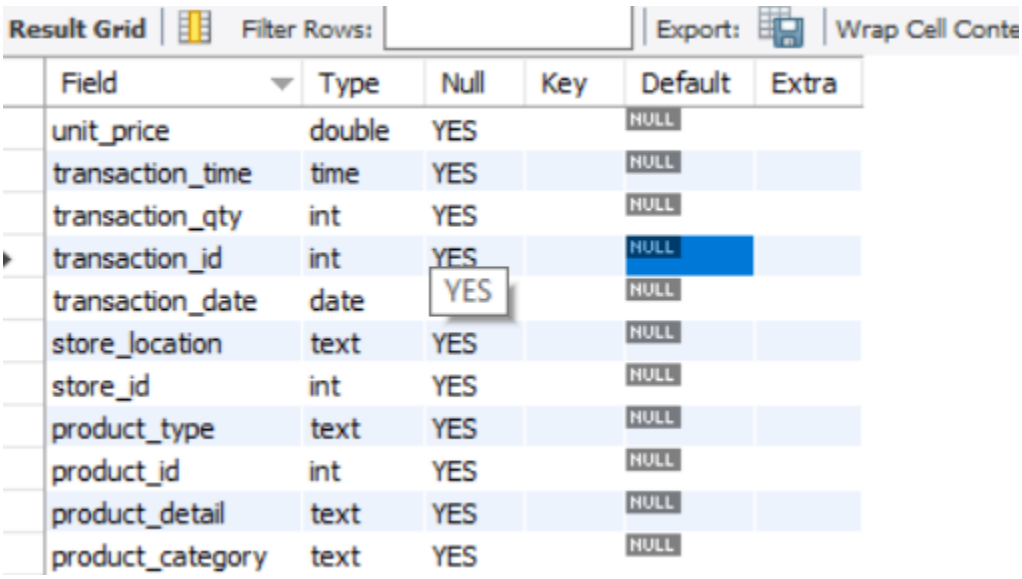
```
ALTER TABLE shop_sales
MODIFY COLUMN transaction_time TIME;
# converted to Time format
```

CHANGE COLUMN NAME `ï»¿transaction_id` to transaction_id

```
ALTER TABLE shop_sales

RENAME COLUMN ï»¿transaction_id  TO transaction_id;
```

DATA TYPES OF DIFFERENT COLUMNS

```
DESC  shop_sales;
```

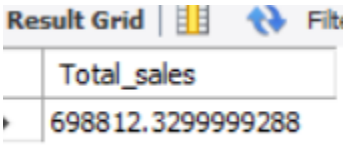


Result Grid | Filter Rows: | Export: | Wrap Cell Conte

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

TOTAL SALES

```
SELECT SUM(unit_Price * transaction_qty) AS Total_sales

FROM

shop_sales;
```

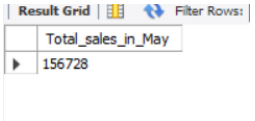


Result Grid | Filt

Total_sales
698812.3299999288

TOTAL SALES IN MAY MONTH

```
SELECT ROUND(SUM(unit_Price * transaction_qty)) AS Total_sales_in_May

FROM

shop_sales WHERE MONTH(transaction_date)= 5 ;
```

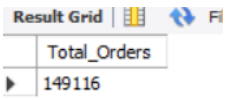


Result Grid | Filter Rows:

Total_sales_in_May
156728

TOTAL ORDERS

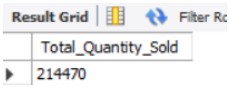
```
SELECT COUNT(transaction_id) as Total_Orders

FROM

shop_sales;
```



Total_Orders
149116

TOTAL QUANTITY SOLD

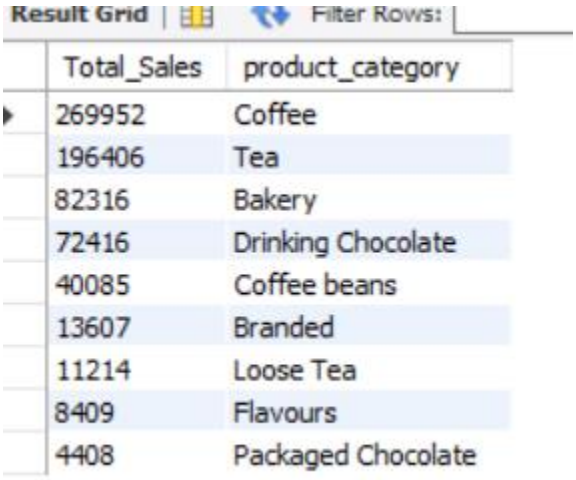
```
SELECT SUM(transaction_qty) as Total_Quantity_Sold

FROM

shop_sales;
```



Total_Quantity_Sold
214470

TOTAL SALES BY PRODUCT CATEGORY

```
SELECT ROUND(SUM(unit_price * transaction_qty)) as Total_Sales ,  product_category

FROM shop_sales

GROUP BY product_category

ORDER BY Total_Sales DESC;
```



Total_Sales	product_category
269952	Coffee
196406	Tea
82316	Bakery
72416	Drinking Chocolate
40085	Coffee beans
13607	Branded
11214	Loose Tea
8409	Flavours
4408	Packaged Chocolate

TOTAL SALES BY PRODUCT TYPE (TOP 10)

```
SELECT ROUND(SUM(unit_price * transaction_qty)) as Total_Sales,product_type
FROM shop_sales
GROUP BY product_type
ORDER BY Total_Sales DESC
LIMIT 10;
```

Result Grid | Filter Rows:

Total_Sales	product_type
91406	Barista Espresso
77082	Brewed Chai tea
72416	Hot chocolate
70035	Gourmet brewed coffee
47932	Brewed Black tea
47540	Brewed herbal tea
38781	Premium brewed coffee
37747	Organic brewed coffee
36866	Scone
31984	Drip coffee

TOTAL SALES BY WEEKDAY / WEEKEND

```
SELECT
CASE
   WHEN DAYOFWEEK(transaction_date) IN (1,7) THEN 'weekend'
   ELSE
   'weekday'
END AS Day_Type,
ROUND(SUM(unit_price * transaction_qty)) as Total_Sales
FROM
shop_sales
GROUP BY Day_Type;
```

Result Grid | Filter Row

Day_Type	Total_Sales
weekend	195225
weekday	503588

TOTAL SALES BY STORE LOCATION

```
SELECT  store_location,ROUND(SUM(transaction_qty * unit_price)) as Total_Sales
FROM
shop_sales
GROUP BY store_location
ORDER BY Total_Sales DESC;
```

esult Grid | Filter Rows:

store_location	Total_Sales
Hell's Kitchen	236511
Astoria	232244
Lower Manhattan	230057

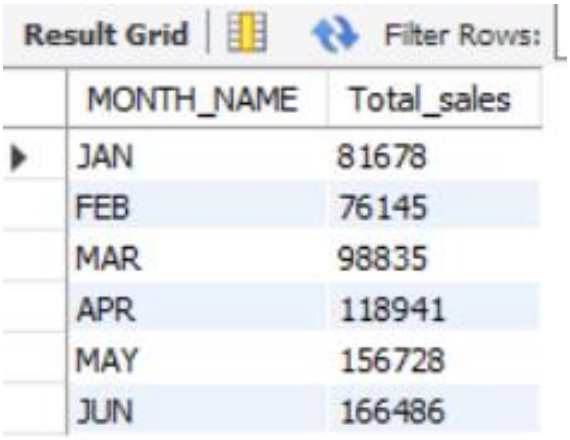
TOTAL SALES BY HOURS

```
SELECT HOUR(transaction_time) as Sale_Hour,
ROUND(SUM(transaction_qty * unit_price)) as Total_Sales
FROM
shop_sales
GROUP BY Sale_Hour
ORDER BY Total_Sales DESC;
```

Sale_Hour	Total_Sales
10	88673
9	85170
8	82700
7	63526
11	46319
15	41733
14	41305
16	41123
13	40367
12	40193
17	40134
18	34286
19	28447
6	21900
20	2936

MONTHLY TOTAL SALES

```
SELECT
CASE
        WHEN MONTH(transaction_date) = 1 THEN 'JAN'
        WHEN MONTH(transaction_date) = 2 THEN 'FEB'
        WHEN MONTH(transaction_date) = 3 THEN 'MAR'
        WHEN MONTH(transaction_date) = 4 THEN 'APR'
        WHEN MONTH(transaction_date) = 5 THEN 'MAY'
        WHEN MONTH(transaction_date) = 6 THEN 'JUN'
END AS MONTH_NAME,
ROUND(SUM(transaction_qty * unit_price))  as Total_sales
FROM shop_sales
GROUP BY MONTH_NAME;
```

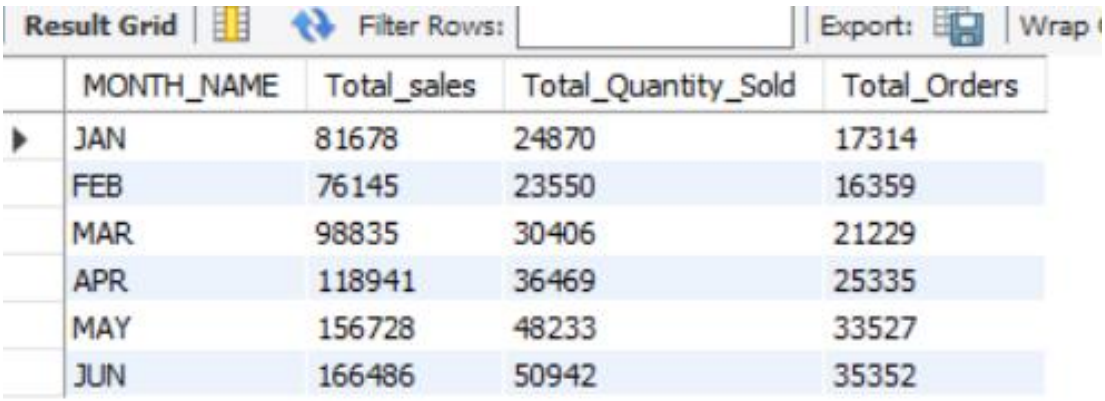


Result Grid | Filter Rows:

MONTH_NAME	Total_sales
JAN	81678
FEB	76145
MAR	98835
APR	118941
MAY	156728
JUN	166486

MONTHLY TOTAL SALES,TOTAL QUANTITY SOLD AND TOTAL ORDERS

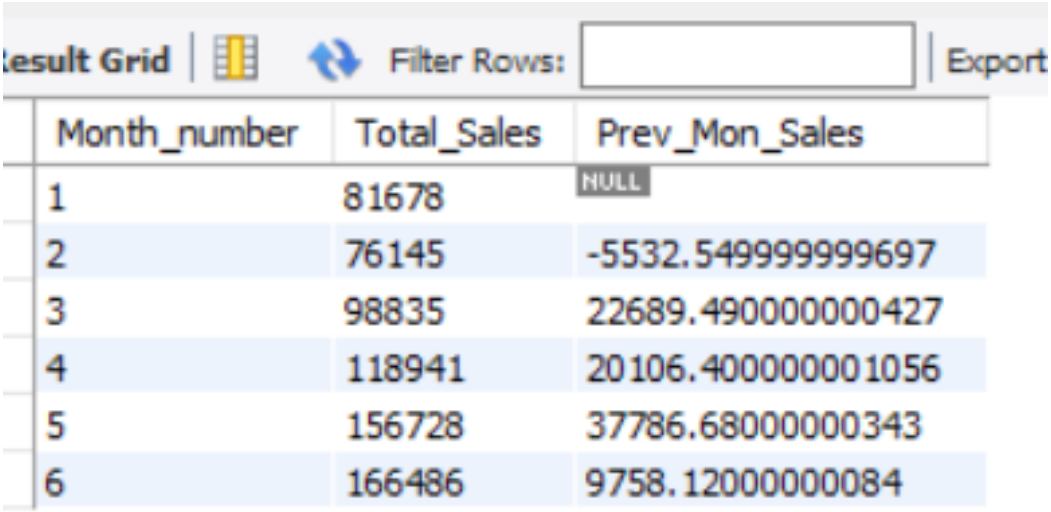
```
SELECT
CASE
        WHEN MONTH(transaction_date) = 1 THEN 'JAN'
        WHEN MONTH(transaction_date) = 2 THEN 'FEB'
        WHEN MONTH(transaction_date) = 3 THEN 'MAR'
        WHEN MONTH(transaction_date) = 4 THEN 'APR'
        WHEN MONTH(transaction_date) = 5 THEN 'MAY'
        WHEN MONTH(transaction_date) = 6 THEN 'JUN'
END AS MONTH_NAME,
ROUND(SUM(transaction_qty * unit_price))  AS Total_sales,
ROUND(SUM(transaction_qty))AS Total_Quantity_Sold,
COUNT(transaction_id) AS Total_Orders
FROM shop_sales
GROUP BY MONTH_NAME;
```



MONTH_NAME	Total_sales	Total_Quantity_Sold	Total_Orders
JAN	81678	24870	17314
FEB	76145	23550	16359
MAR	98835	30406	21229
APR	118941	36469	25335
MAY	156728	48233	33527
JUN	166486	50942	35352

MONTH TO MONTH SALES INCREASE

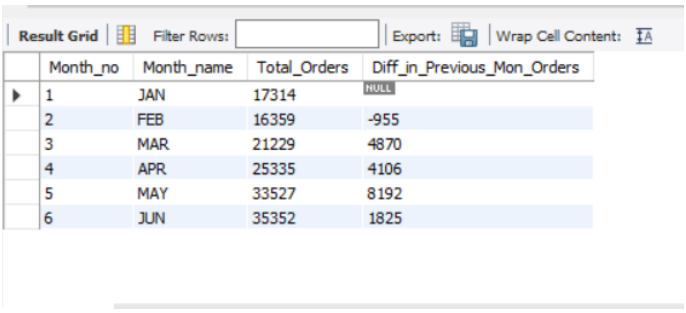
```
SELECT
MONTH(transaction_date) as Month_number,
ROUND(SUM(transaction_qty * unit_price)) AS Total_Sales,
(SUM(transaction_qty * unit_price) - LAG(SUM(transaction_qty * unit_price),1)
OVER (ORDER BY MONTH(transaction_date))) AS Prev_Mon_Sales
FROM
shop_sales
GROUP BY Month_number
ORDER BY Month_number;
```



Month_number	Total_Sales	Prev_Mon_Sales
1	81678	NULL
2	76145	-5532.549999999697
3	98835	22689.490000000427
4	118941	20106.400000001056
5	156728	37786.68000000343
6	166486	9758.12000000084

MONTH TO MONTH ORDERS AND DIFFERENCES

```
WITH Monthly_Orders AS
 (SELECT MONTH(transaction_date) AS Month_no,
CASE
   WHEN MONTH(transaction_date) = 1 THEN 'JAN'
   WHEN MONTH(transaction_date) = 2 THEN 'FEB'
   WHEN MONTH(transaction_date) = 3 THEN 'MAR'
   WHEN MONTH(transaction_date) = 4 THEN 'APR'
   WHEN MONTH(transaction_date) = 5 THEN 'MAY'
   WHEN MONTH(transaction_date) = 6 THEN 'JUN'
END AS Month_name,
COUNT(transaction_id) as Total_Orders
FROM
shop_sales
GROUP BY Month_no , Month_name
)
SELECT Month_no,
Month_name,
Total_Orders,
Total_Orders-LAG(Total_Orders,1) over (ORDER BY Month_no) AS Diff_in_Previous_Mon_Orders
FROM
Monthly_Orders;
```

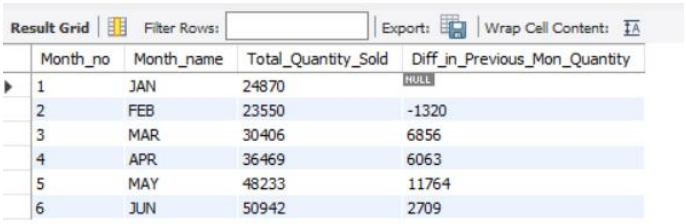


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Month_no	Month_name	Total_Orders	Diff_in_Previous_Mon_Orders
1	JAN	17314	NULL
2	FEB	16359	-955
3	MAR	21229	4870
4	APR	25335	4106
5	MAY	33527	8192
6	JUN	35352	1825

MONTH TO MONTH QUANTITY AND DIFFERENCES

```
WITH Monthly_Quantity AS
 (SELECT MONTH(transaction_date) AS Month_no,
CASE
   WHEN MONTH(transaction_date) = 1 THEN 'JAN'
   WHEN MONTH(transaction_date) = 2 THEN 'FEB'
   WHEN MONTH(transaction_date) = 3 THEN 'MAR'
   WHEN MONTH(transaction_date) = 4 THEN 'APR'
   WHEN MONTH(transaction_date) = 5 THEN 'MAY'
   WHEN MONTH(transaction_date) = 6 THEN 'JUN'
END AS Month_name,
SUM(transaction_qty) as Total_Quantity_Sold
FROM
shop_sales
GROUP BY Month_no,  Month_name
)
SELECT Month_no,
Month_name,
Total_Quantity_Sold,
Total_Quantity_Sold-LAG(Total_Quantity_Sold,1) over (ORDER BY Month_no) AS
Diff_in_Previous_Mon_Quantity
FROM
Monthly_Quantity;
```

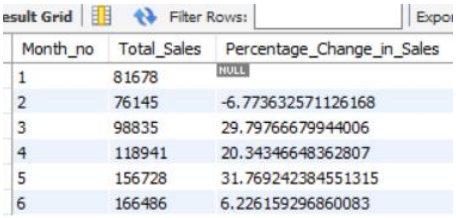


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Month_no	Month_name	Total_Quantity_Sold	Diff_in_Previous_Mon_Quantity
1	JAN	24870	NULL
2	FEB	23550	-1320
3	MAR	30406	6856
4	APR	36469	6063
5	MAY	48233	11764
6	JUN	50942	2709

MONTH TO MONTH % CHANGE IN SALES

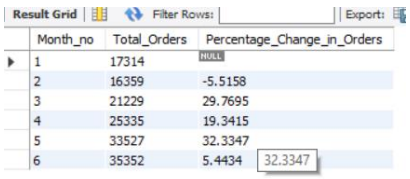
```
SELECT MONTH(transaction_date) as Month_no,

ROUND(SUM(transaction_qty * unit_price)) as Total_Sales,

(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 AS Percentage_Change_in_Sales

 FROM

 shop_sales

 GROUP BY Month_no

 ORDER BY Month_no;
```



Month_no	Total_Sales	Percentage_Change_in_Sales
1	81678	NULL
2	76145	-6.773632571126168
3	98835	29.79766679944006
4	118941	20.34346648362807
5	156728	31.769242384551315
6	166486	6.226159296860083

MONTH TO MONTH % CHANGE IN TOTAL ORDERS

```
SELECT MONTH(transaction_date) AS Month_no,

 COUNT(transaction_id) AS Total_Orders,

 (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 AS Percentage_Change_in_Orders

 FROM

 shop_sales

 GROUP BY MONTH(transaction_date)

 ORDER BY MONTH(transaction_date) ;
```



Month_no	Total_Orders	Percentage_Change_in_Orders
1	17314	NULL
2	16359	-5.5158
3	21229	29.7695
4	25335	19.3415
5	33527	32.3347
6	35352	5.4434

MONTH TO MONTH % CHANGE IN TOTAL QUANTITY SALES

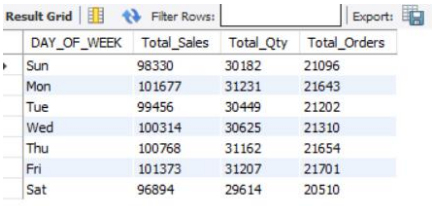
```
SELECT MONTH(transaction_date),
 SUM(transaction_qty) AS Total_Quantity,
 (SUM(transaction_qty) - LAG(SUM(transaction_qty),1) OVER (ORDER BY MONTH(transaction_date)))
 / LAG(SUM(transaction_qty),1) OVER (ORDER BY MONTH(transaction_date)) * 100 AS
Percentage_Change_In_Qty
 FROM
 shop_sales
 GROUP BY MONTH(transaction_date)
 ORDER BY MONTH(transaction_date)
 ;
```

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

MONTH(transaction_date)	Total_Quantity	Percentage_Change_In_Qty
1	24870	NULL
2	23550	-5.3076
3	30406	29.1125
4	36469	19.9401
5	48233	32.2575
6	50942	5.6165

TOTAL SALES , QUANTITY SOLD AND TOTAL ORDERS BY DAY OF WEEK

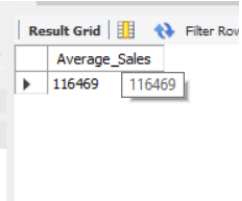
```
SELECT
 CASE
    WHEN DAYOFWEEK(transaction_date) = 2 THEN 'Mon'
    WHEN DAYOFWEEK(transaction_date) = 3 THEN 'Tue'
    WHEN DAYOFWEEK(transaction_date) = 4 THEN 'Wed'
    WHEN DAYOFWEEK(transaction_date) = 5 THEN 'Thu'
    WHEN DAYOFWEEK(transaction_date) = 6 THEN 'Fri'
    WHEN DAYOFWEEK(transaction_date) = 7 THEN 'Sat'
    WHEN DAYOFWEEK(transaction_date)= 1 THEN 'Sun'
END AS DAY_OF_WEEK,
ROUND(SUM(transaction_qty * unit_price)) AS Total_Sales,
SUM(transaction_qty) AS Total_Qty,
COUNT(transaction_id) AS Total_Orders
FROM shop_sales
GROUP BY
CASE
    WHEN DAYOFWEEK(transaction_date) = 2 THEN 'Mon'
    WHEN DAYOFWEEK(transaction_date) = 3 THEN 'Tue'
    WHEN DAYOFWEEK(transaction_date) = 4 THEN 'Wed'
    WHEN DAYOFWEEK(transaction_date) = 5 THEN 'Thu'
    WHEN DAYOFWEEK(transaction_date) = 6 THEN 'Fri'
    WHEN DAYOFWEEK(transaction_date) = 7 THEN 'Sat'
    WHEN DAYOFWEEK(transaction_date)= 1 THEN 'Sun'
END;
```



DAY_OF_WEEK	Total_Sales	Total_Qty	Total_Orders
Sun	98330	30182	21096
Mon	101677	31231	21643
Tue	99456	30449	21202
Wed	100314	30625	21310
Thu	100768	31162	21654
Fri	101373	31207	21701
Sat	96894	29614	20510

TOTAL AVG SALES

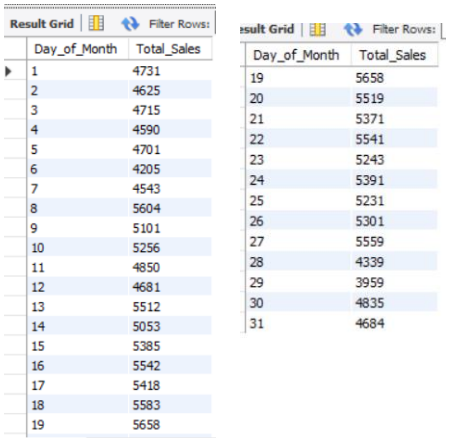
```
SELECT ROUND(AVG(Total_Sales)) AS Average_Sales FROM

(SELECT  MONTH(transaction_date),

SUM(transaction_qty * unit_price) AS Total_sales

FROM

shop_sales

GROUP BY MONTH(transaction_date)

ORDER BY MONTH(transaction_date)) AS Sales_by_month;
```



Average_Sales
116469

DAILY SALES FOR MONTH SELECTED

```
SELECT DAY(transaction_date) AS Day_of_Month,

ROUND(SUM(transaction_qty * unit_price)) AS Total_Sales

FROM

shop_sales

WHERE

MONTH(transaction_date) = 5   --- MAY Month Sales

GROUP BY Day_of_Month

ORDER BY Day_of_Month;
```



Day_of_Month	Total_Sales
1	4731
2	4625
3	4715
4	4590
5	4701
6	4205
7	4543
8	5604
9	5101
10	5256
11	4850
12	4681
13	5512
14	5053
15	5385
16	5542
17	5418
18	5583
19	5658

Day_of_Month	Total_Sales
19	5658
20	5519
21	5371
22	5541
23	5243
24	5391
25	5231
26	5301
27	5559
28	4339
29	3959
30	4835
31	4684

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 Avg'

WHEN    Total_Sales < Avg_Sales THEN 'Below Avg'

ELSE 'Equal to Avg'

END AS Sales_Status,

Total_Sales FROM(

SELECT DAY(transaction_date) AS Day_of_Month,

SUM(transaction_qty * unit_price)  AS Total_Sales,

AVG(SUM(transaction_qty * unit_price))       OVER() AS Avg_Sales

FROM

shop_sales

WHERE MONTH(transaction_date) = 5

GROUP BY DAY(transaction_date)) AS Sales_Data

ORDER BY Day_of_Month;
```

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

TO GET SALES FOR ALL HOURS FOR MONTH OF MAY

```
SELECT
HOUR(transaction_time) AS Hour_of_Day,
ROUND(SUM(unit_price * transaction_qty)) AS Total_Sales
FROM
shop_sales
WHERE MONTH(transaction_date) = 5
GROUP BY HOUR(transaction_time)
ORDER BY HOUR(transaction_time)  ;
```

Result Grid | Filter Rows:

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