COFFEE SHOP SALES ANALYSIS

```
/* CREATING DATABASE */

CREATE DATABASE IF NOT EXISTS coffee_sales;


/* USING THE DATABASE */

USE coffee_sales;


/* IMPORTING TABLE

We have imported the table from a csv file. Now, we have to updates the datatypes and constraints to the columns also the names of the columns in correct and understandable format.*/

UPDATE coffee_sales

SET transaction_date = STR_TO_DATE(transaction_date, '%m/%d/%Y');

/* Updating the dates in correct format*/


ALTER TABLE coffee_sales

MODIFY COLUMN transaction_date DATE;

/* Updating the datatype of the column consisting dates */


UPDATE coffee_sales

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

/* Updating the time in correct format */


ALTER TABLE coffee_sales

MODIFY COLUMN transaction_time TIME;
```

/* DESCRIBING THE COLUMNS OF THE TABLE, THEIR DATATYPE AND CONSTRAINTS */

```
DESCRIBE coffee_sales;
```

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	

/* TOTAL SALES AND SALES OF EACH MONTH*/

```
SELECT SUM(unit_price*transaction_qty) AS Total_Sales
FROM coffee_sales;
```

Total_Sales
698812.3299999288

```
SELECT MONTHNAME(transaction_date) AS Month,
    ROUND(SUM(unit_price*transaction_qty),2) AS Sales
FROM coffee_sales
GROUP BY MONTHNAME(transaction_date);
```

Month	Sales
January	81677.74
February	76145.19
March	98834.68
April	118941.08
May	156727.76
June	166485.88

/* MONTH-ON-MONTH INCREASE OR DECREASE PERCENTAGE IN SALES*/

```
SELECT MONTH(transaction_date) AS Month,
    CONCAT(ROUND((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)) * 100,2),"%") AS MOM_increase_percentage
FROM coffee_sales
GROUP BY MONTH(transaction_date)
ORDER BY MONTH(transaction_date);
```

Month	MOM_increase_percentage
1	NULL
2	-6.77%
3	29.8%
4	20.34%
5	31.77%
6	6.23%

/* TOTAL NUMBER OF ORDERS AND NUMBER OF ORDERS IN EACH MONTH*/

```
SELECT SUM(transaction_id) AS Total_Orders
FROM coffee_sales;  -- Total Number of Orders
```

Total_Orders
11144537944

```
SELECT MONTHNAME(transaction_date) AS Month,
    COUNT(transaction_id) AS Number_of_Orders
FROM coffee_sales
GROUP BY MONTHNAME(transaction_date); -- Number of Orders Each Month
```

Month	Number_of_Orders
January	17314
February	16359
March	21229
April	25335
May	33527
June	35352

/* MONTH-ON-MONTH INCREASE OR DECREASE PERCENTAGE IN NUMBER OF ORDERS */

```
SELECT MONTH(transaction_date) AS Month,
    CONCAT(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 coffee_sales
GROUP BY MONTH(transaction_date)
ORDER BY MONTH(transaction_date);
```

Month	MOM_increase_percentage
1	NULL
2	-5.52%
3	29.77%
4	19.34%
5	32.33%
6	5.44%

/* TOTAL QUANTITY SOLD AND QUANTITY SOLD IN EACH MONTH*/

```
SELECT SUM(transaction_qty) AS Total_Quantity_Sold
FROM coffee_sales;  -- Total Quantity Sold
```

Total_Quantity_Sold
214470

```
SELECT MONTHNAME(transaction_date) AS Month,
    SUM(transaction_qty) AS Quantity_Sold
FROM coffee_sales
GROUP BY MONTHNAME(transaction_date);
```

Month	Quantity_Sold
January	24870
February	23550
March	30406
April	36469
May	48233
June	50942

/* MONTH-ON-MONTH INCREASE OR DECREASE PERCENTAGE IN QUANTITY SOLD*/

```
SELECT MONTH(transaction_date) AS Month,
    CONCAT(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 coffee_sales
GROUP BY MONTH(transaction_date)
ORDER BY MONTH(transaction_date);
```

Month	MOM_increase_percentage
1	NULL
2	-5.31%
3	29.11%
4	19.94%
5	32.26%
6	5.62%

/* DAILY ANALYSIS OF A COMPLETE MONTH (let's say March) */

```
SELECT DAY(transaction_date) AS Day_of_month,
    ROUND(SUM(unit_price*transaction_qty),2) AS Sales,
    COUNT(transaction_id) AS No_of_Orders,
    SUM(transaction_qty) AS Quantity_Sold
FROM coffee_sales
WHERE MONTH(transaction_date)=3
GROUP BY DAY(transaction_date)
ORDER BY DAY(transaction_date); -- Here we can take any month from the dataset, here we have data of months from January to June
```

Day_of_month	Sales	No_of_Orders	Quantity_Sold
1	3040.25	661	968
2	2996.05	673	963
3	3155.15	710	1010
4	2781.9	624	897
5	2945.3	675	952
6	2618.05	587	836
7	2803.5	629	883
8	3523.26	742	1039
9	3459.97	722	983
10	3441.58	750	1051
11	3211.65	689	1002
12	3088.33	667	947
13	3627.65	730	1110

14	3312.66	691	918
15	3338.03	721	1000
16	3386.11	758	1037
17	3181.75	693	942
18	3408.36	731	1001
19	3340.03	725	988
20	3262.28	735	988
21	3209.8	650	949
22	3284.11	669	1014
23	3361.13	694	1040
24	3586.2	718	1111
25	3380.95	688	1050
26	3310.83	671	1038
27	3674.35	764	1177
28	2792.55	626	897
29	2492	582	812
30	2932.82	639	917
31	2888.08	615	886

```
/* COMPARING SALES OF EACH MONTH WITH THE AVERAGE SALE */
SELECT Month,
    CASE
      WHEN total_sales > avg_sales THEN 'Above Average'
      WHEN total_sales < avg_sales THEN 'Below Average'
      ELSE 'Average'
    END AS sales_status
FROM (SELECT MONTHNAME(transaction_date) AS Month,
        SUM(unit_price * transaction_qty) AS total_sales,
        AVG(SUM(unit_price * transaction_qty)) OVER() AS avg_sales
    FROM coffee_sales
    GROUP BY MONTHNAME(transaction_date)) AS sales_data;
```

Month	sales_status
January	Below Average
February	Below Average
March	Below Average
April	Above Average
May	Above Average
June	Above Average

```
/* COMPARING DAILY SALES OF A MONTH WITH THE AVERAGE SALE OF THAT MONTH (Here March is taken) */
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
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) = 3
    GROUP BY
        DAY(transaction_date)
) AS sales_data
ORDER BY
    day_of_month;
```

day_of_month	sales_status
1	Below Average
2	Below Average
3	Below Average
4	Below Average
5	Below Average
6	Below Average
7	Below Average
8	Above Average
9	Above Average
10	Above Average
11	Above Average
12	Below Average
13	Above Average
14	Above Average
15	Above Average
16	Above Average
17	Below Average
18	Above Average
19	Above Average
20	Above Average
21	Above Average
22	Above Average
23	Above Average
24	Above Average
25	Above Average
26	Above Average
27	Above Average
28	Below Average
29	Below Average
30	Below Average
31	Below Average

```
/* COMPARISON ANALYSIS ON WEEKDAYS AND WEEKENDS  */
SELECT CASE
        WHEN DAYOFWEEK(transaction_date) IN (1,7) THEN "Weekends"
        ELSE "Weekdays"
    END AS Day_type,
    ROUND(SUM(transaction_qty*unit_price),2) as Sales
FROM coffee_sales
GROUP BY CASE
        WHEN DAYOFWEEK(transaction_date) IN (1,7) THEN "Weekends"
        ELSE "Weekdays"
    END;
```

Day_type	Sales
Weekends	195224.79
Weekdays	503587.54

```
/* ANALYSIS OF ALL THE STORES OVER THE GIVEN TIME */
SELECT store_location,
    COUNT(transaction_id) AS No_of_Orders,
    SUM(transaction_qty) AS Quantity_Sold,
        ROUND(SUM(transaction_qty*unit_price),2) AS Sales
FROM coffee_sales
GROUP BY store_location
ORDER BY Sales DESC;
```

store_location	No_of_Orders	Quantity_Sold	Sales
Hell's Kitchen	50735	71737	236511.17
Astoria	50599	70991	232243.91
Lower Manhattan	47782	71742	230057.25

/* ANALYSIS OF ALL THE PRODUCT TYPES AND CATEGORY OVER THE GIVEN TIME */

```
SELECT product_category,
    COUNT(transaction_id) AS No_of_Orders,
    SUM(transaction_qty) AS Quantity_Sold,
        ROUND(SUM(transaction_qty*unit_price),2) AS Sales
FROM coffee_sales
GROUP BY product_category;
```

product_category	No_of_Orders	Quantity_Sold	Sales
Coffee	58416	89250	269952.45
Tea	45449	69737	196405.95
Drinking Chocolate	11468	17457	72416
Bakery	22796	23214	82315.64
Flavours	6790	10511	8408.8
Loose Tea	1210	1210	11213.6
Coffee beans	1753	1828	40085.25
Packaged Chocolate	487	487	4407.64
Branded	747	776	13607

```
SELECT product_type,
    COUNT(transaction_id) AS No_of_Orders,
    SUM(transaction_qty) AS Quantity_Sold,
        ROUND(SUM(transaction_qty*unit_price),2) AS Sales
FROM coffee_sales
GROUP BY product_type;
```

product_type	No_of_Orders	Quantity_Sold	Sales
Gourmet brewed coffee	16912	25973	70034.6
Brewed Chai tea	17183	26250	77081.95
Hot chocolate	11468	17457	72416
Drip coffee	8477	12891	31984
Scone	10173	10465	36866.12
Barista Espresso	16403	24943	91406.2
Brewed Black tea	11350	17462	47932
Brewed Green tea	5671	8697	23852.5
Brewed herbal tea	11245	17328	47539.5
Biscotti	5711	5788	19793.53
Pastry	6912	6961	25655.99
Organic brewed coffee	8489	13012	37746.5
Premium brewed coffee	8135	12431	38781.15

	Regular syrup	4979	7606	6084.8
	Herbal tea	305	305	2729.75
	Gourmet Beans	366	366	6798
	Organic Beans	415	420	8509.5
	Sugar free syrup	1811	2905	2324
	Drinking Chocolate	266	266	2728.04
	Premium Beans	336	406	14583.5
	Chai tea	443	443	4301.25
	Green beans	134	134	1340
	Espresso Beans	319	319	5560.25
	Green tea	159	159	1470.75
	Organic Chocolate	221	221	1679.6
	Housewares	526	555	7444
	Black tea	303	303	2711.85
	House blend Beans	183	183	3294
	Clothing	221	221	6163

/* TOP 5 PRODUCT TYPE BY SALES*/

```
SELECT product_type,
    ROUND(SUM(transaction_qty*unit_price),2) AS Sales
FROM coffee_sales
GROUP BY product_type
ORDER BY Sales DESC
LIMIT 5;
```

	product_type	Sales
▶	Barista Espresso	91406.2
	Brewed Chai tea	77081.95
	Hot chocolate	72416
	Gourmet brewed coffee	70034.6
	Brewed Black tea	47932

```
/* HOURLY BASED ANALYSIS */
SELECT HOUR(transaction_time) AS A,
    COUNT(transaction_id) AS No_of_Orders,
    SUM(transaction_qty) AS Quantity_Sold,
        ROUND(SUM(transaction_qty*unit_price),2) AS Sales
FROM coffee_sales
GROUP BY HOUR(transaction_time)
ORDER BY HOUR(transaction_time);
```

Hour_number	No_of_Orders	Quantity_Sold	Sales
6	4594	6865	21900.27
7	13428	19449	63526.47
8	17654	25197	82699.87
9	17764	25370	85169.53
10	18545	26713	88673.39
11	9766	14035	46319.14
12	8708	12690	40192.79
13	8714	12439	40367.45
14	8933	12907	41304.74
15	8979	12923	41733.1
16	9093	12881	41122.75
17	8745	12700	40134.31
18	7498	10826	34286.2
19	6092	8595	28446.68
20	603	880	2935.64

```
/* ANALYSIS OF ALL DAYS OF A 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,
  COUNT(transaction_id) AS No_of_Orders,
  SUM(transaction_qty) AS Quantity_Sold,
      ROUND(SUM(transaction_qty*unit_price),2) AS Sales
FROM coffee_sales
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	No_of_Orders	Quantity_Sold	Sales
Sunday	21096	30182	98330.31
Monday	21643	31231	101677.28
Tuesday	21202	30449	99455.94
Wednesday	21310	30625	100313.54
Thursday	21654	31162	100767.78
Friday	21701	31207	101373
Saturday	20510	29614	96894.48