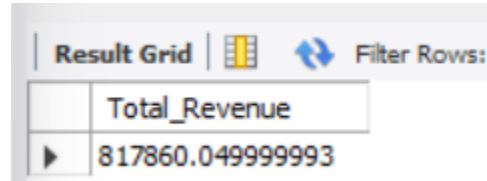


Pizza Sales SQL Queries

A. KPI:

1. Total Revenue:

SELECT SUM(total_price) AS Total_Revenue from pizza_Sales

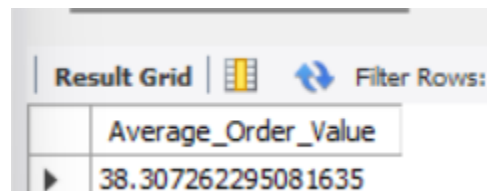


The screenshot shows a SQL query result grid with the following content:

	Total_Revenue
▶	817860.049999993

2. Average Order Value:

SELECT SUM(total_price) / COUNT(DISTINCT order_id) AS
Average_Order_Value from pizza_Sales;

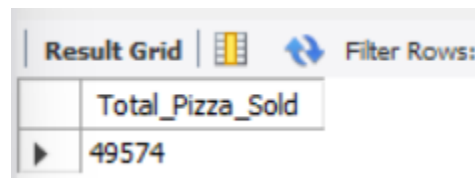


The screenshot shows a SQL query result grid with the following content:

	Average_Order_Value
▶	38.307262295081635

3. Total Pizza Sold:

SELECT SUM(quantity) AS Total_Pizza_Sold from pizza_Sales;

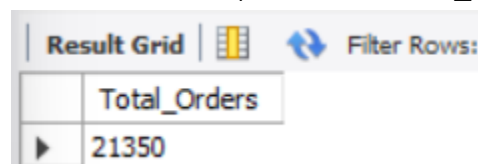


The screenshot shows a SQL query result grid with the following content:

	Total_Pizza_Sold
▶	49574

4. Total Orders:

SELECT COUNT(DISTINCT order_id) as Total_Orders from pizza_Sales;

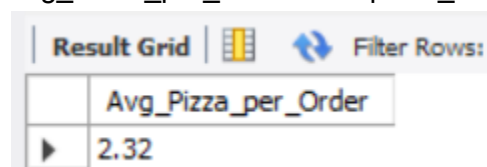


The screenshot shows a SQL query result grid with the following content:

	Total_Orders
▶	21350

5. Average Pizzas per Order:

SELECT CAST(CAST(SUM(quantity) as DECIMAL(10,2))/
CAST(COUNT(DISTINCT order_id) AS DECIMAL(10,2)) AS DECIMAL(10,2)) AS
Avg_Pizza_per_Order from pizza_Sales;





The screenshot shows a SQL query result grid with the following content:

	Avg_Pizza_per_Order
▶	2.32

B.



1. Daily Trend for Total Orders:

```
SELECT
    DAYNAME(STR_TO_DATE(order_date, '%m/%d/%Y')) AS day_of_week,
    COUNT(DISTINCT order_id) AS total_orders
FROM
    pizza_sales
GROUP BY
    day_of_week
ORDER BY
    FIELD(day_of_week, 'Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday');
```

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



2. Monthly Trend for Total Orders:

```
SELECT
    MONTHNAME(STR_TO_DATE(order_date, '%m/%d/%Y')) AS month,
    COUNT(DISTINCT order_id) AS total_orders
FROM
    pizza_sales
GROUP BY
    month
ORDER BY
    FIELD(month, 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
'September', 'October', 'November', 'December');
```

Result Grid   Filter Rows:		
	month	total_orders
▶	January	1845
	February	1685
	March	1840
	April	1799
	May	1853
	June	1773
	July	1935
	August	1841
	September	1661
	October	1646
	November	1792
	December	1680


3. Percentage of Sales by Pizza Category:

SELECT pizza_category, (SUM(total_price)*100)/(SELECT SUM(total_price) from pizza_Sales) AS _Percentage_Total_Sales from pizza_Sales
group by pizza_category

Result Grid   Filter Rows:		
	pizza_category	_Percentage_Total_Sales
▶	Classic	26.9059602556699
	Veggie	23.682590927384783
	Supreme	25.45631126009884
	Chicken	23.955137556847493

Monthwise:

```
SELECT
pizza_category,
(SUM(total_price) * 100) /
(SELECT SUM(total_price)
FROM pizza_sales
WHERE MONTHNAME(STR_TO_DATE(order_date, '%m/%d/%Y')) = 'January') AS
Percentage_Total_Sales
FROM
pizza_sales
WHERE
MONTHNAME(STR_TO_DATE(order_date, '%m/%d/%Y')) = 'January'
GROUP BY
pizza_category;
```


Result Grid  Filter Rows: <input type="text"/>		
	pizza_category	Percentage_Total_Sales
▶	Classic	26.67791894064334
	Veggie	24.437016160577095
	Supreme	25.68978684200349
	Chicken	23.195278056776257

4. Percentage of Sales by Pizza Size

```

SELECT
    pizza_size,
    (SUM(total_price) * 100) /
    (SELECT SUM(total_price)
     FROM pizza_sales
     WHERE MONTHNAME(STR_TO_DATE(order_date, '%m/%d/%Y')) = 'January') AS
    Percentage_Total_Sales
FROM
    pizza_sales
WHERE
    MONTHNAME(STR_TO_DATE(order_date, '%m/%d/%Y')) = 'January'
GROUP BY
    Pizza_size;

```



Result Grid  Filter Rows: <input type="text"/>		
	pizza_size	Percentage_Total_Sales
▶	M	30.00789474061268
	L	46.421934483682676
	S	21.640329372590223
	XL	1.8268229185322973
	XXL	0.10301848458233112

5. Top 5 Pizzas by Revenue

```



SELECT pizza_name, SUM(total_price) AS revenue from pizza_Sales
GROUP BY pizza_name
ORDER BY revenue DESC LIMIT 5;

```

Result Grid   Filter Rows: <input type="text"/>		
	pizza_name	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



6. Bottom 5 pizzas by Revenue:

```
SELECT pizza_name, SUM(total_price) AS revenue from pizza_Sales
GROUP BY pizza_name
ORDER BY revenue LIMIT 5;
```

Result Grid   Filter Rows: <input type="text"/>		
	pizza_name	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



7. Top 5 pizzas by quantity:

```
SELECT pizza_name, SUM(total_price) AS revenue from pizza_Sales
GROUP BY pizza_name
ORDER BY revenue LIMIT 5;
```

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

8. Top 5 Pizzas by 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;
```

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