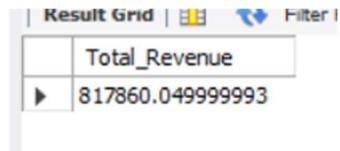


Documentation For Pizza Sales Data (SQL Code with Output)

Finding total revenue

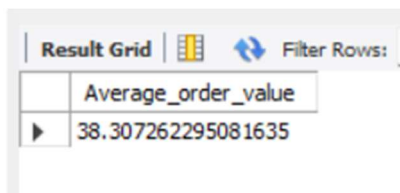
```
select sum(total_price) as Total_Revenue  
from pizza_sales;
```



Total_Revenue
817860.049999993

Average order value

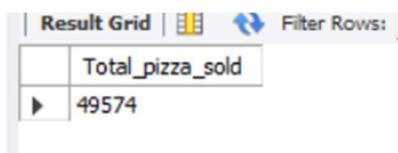
```
select sum(total_price) / count(distinct(order_id)) as Average_order_value  
from pizza_sales;
```



Average_order_value
38.307262295081635

Finding Total pizza sold

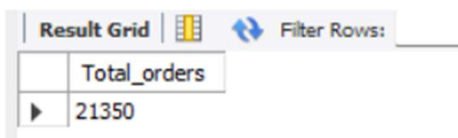
```
select sum(quantity) as Total_pizza_sold from pizza_sales;
```



Total_pizza_sold
49574

Total order placed

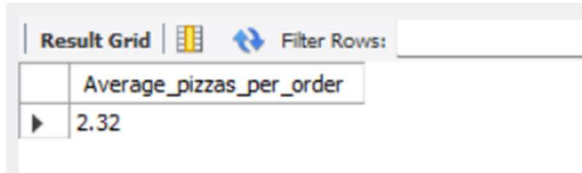
```
select count(distinct(order_id)) as Total_orders from pizza_sales;
```



Total_orders
21350

Average pizza per order

```
select cast(sum(quantity) / count(distinct(order_id)) as decimal(10,2)) as  
Average_pizzas_per_order from pizza_sales;
```

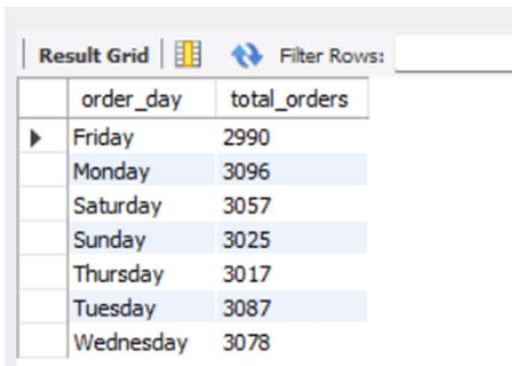


The screenshot shows a SQL query result grid. The header row is labeled 'Average_pizzas_per_order'. The first data row shows the value '2.32'.

Average_pizzas_per_order
2.32

Daily trends for order

```
SELECT DAYNAME(STR_TO_DATE(order_date, '%Y-%m-%d')) AS order_day,  
        COUNT(DISTINCT order_id) AS total_orders  
FROM pizza_sales  
GROUP BY DAYNAME(STR_TO_DATE(order_date, '%Y-%m-%d'));
```



The screenshot shows a SQL query result grid with two columns: 'order_day' and 'total_orders'. The data is grouped by day of the week.

order_day	total_orders
Friday	2990
Monday	3096
Saturday	3057
Sunday	3025
Thursday	3017
Tuesday	3087
Wednesday	3078

hourly trend

```
SELECT HOUR(order_time) AS order_hours,  
        COUNT(DISTINCT order_id) AS total_orders  
FROM pizza_sales  
GROUP BY HOUR(order_time)  
order by HOUR(order_time);
```

Result Grid			Filter Rows:
	order_hours	total_orders	
▶	9	1	
	10	8	
	11	1231	
	12	2520	
	13	2455	
	14	1472	
	15	1468	
	16	1920	
	17	2336	
	18	2399	
	19	2009	
	20	1642	
	21	1198	
	22	663	
	23	28	

Percentage of sales by pizza category

SELECT

 pizza_category,

 sum(total_price) as Total_sales,

 SUM(total_price) * 100 / (

 SELECT SUM(total_price)

 FROM pizza_sales

 WHERE MONTH(STR_TO_DATE(order_date, '%Y-%m-%d')) = 1

) AS PCT

FROM pizza_sales

WHERE MONTH(STR_TO_DATE(order_date, '%Y-%m-%d')) = 1

GROUP BY pizza_category;

Result Grid			
		Filter Rows:	
		Export:	
		Wrap	
	pizza_category	Total_sales	PCT
▶	Classic	18619.4	26.67791894064334
	Veggie	17055.400000000027	24.437016160577095
	Supreme	17929.749999999996	25.68978684200349
	Chicken	16188.75	23.195278056776257

percentage of sale 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

WHERE QUARTER(STR_TO_DATE(order_date, '%Y-%m-%d')) = 1

)

AS DECIMAL(10,2)) AS PCT

FROM pizza_sales

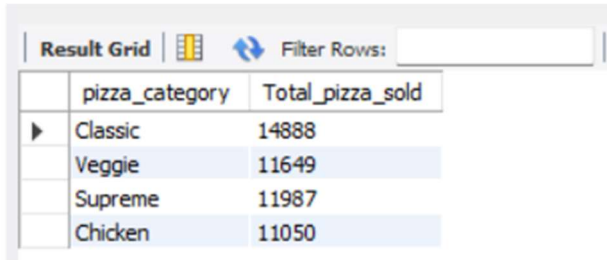
WHERE QUARTER(STR_TO_DATE(order_date, '%Y-%m-%d')) = 1

GROUP BY pizza_size;

Result Grid			
		Filter Rows:	
	pizza_size	Total_sales	PCT
▶	M	61159.00	29.78
	L	95229.65	46.37
	S	45384.25	22.10
	XL	3289.50	1.60
	XXL	287.60	0.14

#Total pizza sold by pizza category

```
select pizza_category , sum(quantity) as Total_pizza_sold from pizza_sales  
group by pizza_category;
```

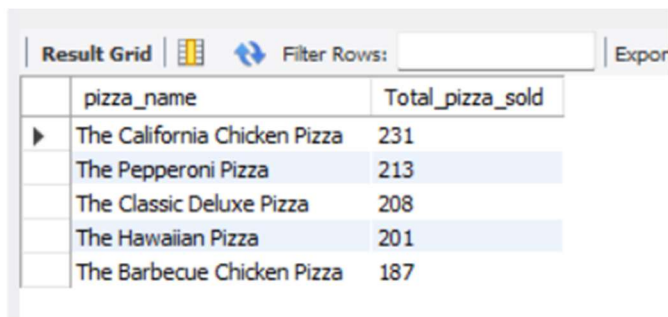


A screenshot of a database result grid. At the top, there is a 'Result Grid' tab, a 'Filter Rows' input field, and a 'Filter Rows' button. Below this is a table with two columns: 'pizza_category' and 'Total_pizza_sold'. The table contains four rows of data.

pizza_category	Total_pizza_sold
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

top 5 best seller by total pizza sold

```
select pizza_name , sum(quantity) as Total_pizza_sold  
from pizza_sales  
where month(str_to_date(order_date , '%Y-%m-%d')) = 8  
group by pizza_name  
order by sum(quantity) desc  
limit 5;
```



A screenshot of a database result grid. At the top, there is a 'Result Grid' tab, a 'Filter Rows' input field, a 'Filter Rows' button, and an 'Export' button. Below this is a table with two columns: 'pizza_name' and 'Total_pizza_sold'. The table contains five rows of data, representing the top 5 best-selling pizzas.

pizza_name	Total_pizza_sold
The California Chicken Pizza	231
The Pepperoni Pizza	213
The Classic Deluxe Pizza	208
The Hawaiian Pizza	201
The Barbecue Chicken Pizza	187

bottom 5 sold or less sold

```
select pizza_name , sum(quantity) as Total_pizza_sold  
from pizza_sales  
group by pizza_name  
order by sum(quantity) ASC  
limit 5;
```

Result Grid			Filter Rows:	Export:
	pizza_name	Total_pizza_sold		
▶	The Brie Carre Pizza	490		
	The Mediterranean Pizza	934		
	The Calabrese Pizza	937		
	The Spinach Supreme Pizza	950		
	The Soppressata Pizza	961		