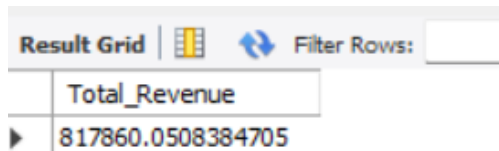


Pizza Sales Analysis (SQL, Excel) Documentation

KPI Requirement

1 Total Revenue

```
SELECT SUM(total_Price) AS Total_Revenue  
FROM pizza_sales;
```

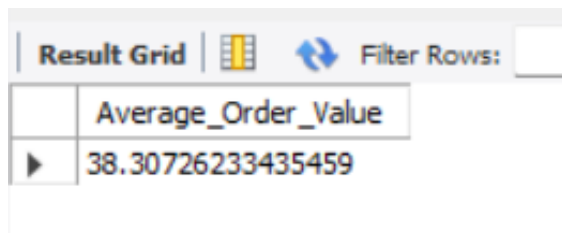


The screenshot shows a SQL query result grid with a header row and one data row. The header row is labeled 'Total_Revenue' and the data row shows the value '817860.0508384705'. The interface includes a 'Result Grid' tab, a 'Filter Rows' button, and a search bar.

Total_Revenue
817860.0508384705

2 Average Order VALUE

```
SELECT SUM(total_Price) / COUNT(DISTINCT order_id) AS Average_Order_Value  
FROM pizza_sales;
```

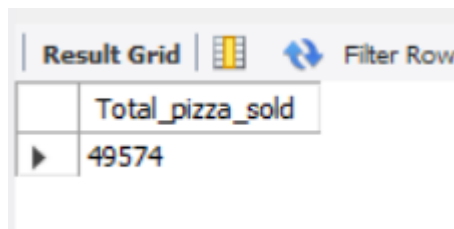


The screenshot shows a SQL query result grid with a header row and one data row. The header row is labeled 'Average_Order_Value' and the data row shows the value '38.30726233435459'. The interface includes a 'Result Grid' tab, a 'Filter Rows' button, and a search bar.

Average_Order_Value
38.30726233435459

3 Total pizzas sold

```
SELECT SUM(quantity) as Total_pizza_sold  
FROM pizza_sales;
```



The screenshot shows a SQL query result grid with a header row and one data row. The header row is labeled 'Total_pizza_sold' and the data row shows the value '49574'. The interface includes a 'Result Grid' tab, a 'Filter Rows' button, and a search bar.

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 SUM(quantity)/COUNT(DISTINCT order_id) AS avg_pizzas_per_order
FROM pizza_sales;
```

Result Grid	Filter Rows:
	avg_pizzas_per_order
▶	2.3220

charts requirement

1 daily trend for total orders

```
SELECT
    DAYNAME(order_date) AS order_day,
    COUNT(DISTINCT order_id) AS total_orders
FROM pizza_sales
GROUP BY order_day
ORDER BY FIELD(order_day,
    'Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday');
```

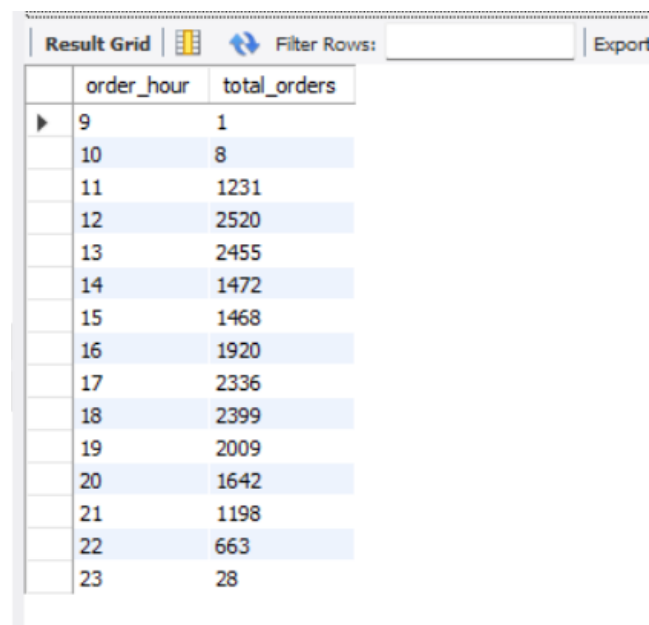
Result Grid

Filter Rows:

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

2 Hourly Trend for total orders

```
SELECT hour(order_time) AS order_hour,  
COUNT(DISTINCT order_id) AS total_orders  
FROM pizza_sales  
GROUP BY order_hour  
ORDER BY order_hour;
```

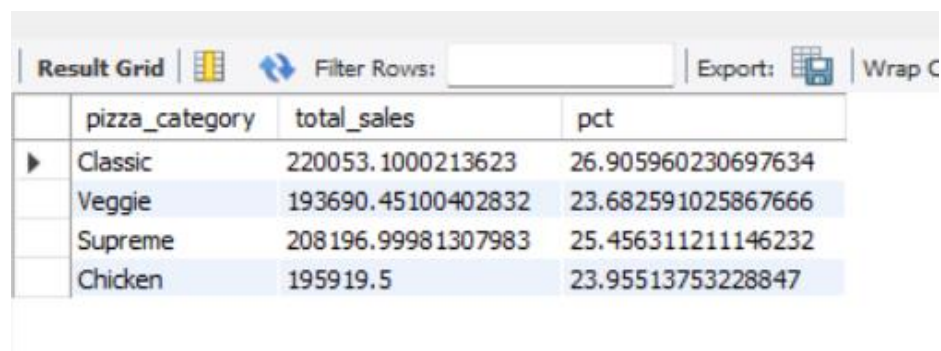


The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'order_hour' and 'total_orders'. The data shows a peak in orders around hour 12 and 13, followed by a decline towards the end of the day.

order_hour	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

3 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) AS pct  
FROM pizza_sales  
GROUP BY pizza_category;
```





The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with three columns: 'pizza_category', 'total_sales', and 'pct'. The data shows that the 'Classic' category has the highest total sales and percentage, followed by 'Veggie', 'Supreme', and 'Chicken'.

pizza_category	total_sales	pct
Classic	220053.1000213623	26.905960230697634
Veggie	193690.45100402832	23.682591025867666
Supreme	208196.99981307983	25.456311211146232
Chicken	195919.5	23.95513753228847



4 Percentage 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
         WHERE MONTH(order_date) IN (1,2,3)) AS DECIMAL(10,2)
    ) AS pct
FROM pizza_sales
WHERE MONTH(order_date) IN (1,2,3)
GROUP BY pizza_size
ORDER BY pct DESC;
```

Result Grid   Filter Rows: <input type="text"/>			
	pizza_size	total_sales	pct
▶	L	95229.65	46.37
	M	61159.00	29.78
	S	45384.25	22.10
	XL	3289.50	1.60
	XXL	287.60	0.14

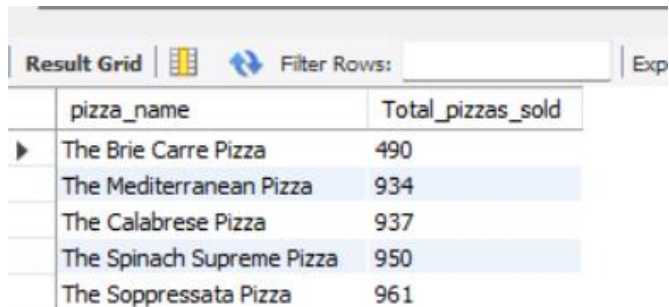
5 Total pizza sold by pizza category

```
SELECT pizza_category , SUM(quantity) AS Total_pizza_sold From pizza_sales
GROUP BY pizza_category;
```

Result Grid   Filter Rows: <input type="text"/>		
	pizza_category	Total_pizza_sold
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

6 Top 5 best sellers by total prize

```
SELECT pizza_name,SUM(quantity) AS Total_pizzas_sold FROM pizza_sales  
GROUP BY pizza_name  
ORDER BY SUM(quantity)  
LIMIT 5;
```

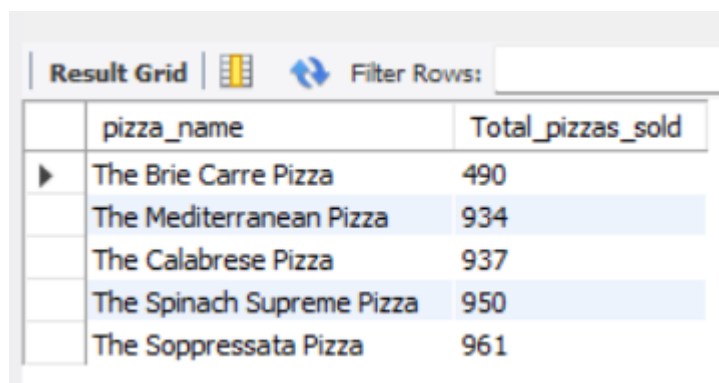


The screenshot shows a database interface with a 'Result Grid' tab. It displays a table with two columns: 'pizza_name' and 'Total_pizzas_sold'. The table lists the top 5 pizzas based on total quantity sold, ordered from highest to lowest. The rows are: The Brie Carre Pizza (490), The Mediterranean Pizza (934), The Calabrese Pizza (937), The Spinach Supreme Pizza (950), and The Soppressata Pizza (961). The interface includes a 'Filter Rows' button and an 'Exp' button.

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

7 Botton 5 sellers by total prize

```
SELECT pizza_name,SUM(quantity) AS Total_pizzas_sold FROM pizza_sales  
GROUP BY pizza_name  
ORDER BY SUM(quantity) ASC  
LIMIT 5;
```



The screenshot shows a database interface with a 'Result Grid' tab. It displays a table with two columns: 'pizza_name' and 'Total_pizzas_sold'. The table lists the bottom 5 pizzas based on total quantity sold, ordered from lowest to highest. The rows are: The Brie Carre Pizza (490), The Mediterranean Pizza (934), The Calabrese Pizza (937), The Spinach Supreme Pizza (950), and The Soppressata Pizza (961). The interface includes a 'Filter Rows' button and an 'Exp' button.

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