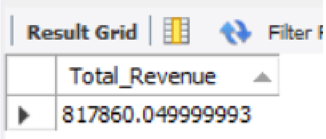
By Neetu Shah

PIZZA SALES SQL QUERIES (MYSQL WORKBENCH)

KPI's

1. **Total Revenue:**

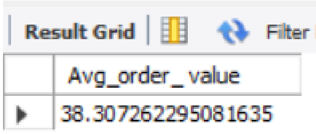
   select sum(total_price) as Total_Revenue from pizza_sales;



	Total_Revenue
▶	817860.049999993

2. **Average order value:**

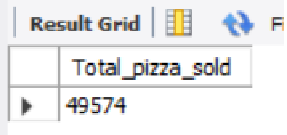
   select (sum(total_price)/ count(distinct order_id)) as 'Avg_order_ value' from pizza_sales;



	Avg_order_ value
▶	38.30726229508 1635

3. **Total Pizzas sold:**

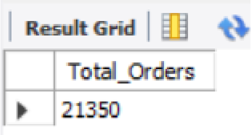
   select sum(quantity) as Total_pizza_sold from pizza_sales;



	Total_pizza_sold
▶	49574

4. **Total Orders:**

   select count(distinct order_id) as Total_Orders from pizza_sales;

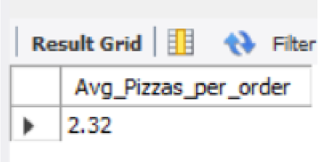


	Total_Orders
▶	21350

5. **Average Pizzas Per Order:**

   select cast(sum(quantity) / count(distinct order_id) as decimal(10,2)) as

   'Avg_Pizzas_per_order' from pizza_sales;



	Avg_Pizzas_per_order
▶	2.32

By Neetu Shah

B. Daily Trend for Total Orders

-- change date to default mysql yyyy-mm-dd format then convert text data type to date data type

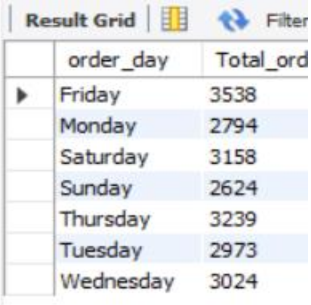
```
update pizza_sales

set order_date=str_to_date(order_date,"%d-%m-%Y");
```

-- here we are converting ddmmyy which is there in our csv data to default sql date yyyy-mm-dd format

```
alter table pizza_sales

modify order_date date; -- here we are changing to date type
```

```
select dayname(order_date) as order_day, count(distinct order_id) as Total_orders

from pizza_sales

group by dayname(order_date)
```

Output:

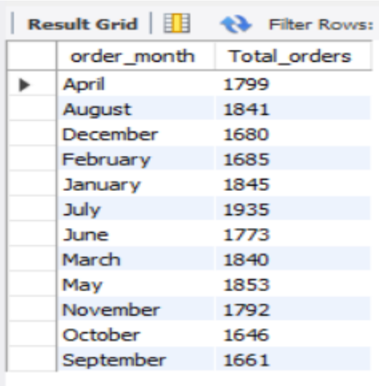


order_day	Total_ord
Friday	3538
Monday	2794
Saturday	3158
Sunday	2624
Thursday	3239
Tuesday	2973
Wednesday	3024

C. Monthly Trend for Total Orders

```
select monthname(order_date) as order_month, count(distinct order_id) as Total_orders from pizza_sales

group by monthname(order_date);
```

Output:

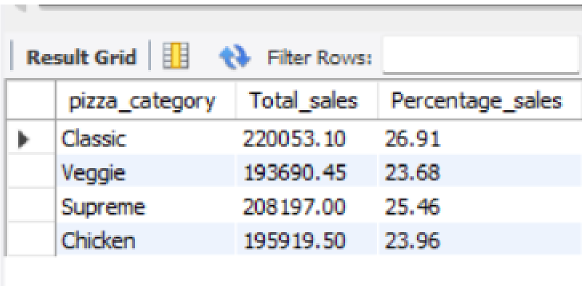


order_month	Total_orders
April	1799
August	1841
December	1680
February	1685
January	1845
July	1935
June	1773
March	1840
May	1853
November	1792
October	1646
September	1661

D. % of Sales by Pizza Category

```
select pizza_category, sum(total_price)*100/ (select sum(total_price) from pizza_sales where
month(order_date)=1) as Percentage_sales   --comment: here we are filtering for month January
 from pizza_sales
where month(order_date)=1
group by pizza_category;


-- same as above but removing where clause, keeping upto 2 decimal places and adding total price
column as 'total sales' alias
select pizza_category,cast(sum(total_price) as decimal(10,2)) as
Total_sales,cast(sum(total_price)*100/ (select sum(total_price) from pizza_sales) as decimal(10,2))
as Percentage_sales
from pizza_sales
group by pizza_category;
```

Output:

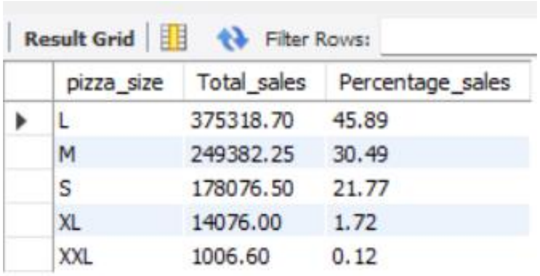


pizza_category	Total_sales	Percentage_sales
Classic	220053.10	26.91
Veggie	193690.45	23.68
Supreme	208197.00	25.46
Chicken	195919.50	23.96

E. % of Sales by Pizza Size

```
select pizza_size,cast(sum(total_price) as decimal(10,2)) as Total_sales,cast(sum(total_price)*100/
(select sum(total_price) from pizza_sales) as decimal(10,2)) as Percentage_sales
from pizza_sales
group by pizza_size
order by Percentage_sales desc;
```

Output:



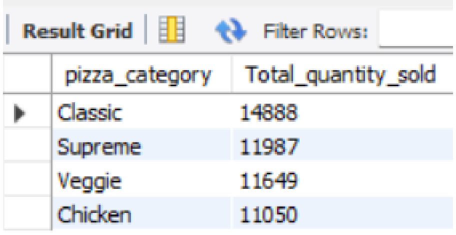
pizza_size	Total_sales	Percentage_sales
L	375318.70	45.89
M	249382.25	30.49
S	178076.50	21.77
XL	14076.00	1.72
XXL	1006.60	0.12

By Neetu Shah

F. Total Pizzas Sold by Pizza Category

```
select pizza_category, sum(quantity) as Total_quantity_sold

from pizza_sales

group by pizza_category

order by Total_quantity_sold desc;
```

Output:

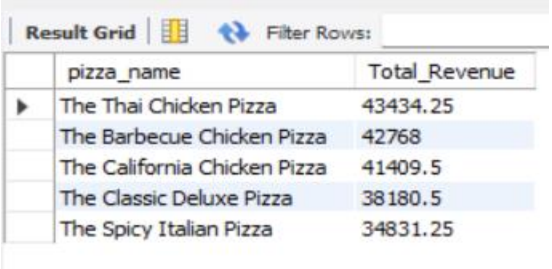


pizza_category	Total_quantity_sold
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

G. Top 5 Pizzas by Revenue

```
select pizza_name,sum(total_price) as Total_Revenue

from pizza_sales

group by pizza_name

order by Total_Revenue desc

limit 5;
```

Output:

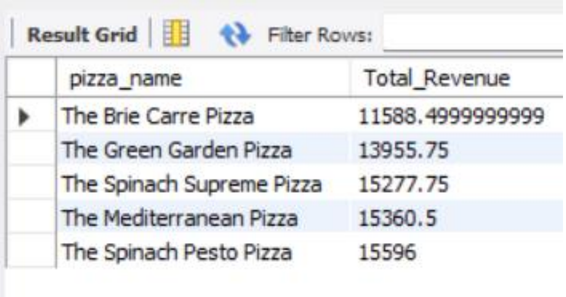


pizza_name	Total_Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Spicy Italian Pizza	34831.25

H. Bottom 5 Pizzas by Revenue

```
select pizza_name,sum(total_price) as Total_Revenue

from pizza_sales

group by pizza_name

order by Total_Revenue

limit 5;
```

Output:

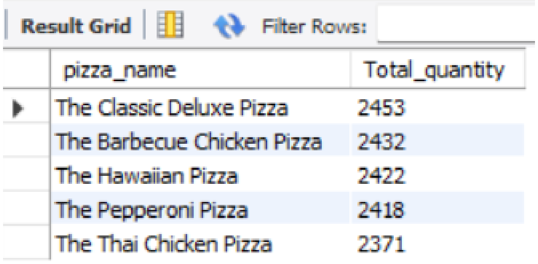


pizza_name	Total_Revenue
The Brie Carre Pizza	11588.4999999999
The Green Garden Pizza	13955.75
The Spinach Supreme Pizza	15277.75
The Mediterranean Pizza	15360.5
The Spinach Pesto Pizza	15596

I. Top 5 Pizzas by Quantity

```
select pizza_name,sum(quantity) as Total_quantity

from pizza_sales

group by pizza_name

order by Total_quantity desc

limit 5;
```

Output:

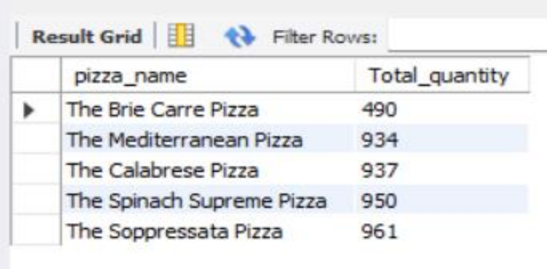


pizza_name	Total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

J. Bottom 5 Pizzas by Quantity

```
select pizza_name,sum(quantity) as Total_quantity

from pizza_sales

group by pizza_name

order by Total_quantity

limit 5;
```

Output:

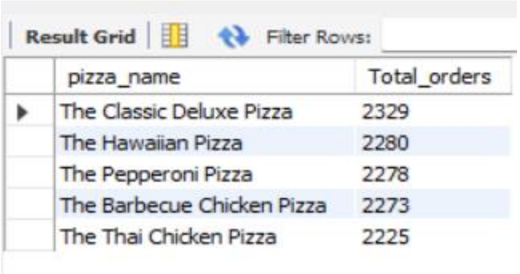


pizza_name	Total_quantity
The Brie Carre Pizza	490
The Mediterranean Pizza	934
The Calabrese Pizza	937
The Spinach Supreme Pizza	950
The Soppressata Pizza	961

K. Top 5 Pizzas by Total Orders

```
select pizza_name,count(distinct order_id) as Total_orders
from pizza_sales
group by pizza_name
order by Total_orders desc
limit 5;
```

Output:

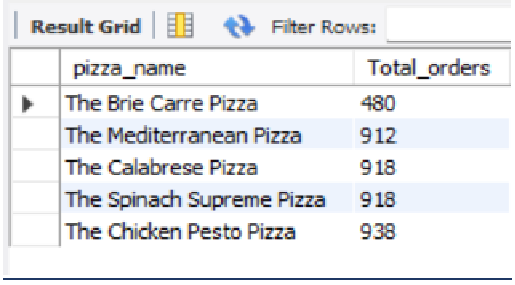


pizza_name	Total_orders
The Classic Deluxe Pizza	2329
The Hawaiian Pizza	2280
The Pepperoni Pizza	2278
The Barbecue Chicken Pizza	2273
The Thai Chicken Pizza	2225

L. Bottom 5 Pizzas by Total Orders

```
select pizza_name,count(distinct order_id) as Total_orders
from pizza_sales
group by pizza_name
order by Total_orders
limit 5;
```

Output:



pizza_name	Total_orders
The Brie Carre Pizza	480
The Mediterranean Pizza	912
The Calabrese Pizza	918
The Spinach Supreme Pizza	918
The Chicken Pesto Pizza	938

NOTE

If you want to apply the pizza_category or pizza_size filters to the above queries you can use WHERE clause. Follow some of below examples

```
SELECT Top 5 pizza_name, COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales
WHERE pizza_category = 'Classic'
GROUP BY pizza_name
ORDER BY Total_Orders ASC
```