Data Preparation and MySQL Queries

Project: Coffee Shop Sales Analysis

Dataset Information:

- Total Rows: 1,49,116
- Total Columns: 11
- Time Period Covered**:** January 2023 to June 2023 (6 months)

Tools Used:

- MySQL Workbench
- Excel
- Power BI

1. Excel to CSV Conversion
2. Import CSV into MySQL Workbench

```
CREATE DATABASE coffee_shop_db;
```

3. View & Explore the Table(coffee_sales)

```
-- View entire table
SELECT * FROM coffee_sales;

-- View the table structure
DESCRIBE coffee_sales;
```

4. Fixing Column Data Types

Convert transaction_date from text to DATE:

```
UPDATE coffee_sales
SET transaction_date = STR_TO_DATE(transaction_date, '%d-%m-%Y');

ALTER TABLE coffee_sales
MODIFY COLUMN transaction_date DATE;
```

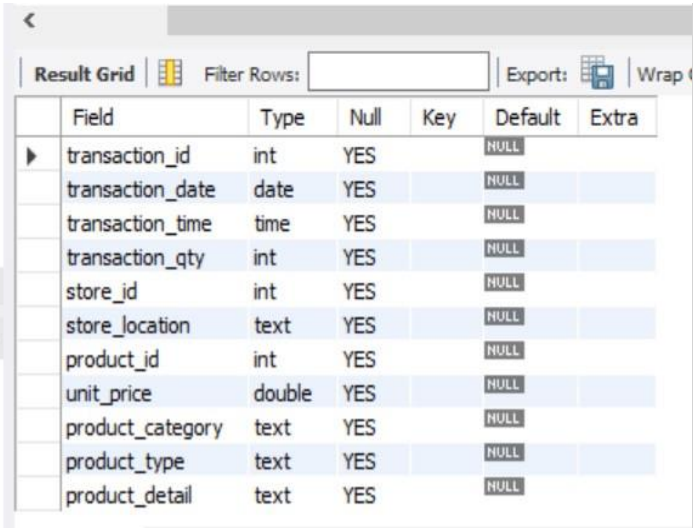
Convert transaction_time to TIME format:

```
UPDATE coffee_sales
SET transaction_time = STR_TO_DATE(transaction_time, '%H:%i:%s');
```

```
ALTER TABLE coffee_sales
MODIFY COLUMN transaction_time TIME;
```

5. Rename Incorrect Column Names

```
ALTER TABLE coffee_sales

CHANGE COLUMN i>>¿transaction_id  transaction_id INT;


DESCRIBE coffee_sales;
```



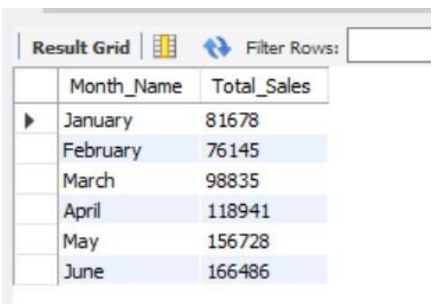
Result Grid | Filter Rows: | Export: | Wrap

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	

KPI REQUIREMENTS

- **TOTAL SALES OF EACH MONTH**

```
SELECT
  MONTHNAME(transaction_date) AS Month_Name,
  ROUND(SUM(unit_price * transaction_qty)) AS Total_Sales
FROM
  Coffee_sales
GROUP BY
  MONTHNAME(transaction_date)
ORDER BY
  STR_TO_DATE(Month_Name, '%M');
```

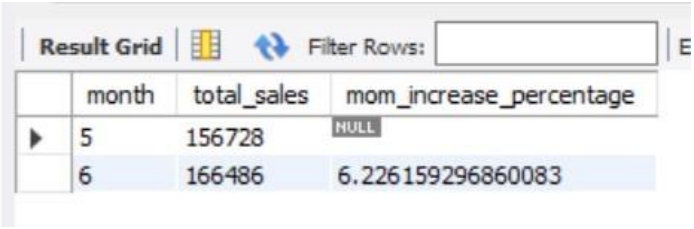


Result Grid | Filter Rows:

Month_Name	Total_Sales
January	81678
February	76145
March	98835
April	118941
May	156728
June	166486

➢ M-O-M DIFFERENCE AND M-O-M GROWTH (SALES)

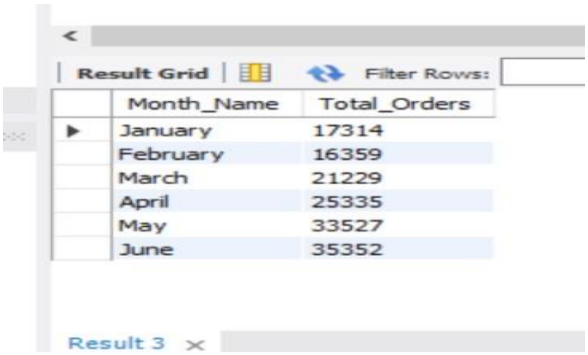
```
 SELECT
   MONTH(transaction_date) AS month,
   ROUND(SUM(unit_price * transaction_qty)) AS total_sales,
   (SUM(unit_price * transaction_qty) - LAG(SUM(unit_price * transaction_qty), 1)
   OVER (ORDER BY MONTH(transaction_date))) / LAG(SUM(unit_price *
transaction_qty), 1)
   OVER (ORDER BY MONTH(transaction_date)) * 100 AS mom_increase_percentage
FROM
coffee_sales
WHERE
   MONTH(transaction_date) IN (5, 6) -- for months of May and June
GROUP BY
   MONTH(transaction_date)
ORDER BY
   MONTH(transaction_date);
```



month	total_sales	mom_increase_percentage
5	156728	NULL
6	166486	6.226159296860083

➢ TOTAL ORDERS OF EACH MONTH

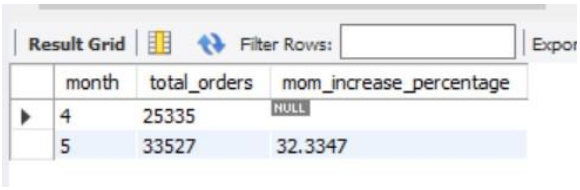
```
SELECT
  MONTHNAME(transaction_date) AS Month_Name,
  COUNT(transaction_id) as Total_Orders
FROM coffee_sales
GROUP BY
  MONTHNAME(transaction_date)
ORDER BY
  STR_TO_DATE(Month_Name, '%M');
```



Month_Name	Total_Orders
January	17314
February	16359
March	21229
April	25335
May	33527
June	35352

➢ MOM DIFFERENCE AND MOM GROWTH (ORDERS)

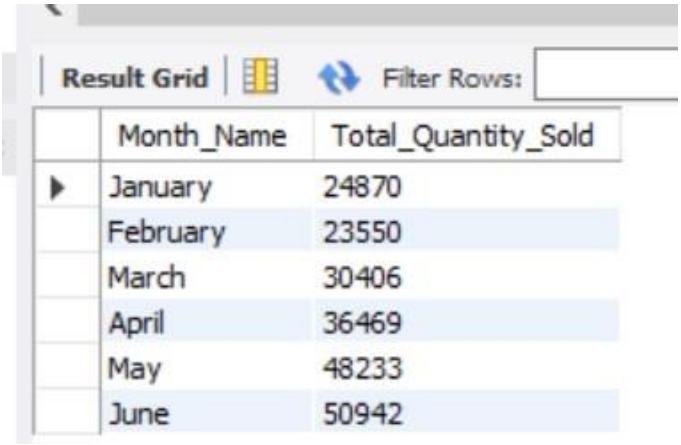
```
SELECT
   MONTH(transaction_date) AS month,
   COUNT(transaction_id) AS total_orders,
   (COUNT(transaction_id) - LAG(COUNT(transaction_id), 1)
   OVER (ORDER BY MONTH(transaction_date))) / LAG(COUNT(transaction_id), 1)
   OVER (ORDER BY MONTH(transaction_date)) * 100 AS mom_increase_percentage
FROM
   coffee_sales
WHERE
   MONTH(transaction_date) IN (5, 6) -- for May and June
GROUP BY
   MONTH(transaction_date)
ORDER BY
   MONTH(transaction_date);
```



month	total_orders	mom_increase_percentage
4	25335	NULL
5	33527	32.3347

➢ TOTAL QUANTITY SOLD EACH MONTH

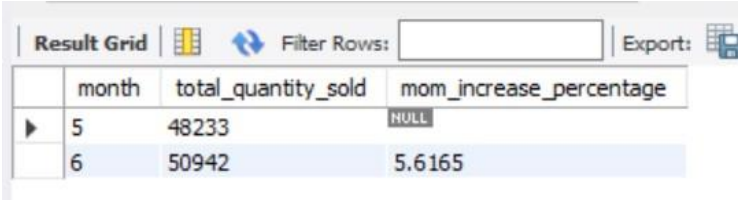
```
SELECT
MONTHNAME(transaction_date) AS Month_Name,
SUM(transaction_qty) as Total_Quantity_Sold
FROM coffee_sales
GROUP BY
  MONTHNAME(transaction_date)
ORDER BY
  STR_TO_DATE(Month_Name, '%M');
```



Month_Name	Total_Quantity_Sold
January	24870
February	23550
March	30406
April	36469
May	48233
June	50942

➢ MOM DIFFERENCE AND MOM GROWTH (QUANTITY SOLD)

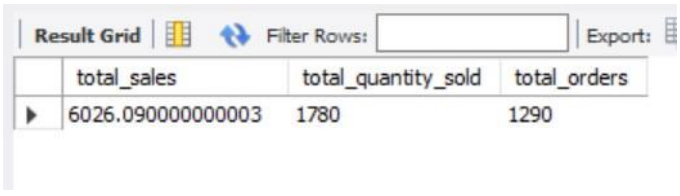
```
SELECT
   MONTH(transaction_date) AS month,
   ROUND(SUM(transaction_qty)) AS total_quantity_sold,
   (SUM(transaction_qty) - LAG(SUM(transaction_qty), 1)
   OVER (ORDER BY MONTH(transaction_date))) / LAG(SUM(transaction_qty), 1)
   OVER (ORDER BY MONTH(transaction_date)) * 100 AS mom_increase_percentage
FROM
   coffee_sales
WHERE
   MONTH(transaction_date) IN (5, 6)   -- for May and June
GROUP BY
   MONTH(transaction_date)
ORDER BY
   MONTH(transaction_date);
```



month	total_quantity_sold	mom_increase_percentage
5	48233	NULL
6	50942	5.6165

➢ CALENDAR TABLE – DAILY SALES, QUANTITY and TOTAL ORDERS

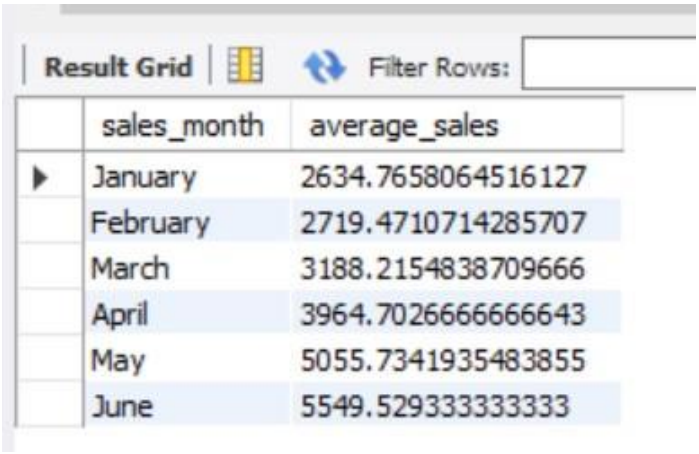
```
SELECT
   SUM(unit_price * transaction_qty) AS total_sales,
   SUM(transaction_qty) AS total_quantity_sold,
   COUNT(transaction_id) AS total_orders
FROM
   coffee_sales
WHERE
   transaction_date = '2023-06-18'; --For 18 June 2023
```



total_sales	total_quantity_sold	total_orders
6026.090000000003	1780	1290

➢ SALES TREND OVER PERIOD

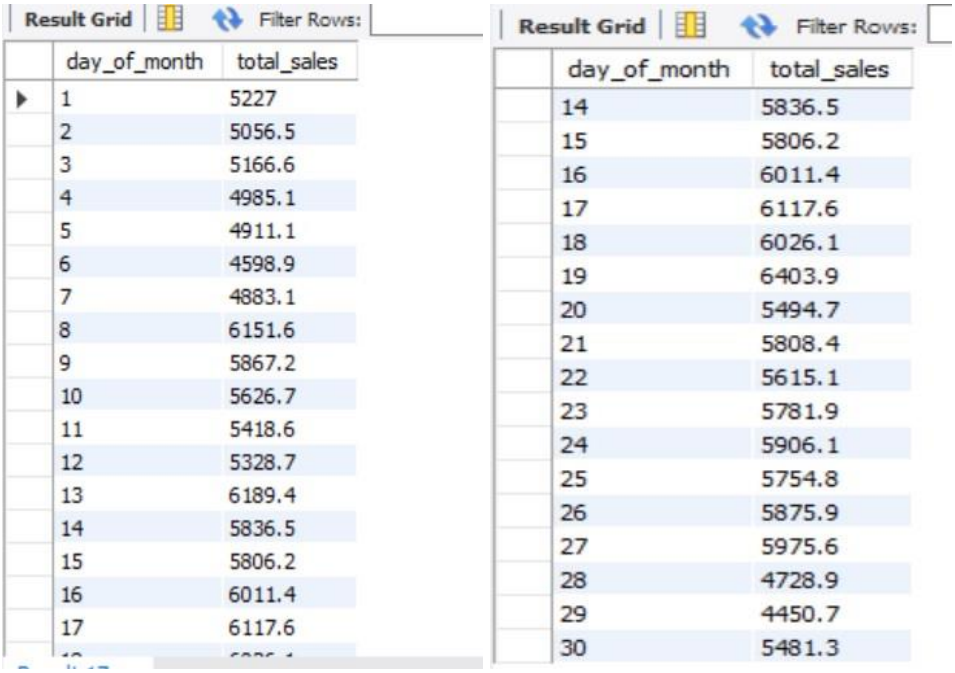
```
SELECT
   MONTHNAME(transaction_date) AS sales_month,
   AVG(daily_sales) AS average_sales
FROM
 (
   SELECT
      transaction_date,
      SUM(unit_price * transaction_qty) AS daily_sales
   FROM
      coffee_sales
   GROUP BY
      transaction_date
)
AS daily_data
GROUP BY
   MONTHNAME(transaction_date)
ORDER BY
   STR_TO_DATE(sales_month,'%M');
```



	sales_month	average_sales
▶	January	2634.7658064516127
	February	2719.4710714285707
	March	3188.2154838709666
	April	3964.7026666666643
	May	5055.7341935483855
	June	5549.529333333333

➢ DAILY SALES FOR MONTH SELECTED

```
SELECT
   DAY(transaction_date) AS day_of_month,
   ROUND(SUM(unit_price * transaction_qty),1) AS total_sales
FROM
   coffee_sales
WHERE
   MONTH(transaction_date) = 6  -- Filter for June
GROUP BY
   DAY(transaction_date)
ORDER BY
   DAY(transaction_date);
```



Result Grid | Filter Rows:

day_of_month	total_sales
1	5227
2	5056.5
3	5166.6
4	4985.1
5	4911.1
6	4598.9
7	4883.1
8	6151.6
9	5867.2
10	5626.7
11	5418.6
12	5328.7
13	6189.4
14	5836.5
15	5806.2
16	6011.4
17	6117.6

Result Grid | Filter Rows:

day_of_month	total_sales
14	5836.5
15	5806.2
16	6011.4
17	6117.6
18	6026.1
19	6403.9
20	5494.7
21	5808.4
22	5615.1
23	5781.9
24	5906.1
25	5754.8
26	5875.9
27	5975.6
28	4728.9
29	4450.7
30	5481.3

- **COMPARING DAILY SALES WITH AVERAGE SALES**

```
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_sales
```

```
    WHERE
        MONTH(transaction_date) = 6 -- Filter for June
    GROUP BY
        DAY(transaction_date)
) AS sales_data
ORDER BY
    day_of_month;
```

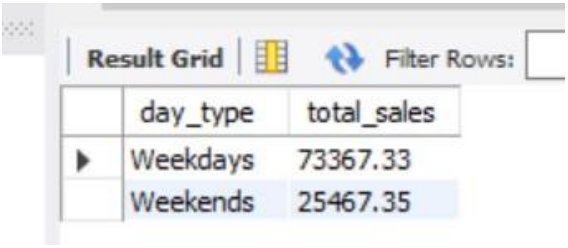
day_of_month	sales_status	total_sales
1	Below Average	5227.000000000001
2	Below Average	5056.499999999996
3	Below Average	5166.649999999994
4	Below Average	4985.149999999999
5	Below Average	4911.149999999995
6	Below Average	4598.899999999996
7	Below Average	4883.099999999998
8	Above Average	6151.5899999999965
9	Above Average	5867.159999999994
10	Above Average	5626.749999999998
11	Below Average	5418.609999999998
12	Below Average	5328.699999999996
13	Above Average	6189.360000000001
14	Above Average	5836.519999999998
15	Above Average	5806.240000000002
16	Above Average	6011.430000000001
17	Above Average	6117.599999999999

Result Grid | Filter Rows: | Export:

day_of_month	sales_status	total_sales
14	Above Average	5836.519999999998
15	Above Average	5806.240000000002
16	Above Average	6011.430000000001
17	Above Average	6117.599999999999
18	Above Average	6026.090000000003
19	Above Average	6403.910000000004
20	Below Average	5494.659999999999
21	Above Average	5808.380000000005
22	Above Average	5615.099999999998
23	Above Average	5781.859999999997
24	Above Average	5906.1
25	Above Average	5754.849999999996
26	Above Average	5875.9
27	Above Average	5975.65
28	Below Average	4728.899999999999
29	Below Average	4450.749999999998
30	Below Average	5481.32

➢ SALES BY WEEKDAY / WEEKEND

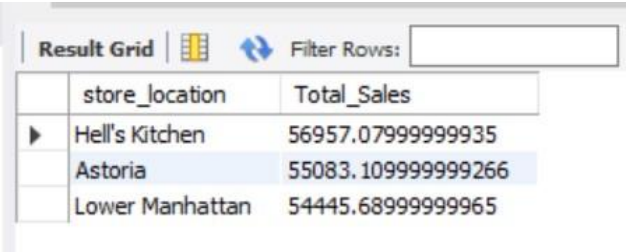
```
SELECT
  CASE
    WHEN DAYOFWEEK(transaction_date) IN (1, 7) THEN 'Weekends'
    ELSE 'Weekdays'
  END AS day_type,
  ROUND(SUM(unit_price * transaction_qty),2) AS total_sales
FROM
  coffee_sales
WHERE
  MONTH(transaction_date) = 3  -- March
GROUP BY
  CASE
    WHEN DAYOFWEEK(transaction_date) IN (1, 7) THEN 'Weekends'
    ELSE 'Weekdays'
  END;
```



day_type	total_sales
Weekdays	73367.33
Weekends	25467.35

➢ SALES BY STORE LOCATION

```
SELECT
  store_location,
  SUM(unit_price * transaction_qty) as Total_Sales
FROM coffee_sales
WHERE
  MONTH(transaction_date) =6 -- June
GROUP BY
  store_location
ORDER BY
   SUM(unit_price * transaction_qty) DESC
```

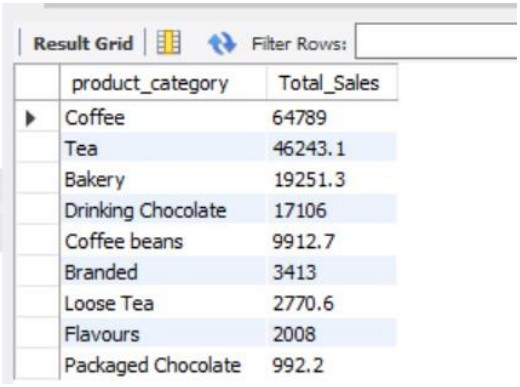


Result Grid | Filter Rows:

store_location	Total_Sales
Hell's Kitchen	56957.07999999935
Astoria	55083.109999999266
Lower Manhattan	54445.68999999965

➢ SALES BY PRODUCT CATEGORY

```
SELECT
  product_category,
  ROUND(SUM(unit_price * transaction_qty),1) as Total_Sales
FROM coffee_sales
WHERE
  MONTH(transaction_date) = 6 --June
GROUP BY product_category
ORDER BY SUM(unit_price * transaction_qty) DESC
```

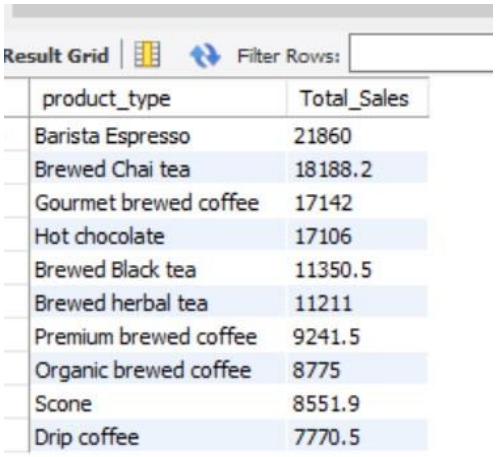


Result Grid | Filter Rows:

product_category	Total_Sales
Coffee	64789
Tea	46243.1
Bakery	19251.3
Drinking Chocolate	17106
Coffee beans	9912.7
Branded	3413
Loose Tea	2770.6
Flavours	2008
Packaged Chocolate	992.2

➢ SALES BY PRODUCTS (TOP 10)

```
SELECT
    product_type,
    ROUND(SUM(unit_price * transaction_qty),1) as Total_Sales
FROM
   coffee_sales
WHERE
   MONTH(transaction_date) = 6
GROUP BY
    product_type
ORDER BY
     SUM(unit_price * transaction_qty) DESC
LIMIT 10
```

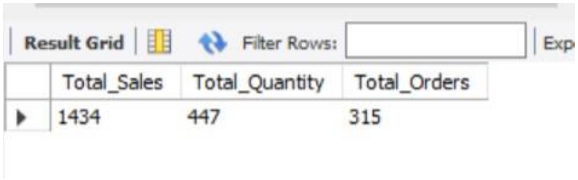


Result Grid | Filter Rows:

product_type	Total_Sales
Barista Espresso	21860
Brewed Chai tea	18188.2
Gourmet brewed coffee	17142
Hot chocolate	17106
Brewed Black tea	11350.5
Brewed herbal tea	11211
Premium brewed coffee	9241.5
Organic brewed coffee	8775
Scone	8551.9
Drip coffee	7770.5

- **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_sales
WHERE
   DAYOFWEEK(transaction_date) = 1 -- Filter for Sunday
   AND HOUR(transaction_time) = 11 -- Filter for hour number 11
   AND MONTH(transaction_date) = 6; -- Filter for June
```



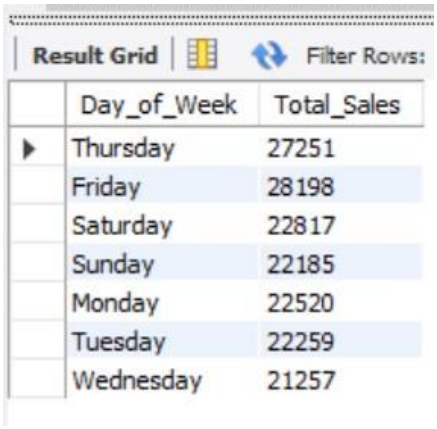
Result Grid | Filter Rows: | Exp

Total_Sales	Total_Quantity	Total_Orders
1434	447	315

- **TO GET SALES FROM MONDAY TO SUNDAY FOR MONTH OF JUNE**

```
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_sales
WHERE
    MONTH(transaction_date) = 6 -- June
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;
```

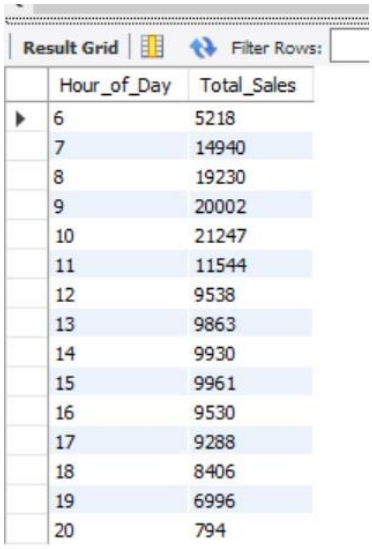


Day_of_Week	Total_Sales
Thursday	27251
Friday	28198
Saturday	22817
Sunday	22185
Monday	22520
Tuesday	22259
Wednesday	21257

- **TO GET SALES FOR ALL HOURS FOR MONTH OF JUNE**

```
SELECT
    HOUR(transaction_time) AS Hour_of_Day,
    ROUND(SUM(unit_price * transaction_qty)) AS Total_Sales
FROM
```

```
    coffee_sales
WHERE
    MONTH(transaction_date) = 6
GROUP BY
    HOUR(transaction_time)
ORDER BY
    HOUR(transaction_time);
```



Result Grid | Filter Rows:

Hour_of_Day	Total_Sales
6	5218
7	14940
8	19230
9	20002
10	21247
11	11544
12	9538
13	9863
14	9930
15	9961
16	9530
17	9288
18	8406
19	6996
20	794