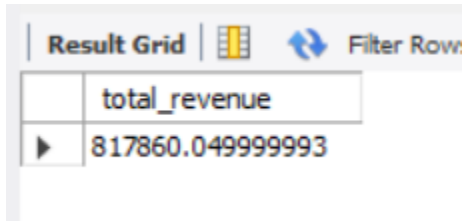


## A. KPI'S

### 1. TOTAL REVENUE :

```
select SUM(total_price) AS total_revenue from pizza_sales
```

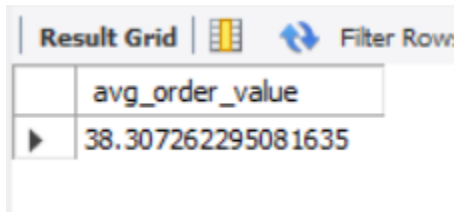


The screenshot shows a 'Result Grid' with a toolbar containing a grid icon, a refresh icon, and a 'Filter Rows' button. The grid has two columns: the first column is empty, and the second column is labeled 'total\_revenue'. There is one row with a right-pointing arrow in the first column and the value '817860.049999993' in the second column.

	total_revenue
▶	817860.049999993

### 2. AVERAGE ORDER VALUE

```
SELECT Sum(total_price)/count(distinct order_id) as avg_order_value  
from pizza_sales
```



The screenshot shows a 'Result Grid' with a toolbar containing a grid icon, a refresh icon, and a 'Filter Rows' button. The grid has two columns: the first column is empty, and the second column is labeled 'avg\_order\_value'. There is one row with a right-pointing arrow in the first column and the value '38.307262295081635' in the second column.

	avg_order_value
▶	38.307262295081635

### 3. TOTAL PIZZAS SOLD

```
select SUM(quantity) AS total_pizza_sold from pizza_sales
```



The screenshot shows a 'Result Grid' with a toolbar containing a grid icon, a refresh icon, and a 'Filter Rows' button. The grid has two columns: the first column is empty, and the second column is labeled 'total\_pizza\_sold'. There is one row with a right-pointing arrow in the first column and the value '49574' in the second column.

	total_pizza_sold
▶	49574

### 4. TOTAL ORDERS

```
SELECT COUNT(DISTINCT order_id)AS TOTAL_ORDERS FROM pizza_sales
```

Result Grid		Filter Rows:
	TOTAL_ORDERS	
▶	21350	

## 5. AVERAGE PIZZAS PER ORDER

```
SELECT CAST(SUM(quantity) AS DECIMAL(10,2)) /
CAST(COUNT(DISTINCT order_id) AS DECIMAL(10,2))
AS Avg_Pizzas_per_order
FROM pizza_sales
```

Result Grid		Filter Rows:
	Avg_Pizzas_per_order	
▶	2.321967	

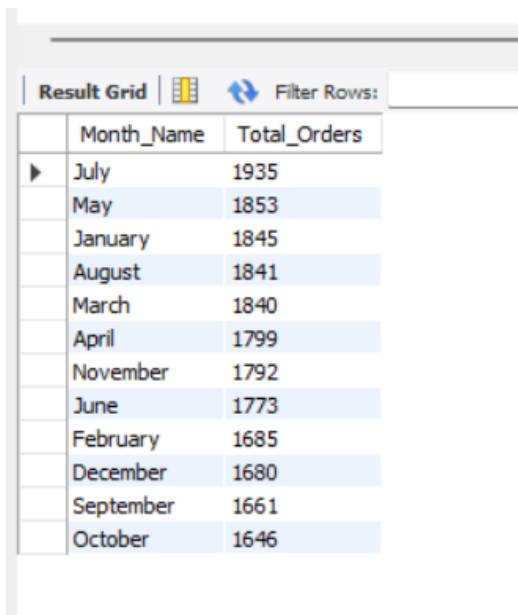
## B.DAILY TREND FOR TOTAL ORDERS

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

Result Grid		Filter Rows:
	order_day	total_orders
▶	Friday	3538
	Monday	2794
	Saturday	3158
	Sunday	2624
	Thursday	3239
	Tuesday	2973
	Wednesday	3024

## C. MONTHLY TREND FOR ORDERS

```
SELECT MONTHNAME(STR_TO_DATE(order_date, '%d-%m-%Y')) AS Month_Name,  
COUNT(DISTINCT order_id) AS Total_Orders  
FROM pizza_sales  
GROUP BY MONTHNAME(STR_TO_DATE(order_date, '%d-%m-%Y'))  
ORDER BY Total_Orders DESC
```





The screenshot shows a database query result grid with two columns: 'Month\_Name' and 'Total\_Orders'. The data is sorted in descending order of total orders. The months are listed from July at the top to October at the bottom. Each row has a small blue triangle icon to its left, indicating it is a clickable link. The grid is titled 'Result Grid' and has a 'Filter Rows' button.

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



## D. % OF SALES BY PIZZA CATEGORY

```
SELECT pizza_category, CAST(SUM(total_price) AS DECIMAL(10,2)) as  
total_revenue,  
CAST(SUM(total_price) * 100 / (SELECT SUM(total_price) from pizza_sales) AS  
DECIMAL(10,2)) AS PCT  
FROM pizza_sales  
GROUP BY pizza_category
```

Result Grid     Filter Rows: <input type="text"/>			
	pizza_category	total_revenue	PCT
	Chicken	195919.50	23.96
	Classic	220053.10	26.91
▶	Supreme	208197.00	25.46
	Veggie	193690.45	23.68



## E. % of Sales by Pizza Size

```
SELECT pizza_size, CAST(SUM(total_price) AS DECIMAL(10,2)) as total_revenue,
CAST(SUM(total_price) * 100 / (SELECT SUM(total_price) from pizza_sales) AS
DECIMAL(10,2)) AS PCT
FROM pizza_sales
GROUP BY pizza_size
ORDER BY pizza_size
```

Result Grid     Filter Rows: <input type="text"/>			
	pizza_size	total_revenue	PCT
▶	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

## F. Total Pizzas Sold by Pizza Category in February

```
SELECT pizza_category, SUM(quantity) AS Total_Quantity_Sold
FROM pizza_sales
WHERE MONTH(STR_TO_DATE(order_date, '%d-%m-%Y')) = 2
GROUP BY pizza_category
ORDER BY Total_Quantity_Sold DESC
```

Result Grid     Filter Rows: <input type="text"/>		
	pizza_category	Total_Quantity_Sold
▶	Classic	1178
	Supreme	964
	Veggie	944
	Chicken	875

## 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
```

Result Grid     Filter Rows: <input type="text"/>   Exp		
	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 ASC
```

```
LIMIT 5
```

Result Grid			Filter Rows:	Export
	pizza_name	Total_Revenue		
▶	The Brie Carre Pizza	11588.499999999999		
	The Green Garden Pizza	13955.75		
	The Spinach Supreme Pizza	15277.75		
	The Mediterranean Pizza	15360.5		
	The Spinach Pesto Pizza	15596		

